

Contents

Chapter 1

Anise Manual

1.1 License

anise is written by Josef Hahn under the terms of the GPLv3 or higher.

Please read the `LICENSE` file from the package and the [Dependencies](#) section for included third-party stuff.

1.2 About

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 which 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.

1.3 Up-to-date?

Are you currently reading from another source than the homepage? Are you in doubt if that place is up-to-date? If yes, you should visit <https://pseudopolis.eu/wiki/pino/projs/anise> and check that. You are currently reading the manual for version 4.1.4792.

1.4 Maturity

In this version, the state of anise is considered as production-stable.

1.5 Dependencies

There are external parts which are used by anise. Many thanks to the projects and all participants.

Python 3.4 *required*

Typical GNU/Linux Desktop *recommended*

PyQt5 incl. WebEngine *required (has alternatives)*: for some user interface enhancements; you can comfortably work without it (inside the browser then)

subversion *optional*

git *optional*

font 'Symbola' *included*: for logo symbol; free for use; copied from [here](#).

banner image *included*: [_meta/homepage_bannerimage.png](#); license [CC BY-SA 3.0](#); copied from [here](#).

third-party project logos *included*: from some projects (see their websites for details).

all files in `/_meta` *included*: if not mentioned otherwise, Copyright 2015 Josef Hahn under license [CC BY-SA 3.0](#) license.

icon set 'oxygen' *included*: files 'anise/data/icons/userfeedback'; license see [homepage](#).

1.6 Introduction

Please read how to make Anise ready for the first steps in [Appendix: Installation](#).

1.6.1 Typical Usage Scenarios

In very general words, the About section already described some scenarios, in which Anise could bring substantial advantages in terms of ergonomics, reliability and quality for your processes.

Basically, whenever you are going to automate the interaction of many different tools, Anise provides you with an engine for your glue code. This can either mean to orchestrate a large sequence of tool executions and to fulfill the dependencies they might have. But a large field of small tasks, each just calling one or two of tools, is also worth considering to consolidate in a list of Anise projects. Instead of zillions of different usage conventions, you run all of them in the same way on top of Anise with always the same user interface. This also includes access to the Anise user interface foundation, which is be the basis for many kinds of user interaction in task executions as well as lots of graphical helpers and configuration guides.

While the Anise engine is open for lots of very different automation tasks, the program comes with a large list of ready-to-use implementations for some specific tasks around software development and package deployment. Those tasks can run some compilers, create tarballs, Debian packages and some more, can run some test frameworks, documentation generation via Doxygen (development documentation with a programming interface reference as well as textual documentation) and can build and deploy a project website. Some of those included tasks may be useful for certain scenarios. But in most cases it is probably required to actually implement most of the tasks (i.e. the glue between your tools), while Anise provides a scaffold and a common facade for them.

You will learn later about the [The Anise Execution Model](#), the [Programming Interface](#) and about ways to actually execute tasks. You will also read a bit more about the [specific task implementations](#) later.

1.6.2 First Steps

The first steps are to create a project description file for your Anise automation project and to fill it with an actual configuration. This section shows a simple example which helps for a very trivial but functional Anise project.

Start Anise from the start menu or execute 'anise' on the command line. An Anise interface should come up, asking you for some action.

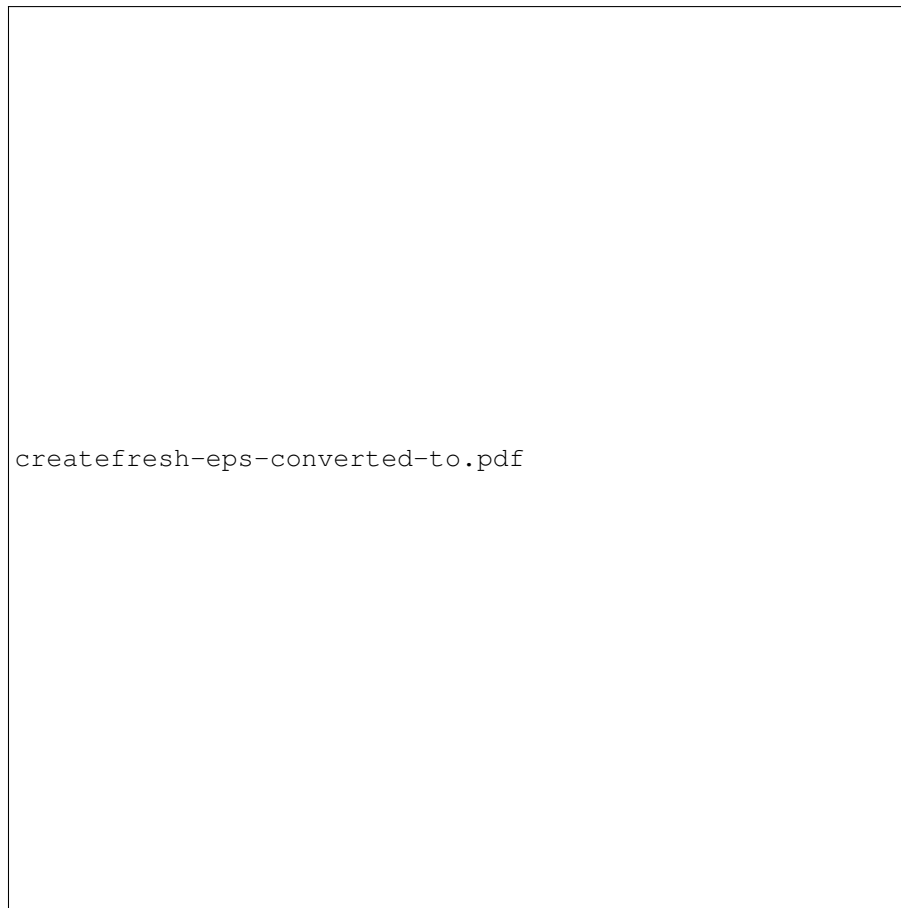


Figure 1.1 The first Anise screen

Choose the root directory of your project here. If you want to create a fresh automation project and you don't have any files yet, create a fresh directory for it somewhere beforehand.

Select 'Create a fresh project here' when you have reached the project root directory. This creates the file `_↔_meta/_projectdesc` (Windows: `_meta_projectdesc`), which is called the *project description file*. This file can either be edited manually with an arbitrary text editor or in a guided way.

For a fresh project, the Anise Console will open (for an unfresh one, the console is still available as task 'console' from the task chooser). It contains all kinds of configuration guides and is useful in other situations as well as we see later.

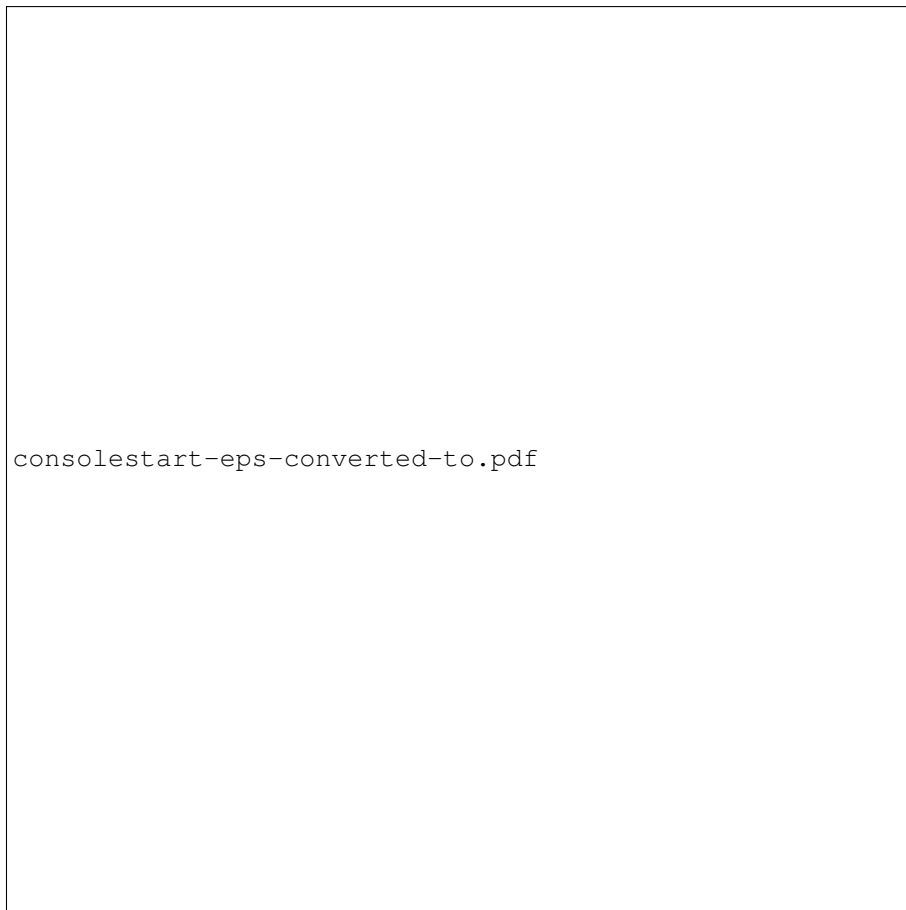


Figure 1.2 The Anise Console after start

Find the "Manage Tasks" action in the main part, trigger it and follow the on-screen instructions.

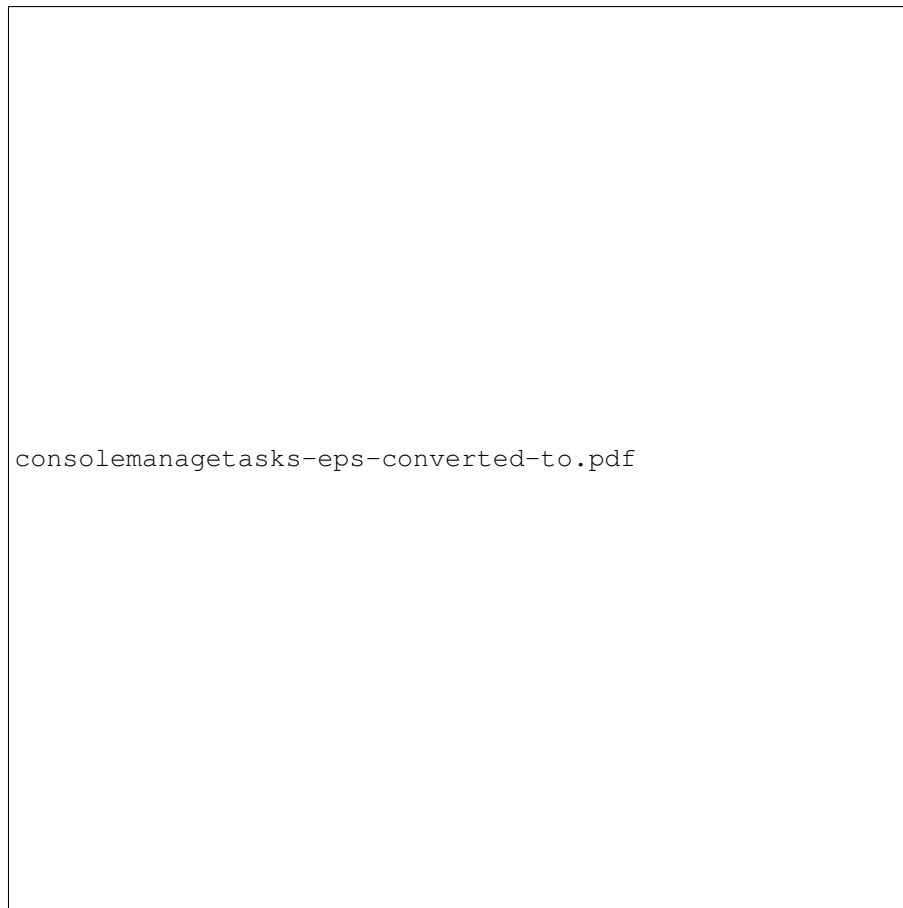


Figure 1.3 The available actions

In the end, you should have a new task with a more or less useful Python implementation added to your project.

After closing the console, execute `anise` again for a task chooser or `anise mytask` for directly executing the task `mytask`.



Figure 1.4 The Anise task chooser

You will find more details about the console in the [Anise Console](#) section.

1.7 The Anise Execution Model

The execution model describes what the Anise engine actually executes and which runtime environment and infrastructure it prepares for a task implementation.

1.7.1 The Universe Object

The universe object is the central part of the execution of tasks. It is one global data structure, which contains all the information for executing tasks. This includes all the metadata, like the project's name, maybe its license and so on, as well as all task implementations.



Figure 1.5 The Universe Object seen in the console

You will learn later how to get and manipulate this object in different situations.

1.7.2 The Task

A task is the piece of program logic for execution on the Anise engine. Each automation action of your project is to be implemented in one task.

Technically, a task is just a simple Python function (for people who already took a look at `_meta/_projectdesc`: [anise.features.base.pytask](#) is just a small wrapper around that fact), which is available as a member of the universe object. It must not require any arguments, otherwise the engine can't execute it as a task. Instead, member variables from the universe objects or user interaction are used for getting information or parameters from outside.

In [First Steps](#) you already created a task in a very direct way. Later parts will also show more indirect ways of creating tasks.

1.7.3 The Project Description File

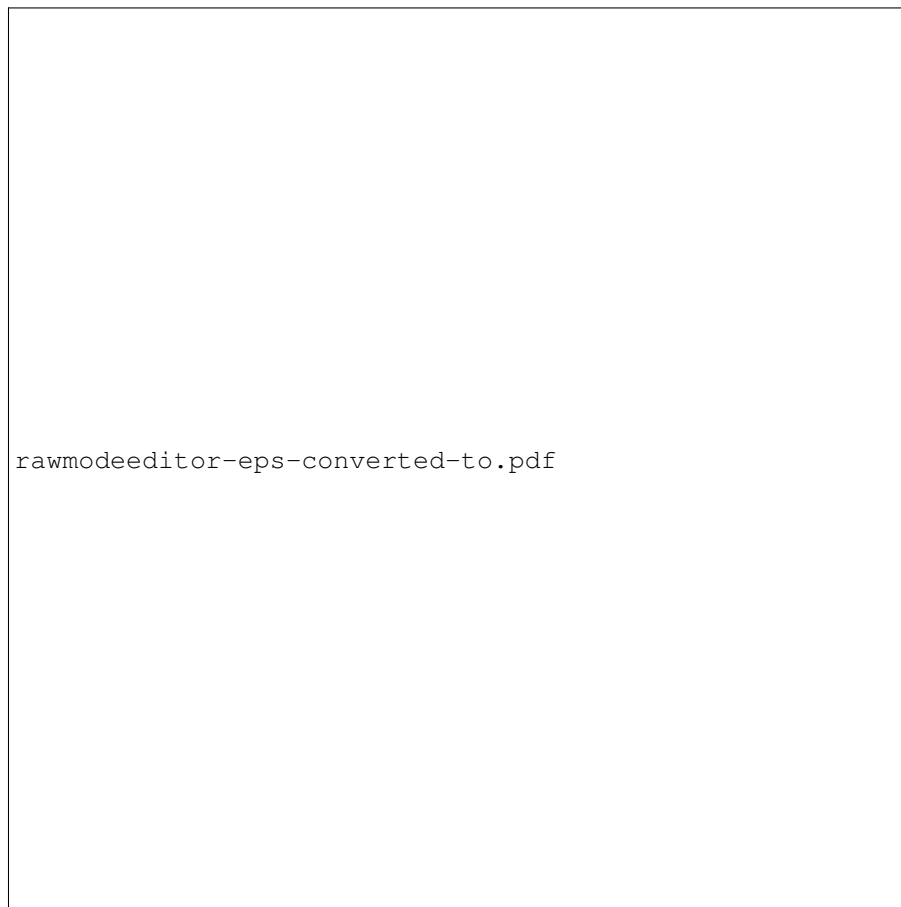
The project description file must be either in the `_meta` subdirectory (recommended!) or in the root directory of your project and must be named `_projectdesc`. All configurations and every relevant piece of information about your project are specified in this file. The complete Anise operation will run just on top of it (this is not entirely correct, but correct enough for the moment).

You might already have created the initial version of this file in the [First Steps](#). This file contains an xml document in an Anise-specific format. The general structure is the following:

```
<?xml version="1.0" ?>
<aniseprojectdesc>
  <val k="name" v="myprojectname"/>
  <val k="base.version" v="0.1"/>
  <let k="somefeature.someobject.somemethod">
    <val k="someparam" t="str" v="foo"/>
    <val k="anotherparam" t="bool" v="True"/>
  </let>
</aniseprojectdesc>
```

This file essentially populates the universe object with information before the actual execution begins. The root `aniseprojectdesc` contains a list of `val` and `let` nodes.

A graphical, yet not too abstract way to inspect and change the content of this file is the 'Raw Mode Editor'. You find it in the console, listed in the 'Available actions' for the root node.



rawmodeeditor-eps-converted-to.pdf

Figure 1.6 Project description in the Raw Mode Editor

1.7.3.1 The 'val' Node

A `val` node assigns values to the universe object (at least if it is a direct child of the root node).

The `k` attribute specifies the member name used for storing the value in the universe object.

The value either has a primitive type like a string or a boolean or something composed like an instance of a Python class. The type is specified in the `t` attribute. If a `val` node defines a non-primitive value, it contains nested `val` structures for each constructor parameter.

The types `list` and `dict` exist as typical containers. They also contain nested `val` structures holding the actual content. For the former one, the child nodes must not have `k` attributes (since elements have keys in dictionaries but not in list)!

There is one more type; it has a very special behavior: A value of type `objectref` contains a snippet of python code which will be evaluated at the time the xml file is read. It may be something as simple as a variable name like `<val k="base.license" oref="licensing.GPLv3OrHigher"/>` or something more complex like `<val k="sometext" oref="utils.basic.readfromfile('...')">`, which dynamically generates the content for a variable.

Note: This evaluation happens very early. It is not possible to refer to values, which become assigned in later stages, like the [anise.framework.engine.HOOK_AFTER_PREPARATION](#) hook.

Those values will potentially be read by the task at execution time in order to configure what it does in detail.

1.7.3.2 The 'let' Node

A `let` node is a different way to populate the universe object. It doesn't directly lead to a new member in it, but it triggers a management function (don't confuse those functions with tasks; tasks could become executed in later steps if the user requested it), which in turn modifies some data structures of the universe object. This encapsulates some rather complex data structures and give you an easier interface to them (as an example, think about a function for adding a bunch of file names together with descriptions to an `mediagallery` data structure).

The `k` attribute specifies the full qualified method name, which includes the feature name (the [Features](#) section will explain this term), maybe a sub-structure within it, and, of course, the method name - like `documentation.↔pools.add`.

Parameter assignments to this function are specified with `val` subnodes. The `k` attributes of them specify the names of the parameters. As a shortcut, string parameters are also allowed to be written directly as attributes of the `let` node.

1.7.4 Executing Tasks

Once a [project description file](#) is created and configured, the user primarily runs Anise for executing tasks. Which tasks are available is specified - although often in an indirect way - and/or parameterized in the project description file. Assuming to have a `dorelease` task available, simply call

```
anise dorelease
```

in a command line **from the project directory**. If this call returns a result value, it is written to the terminal. Find a description of [command-line options](#) below.

1.7.5 The Hook System

The hook system is a simple and flexible event mechanism. The Anise engine itself as well as various parts of [feature](#) implementations use them for loosely working together. It is one of the most important and visible mechanisms in Anise.

Events are represented by a name string, which typically is available as a constant somewhere. The name of this constant typically begins with `HOOK_`, like in `HOOK_MYEVENT`.

```
HOOK_MYEVENT = features.Hook()
```

A feature can register some handlers to it. Those will eventually be executed when the event is triggered.

Some piece of code may feel responsible for *providing* (resp. *triggering*) this event. It does so by asking the Anise infrastructure for a list of all registered handlers and executing them. The exact event execution model is up to this provider. It may execute all registered handlers, parts of them or cancel the execution when some conditions arrive.

You find more infos and sample code in [... use hooks](#) and [... define and trigger hooks](#).

The exact meaning of the phrase 'hook' may differ from case to case: Sometimes it means an event itself, e.g. the notional `HOOK_MYEVENT`. In other situations it means a single handler for an event. It should be clear by the context which meaning applies.

1.8 The Anise Programming Interface

The Anise Programming Interface provides ways for automation projects to interact with the Anise infrastructure. This section gives a coarse overview of all the building blocks and how they play together. It also touches some of the internal parts.

After reading the manual, the best way to get a less abstract knowledge is to take a look at [Appendix: Recipes, How To](#).

There is also an API reference available, which gives detailed information about the existing classes, functions and parameters. Whenever the manual refers to those items (they all begin with 'anise.'), it is a good idea to read the reference as well. It often contains lots of relevant additional details.

1.8.1 Framework

The namespace [anise.framework](#) contains the core foundation. The Anise engine resides here as well as its auxiliary data structures and object classes. Find those modules (and some more) in this namespace:

- [anise.framework.engine](#): The Anise engine and some direct helpers.
- [anise.framework.projects](#): This module read and writes [project description files](#) and implements the [The Universe Object](#) (in [anise.framework.projects.Universe](#)) as well as the interaction between those parts.
- [anise.framework.features](#): The Anise plug-in system (typically called 'features'). The entire high-level functionality of an Anise project (the custom one as well as included one) comes as a 'feature'. This module does not contain any actual features but just the framework logic for loading and running them. More details are described in the [Features](#) section.
- [anise.framework.exceptions](#): Anise exception types.
- [anise.framework.files](#): Some data structures for working with file hierarchies.
- [anise.framework.report](#): Keeps progress information and output text of the task execution.

1.8.2 Features

Anise features provide the complete high-level automation functionality. Each available feature is plugged into the engine on startup. By adding handlers to some [hooks](#), it can execute initialization steps for bringing some configuration data structures in place. Those are configured according to the actual project needs in the [project description file](#). A feature can also provide task implementations - either for manually adding them or automatically added to the universe object and thereby directly ready for execution.

Manually adding a task is possible with [anise.framework.projects.Universe.addtask](#), as [some examples later on](#) will show.

The namespace [anise.features](#) contains all features which are included in Anise. Some of them fulfill rather special tasks, but others are more generic, maybe even have infrastructure characteristics, and often used by other features as well.

Referring to items of features from within a project description file requires a name in Python module notation relative to the feature loading locations! This means, for the features included in anise, the prefix [anise.features](#) is to be omitted.

```
<?xml version="1.0" ?>
<aniseprojectdesc>
  <let k="releasing.tasks.add" destination="/tmp/mymotdserver/">
    <!-- this is from anise.features.releasing --> ...
  </let>
  <let k="dependencies.pool.add">
    <!-- this is from anise.features.dependencies --> ...
  </let>
</aniseprojectdesc>
```

Note: Although the developer reference contains documentation about [anise.features](#) as well, the value of it is restricted due to its static nature. The relevant parts of most features appear at runtime by means of dynamic initialization routines and can differ depending on the configuration and situation. For a deeper understanding how an existing feature works, it is a much better approach to inspect it in the [Console](#).

Note: A feature name may be identical to a namespace which contains sub-items, similar as it is with Python modules. Due to that, there might be a feature `foo`, but also some sub-features `foo.abc` and `foo.xyz`.

1.8.3 Utils

The namespace [anise.utils](#) contains basic utility functions and low-level counterparts of many of the included features. They are often more flexible on the one hand, but more technical on the other hand. They also do not use the Anise infrastructure (there might be rare exceptions for some reasons), so they also do not interact with the [The Universe Object](#).

They are used by the Anise features, but are often flexible enough for being useful for custom features as well.

1.9 More About Executing Anise

The `anise` executable can be called directly from a start menu link or from command line. It must be called from the root directory of your project (important also for start menu links). It is used for executing your tasks and for guided configuration (which can be more convenient than editing the xml file). The [Appendix: Command Line](#) lists all available parameters.

This tool runs either in web mode or in terminal mode. The web mode is much more comfortable and is the default. There is nothing special to know for the user and no special requirement. It just needs to start a web browser. If this is undesired, please add `--no-gui` to your `anise` command. The same functionality is available in terminal mode as well, but the usability of the web mode is much higher.

1.9.1 Anise Console

The Anise Console is a tool for inspecting the universe object at runtime and to run various configuration tasks. More than that, it is a swiss army knife for issue diagnostics and feature development. The console is available as the 'console' task from your task chooser, but is also started automatically when you create a fresh [project description file](#).

1.9.1.1 Window Parts

The console is made up of those parts:

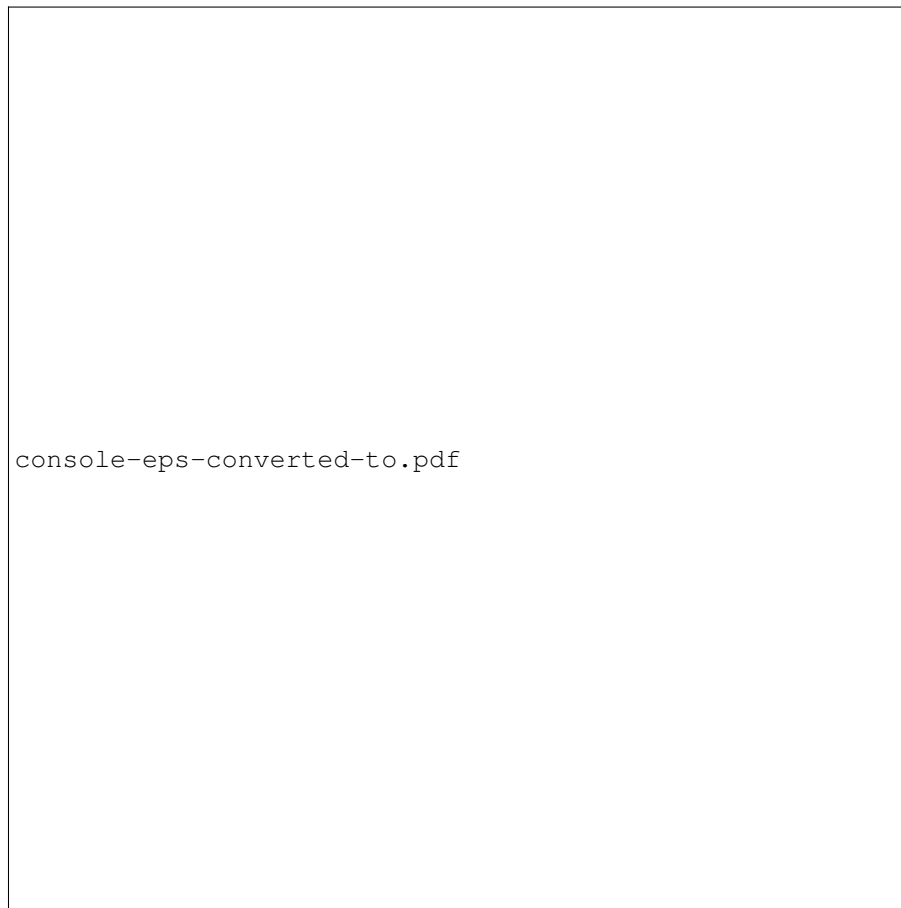


Figure 1.7 The Anise Console

In the top part, there is a header, which shows some general infos and actions.

A tree on the left side shows the runtime content of the [The Universe Object](#).

The info panel on the right side of the tree shows detail information about the node, which is selected in the tree. It lists (amongst other things) actions, which are available to execute directly in the console. Those actions contain graphical configuration guides as well as ways to interact with object in the console.

The terminal in the bottom displays processing output and allows to directly execute Python expressions.

Be aware that all object modifications you make in the terminal are volatile. They are able to modify the current state of the [The Universe Object](#), but they don't change the [project description file](#). When you start Anise again, the old values will be back.

1.9.1.2 Configuration Guides

As noted before, many Feature nodes provide configuration guides. Find those nodes by the 'package' symbol and select them in order to find those guides.

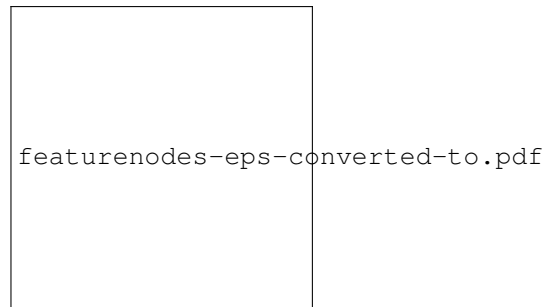


Figure 1.8 Feature nodes

Warning: The configuration of your project description and the runtime content of your universe object are two different things! At startup, the latter is created from the former one, but afterwards they are decoupled. Whenever you make any configuration changes in the console, you have to start a fresh Anise session in order to see the changes applied to your universe object.

1.10 Appendix: Command Line

The Anise command-line tool understands this syntax:

```
anise [taskname] [options]*
```

If a taskname is given, Anise will execute this task, otherwise the `DEFAULT` task, which typically is a task chooser.

Options can be some of the following:

- `--projectfile [file]` : Use this project file (instead of the typical location)
- `--loadfeaturesfrom [path1:...:pathn]` : Sets the paths for loading modules (separated by ":"; builtin features by "builtin")
- `--debug` : Enables debug log output
- `--keeptemp` : Don't remove the temporary stuff (for post-execution diagnostic)
- `--no-autoopen` : Never automatically open files after execution
- `--no-gui` : Don't try to use graphic mode for user interaction
- `--createfresh` : Create a fresh project description file
- `--s:key=value` : Sets a value to the universe object before execution; also works with `i`, `b` and `f` at the beginning for integers, bools and floats instead of strings
- `--help` : Shows this help text

The easiest way to begin working with an unknown Anise project is to open the graphical task chooser by just calling `anise`.

Note: Users of the Anise package for Windows need to open the 'Anise Shell'. Calling `anise` the above way is possible from within this environment only.

1.11 Appendix: Recipes, How To

1.11.1 ... use the included high-level features

The included high-level features of Anise are all visible in the [Console](#). The easiest way is to check the interesting feature nodes there and use configuration guides provided there. Those often give only access to basic functionality and sometimes no guides are implemented at all. In such cases, use the documentation of the subnodes in order to understand how to use the feature.

1.11.2 ... implement custom functionality

There is a two different ways to add custom functionality:

The quick-and-dirty way

Use the 'Manage Tasks' action in the root node of the console in order to add an own task. You will then find this task in the task chooser and also execute it directly with `anise mytask`.

More flexibility brings the 'Manage Hook Scripts' action, which allows you to add handlers for [hooks](#) and so can add a much broader range of custom functionality. For more infos about a particular hook, find its node in the console and read what the info panel provides. Also read [... use hooks](#). A hook script is executed in a very similar way as custom features (i.e. the clean way), so the following recipes typically can be used this way as well.

Both approaches will insert the customization code into your project description file.

The clean way

Implement your custom functionality as an Anise feature and add it to your Anise installation. The entire custom automation code of your project then has the same technical form as the included features. They all use the same infrastructure and are loaded and executed in the same way.

For adding custom logic this way, create a new file for your new feature and call it like `myfeature.py`. In order to actually add it to the Anise loading sequence, place this file into `$HOME/.anise/features` (for Windows users, this is `USERPROFILE%\anise\features`). Placing it deeper into subdirectories give composed feature names, as explained [above](#).

This file will be loaded by the Anise engine in a very similar way as Python would load plain modules. However, the feature loader makes the feature available as a member in the [The Universe Object](#) (and also some bookkeeping stuff). The feature code typically defines some functions and classes for different reasons (see other recipes). It does not directly execute code, but may mark some of the defined objects as hook handlers, so adding all kinds of custom functionality to the Anise execution for all projects (since it is not bound to one project description file anymore).

For executing some code in the initialization of the universe object, the feature could like like this:

```
from anise.framework import engine
from anise.framework import features

@features.hookhandler(engine.HOOK_AFTER_PREPARATION)
def somefoolishinitialization():
    print("Hello World.")
```

This uses the `anise.engine.HOOK_AFTER_PREPARATION` hook, which is triggered in the initialization of the universe object. Details about the existing [hooks](#) can be found in the [Console](#). Find the hook nodes in the universe object tree and read what the info panel provides about them. Also read [... use hooks](#).

1.11.3 ... implement tasks

A task is a Python function which implements a particular automation routine. Whenever the Anise engine executes a task, it gets the implementation (the Python function) by taking a particular member of the [The Universe Object](#) and calls it.

A very easy way to add a new task (with some restrictions) is described in [... implement custom functionality](#). The following shows how to add a task to the universe object in a feature implementation:

```
from anise.framework import engine
from anise.framework import features
from anise.framework.projects import universe

def somecrazyfunction():
    print("Hello World.")

@features.hookhandler(engine.HOOK_AFTER_PREPARATION)
def somefoolishinitialization():
    # of course we can also write arbitrarily complex code here,
    # which dynamically creates tasks depending on some conditions.
    universe.addtask(somecrazyfunction)
```

This feature allows you to directly execute `mytask` (e.g. from the task chooser or from command line) on all your Anise projects.

1.11.4 ... define and access variables

The [values](#) defined in the [project description file](#) become members of the [The Universe Object](#) at runtime, as well as the defined [lets](#) indirectly assign further members.

At runtime, they can be read and modified in the very same way as you would do with any other Python object member. The `anise.framework.projects.universe` function returns the current universe object. You will see this used in virtually every part of automation code, since it is 'the' global data and control hub for the entire automation logic.

```
from anise.framework.projects import universe

def sometask():
    universe.name2 = universe.name.upper()
    print("Your project name: " + universe.name2)
```

Last but not least, you can also set (primitive) values on the [command line](#).

1.11.5 ... output or log messages

The Anise logging foundation can (and should) be used for all kinds of output. Read this example and [anise.utils.↔ logging](#) for more details:

```
from anise import utils

...
def foo():
    ...
    utils.logging.loginfo("Hello World.")
    utils.logging.loginfo("Hello World 2.", printcaller=False)
```

Note: User dialogues, which [are explained later](#), force the user attention, while logging is the right way for monologue-type communication.

1.11.6 ... load external features

For accessing stuff from an external feature, in most situations, it is enough to directly access the corresponding parts of the universe object, like this:

```
from anise.framework.projects import universe

...
def foo():
    ...
    universe.someotherfeature.someobject.foo = 42
    universe.someotherfeature.somemethod()
```

From time to time it is required to explicitly load an external feature. Early initialization phases are an example for such situations, since the universe object might be not yet populated.

You can load a feature directly by means of the Anise infrastructure and access objects in a direct way. The [anise.framework.features.loadfeature](#) function loads another feature by name and returns an [anise.framework.features.Feature](#):

```
from anise.framework import features
foo = features.loadfeature("foo").featuremodule
x = foo.some_object_in_foo
```

1.11.7 ... use hooks

The [hook system](#) is the Anise event mechanism. Customization of Anise behavior virtually always has to do with hooks in some way. An existing functionality (either something from the Anise engine, from included features or from custom features) can define and export a hook in order to provide an interaction point where new customization can plug in. It triggers the hook in some situations (i.e. it calls all registered handlers), so the external parts can act and influence the process, behavior or result of something.

In [former recipes](#) you already have seen how to use a hook by registering a handler to it. Before you can do so, you must find out which hook you want to use (e.g. from the documentation of the feature or part you want to customize or by finding it in the console). You also have to find out the calling convention: Some hooks assume that classes are registered as handlers, while others work on function. For both of them, the argument signature must be known as well (for classes this is the constructor's signature). Find out those details in the documentation, so you are able to write a handler. For some hooks, the 'Manage Hook Scripts' [console](#) action is able to create a valid skeleton.

The [anise.framework.features.hookhandler](#) function decorator is used for registering a hook handler in a feature:

```
from anise.framework import engine
from anise.framework import features
from anise.framework.projects import universe
from anise import utils

# yet another hook with trivial argument signature
@features.hookhandler(ui.HOOK_UIMODE_SELECTED)
def foo():
    utils.logging.loginfo('UI mode: ' + str(universe.ui.mode))
```

1.11.8 ... use hooks more dynamically

In some situations it is required to add hook handlers in a more dynamic way than in [... use hooks](#), i.e. depending on some conditions.

The [anise.framework.projects.Universe.addhook](#) function is used by program logic in order to add a hook handler:

```
from anise.framework.projects import universe

def myhandler():
    ...

def foo():
    ...
    if some_condition_fulfilled:
        universe.addhook(HOOK_SOMEHOOK, myhandler)
    ...
```

1.11.9 ... control the order of hook handlers

If the order of hook handlers matter, the `requires`, `provides` and `prepares` arguments of the handler insertions can be used for controlling them (see [anise.framework.features.hookhandler](#)). All those arguments expect lists of strings, which are used for describing dependencies between handlers. The engine will automatically compute a valid order on top of this.

Note: The `provides` value specifies the 'own symbols', i.e. which condition it participates to fulfill. Even if you leave it empty, it always contains the feature name and the full name of the handler function or class. This makes it easier to refer to particular elements from outside even if it does not explicitly provide any symbols.

1.11.10 ... define and trigger hooks

On the other side, some module can define a [hook](#) and trigger it in some situations in order to give external parties a way to customize the general behavior. The [anise.framework.projects.Universe.gethooks](#) function is used by program logic in order to *provide* resp. *trigger* a hook. It returns all registered handlers for execution. Definition and execution is shown in the following example:

```
from anise.framework import engine
from anise.framework import features
from anise.framework.projects import universe
from anise import utils

HOOK_MYEVENT = features.Hook()

def mytask():
    result = 0
    # note how flexible you are with using the handlers
    for handler in universe.gethooks(HOOK_MYEVENT):
        result += handler()
    utils.logging.loginfo("Result: " + str(result))

@features.hookhandler(engine.HOOK_AFTER_PREPARATION)
def foo():
    universe.addtask(mytask)
```

The handlers are typically added from external features (so we need to use [... load external features](#)) during initialization, like this:

```
from anise.framework import engine
from anise.framework import features

# we assume the feature from above available as 'foo'
foo = features.loadfeature("foo").featuremodule

@features.hookhandler(foo.HOOK_MYEVENT)
def foo():
    return 13.37
```

Note: [anise.framework.features.Hook](#) has some optional parameters, which can be used for providing additional information about your hook. None of them are required for using it, but they may help configuration guides or other auxiliary components for getting a better idea about what your hook does.

1.11.11 ... implement custom initialization steps

Use the [anise.framework.engine.HOOK_AFTER_PREPARATION](#) hook and `h_recipe8a`:

```
from anise.framework.projects import universe
from anise.framework import engine
from anise.framework import features
from anise import utils

@features.hookhandler(engine.HOOK_AFTER_PREPARATION)
def somefoolishinitialization():
    universe.somevalue = 123
    utils.logging.loginfo("Hello World.")
```

1.11.12 ... implement configuration classes and functions

The configuration of complex aspects in the [project description file](#) is often realized with custom classes carrying the specified configuration. Those can be constructed directly with nested `val` nodes in the project description file. This encapsulates complexity and allows for easier specifications in the project description file.

```
...
class Foo:
    def __init__(self, x, y):
        ...
```

can be used this way in the project description file (either as root value, as nester value or as a parameter somewhere):

```
<val t="myfeature.Foo">
  <val k="x" v="foobar"/>
  <val k="y" v="baz"/>
</val>
```

Configuration helper functions are small program routines. They are another way to provide a convenient configuration interface to some functionality for usage from within the [project description file](#). They are used by means of an [The 'let' Node](#) in order to allow complex configuration specifications. It has somewhat similar use cases as the former point. However, since the former is a class and the latter is a function, there are all kinds of technical details which make either the one or the other match better to a particular situation.

```
...
def foo(x, y):
    ...
```

can be used this way in the project description file:

```
<let k="myfeature.foo">
  <val k="x" v="foobar"/>
  <val k="y" v="baz"/>
</let>
```

1.11.13 ... interact with the user

There are some methods, which implement typical user interaction patterns. At runtime, they are available in the [The Universe Object](#) as `ui.userfeedback`. Read the documentation of [anise.features.ui.internals.UserFeedback](#) for all available methods and the following example:

```
from anise.framework.projects import universe

...
def foo():
    r = universe.ui.userfeedback.message("How are you?",
                                         btns=["Great.", "Not so good."])
    if r == 0:
        universe.ui.userfeedback.messageok("I'm glad to hear that!")
    elif r == 1:
        universe.ui.userfeedback.messageok("That's bad news :-(")
```

1.11.14 ... read and modify the project description file

Some functionality aspects - typically configuration guides - need to read and write the [project description file](#) in a program routine. There is a programming interface available for those operations in [anise.framework.projects](#).↔ [IntermediateStructure](#). The following example shows some usual operations:

```
from anise.framework.projects import universe
from anise.framework import projects

...
def foo():
    ais = projects.IntermediateStructure.fromfile(universe.specialpaths.projectfile)
    newletentry = projects.IntermediateStructure.Let("some.functionname")
    ais.addentry(newletentry)
    newvalentry = projects.IntermediateStructure.Value("bool", v=True)
    ais.addentry(newvalentry)
    newvalentry2 = projects.IntermediateStructure.Value("some.ClassName",
        foo=projects.IntermediateStructure.Value("objectref", oref="some.other.object"),
        bar=projects.IntermediateStructure.Value.fromstring("Horray."))
    ais.addentry(newvalentry2)
    ais.removeentry(newvalentry)
    newvalentry2.removearg("bar")
    ais.save()
```

1.11.15 ... implement configuration guides

The guided editors typically come somehow together with the corresponding features (although in Anise itself this relationship is realized differently). They include some graphical assistants for doing typical configuration handling.

A configuration assistant implements the [anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant](#) interface and is registered as handler for the [anise.features.projectdescriptioneditor.HOOK_GET_PROJECTDES](#)↔ [C_EDITOR_ASSISTANTS](#) hook.

They provide a filtered overview to the [The Project Description File](#) and can offer some custom actions in the overview. Those custom actions use the [user interaction methods](#) for communication with the user. They also need to [work with the project description](#).

```
from anise.framework.projects import universe
from anise.framework import features

projectdescriptioneditor = features.loadfeature("projectdescriptioneditor").featuremodule

@features.hookhandler(projectdescriptioneditor.HOOK_GET_FEATURE_ACTIONS)
class MyAssistant(projectdescriptioneditor.ProjectDescriptionEditorAssistant):

    def __init__(self):
        projectdescriptioneditor.ProjectDescriptionEditorAssistant.__init__(self,
            "Manage miracle", "Specify which miracles shall occur in your project.",
            "mymiraclefeature")

    def getcustomactions(self, ais):
        return [projectdescriptioneditor.CustomAction("Add new miracle",
            self._add_miracle)]

    def _add_miracle(self, ais):
        name = universe.ui.userfeedback.input("Please enter a name for the miracle.",
            "Peace in the world")
        ...

    def getentries(self, ais):
        r = []
        for node in ais.entries():
            if node.islet and node.key() == "mymiraclefeature.pool.add":
                r.append(e)
        return r
```

Additionally, you can register custom actions for particular kinds of nodes (instead of custom actions globally available in the overview). Those custom actions are easily accessible from the overviews and are also available in the [Raw Mode Editor](#). Implement them this way:

```

from anise.framework.projects import universe
from anise.framework import features

projectdescriptioneditor = features.loadfeature("projectdescriptioneditor").featuremodule

@features.hookhandler(projectdescriptioneditor.HOOK_GET_PROJECTDESCRIPTIONNODE_CUSTOMACTION_IMPLEMENTATIONS)
class MyNodeCustomAction(projectdescriptioneditor.CustomAction):

    def __init__(self):
        projectdescriptioneditor.CustomAction.__init__(self, "Change miracle kind")

    def visible(self, ais, node):
        return node.islet and node.key() == "mymiraclefeature.pool.add"

    def execute(self, ais, node):
        command = universe.ui.userfeedback.choice(...)
        ...

```

1.11.16 ... implement configuration guides (more flexible)

More flexible (but with more own responsibilities) than [the former recipe](#) is this interface:

```

from anise.framework.projects import universe
from anise.framework import features
from anise import utils

diagnostics = features.loadfeature("diagnostics").featuremodule
projectdescriptioneditor = features.loadfeature("projectdescriptioneditor").featuremodule

@features.hookhandler(projectdescriptioneditor.HOOK_GET_FEATURE_ACTIONS)
class MyAction(diagnostics.FeatureAction):

    def __init__(self):
        diagnostics.FeatureAction.__init__(self, "Do Something",
                                           "Does something somehow.",
                                           forfeature="myfeature")

    def execute(self):
        utils.logging.loginfo("Hello World.")

```

1.11.17 ... hide some internal stuff of my feature in the console

The console shows the complete tree of the universe object at runtime, which can decrease the visibility of the relevant objects and can weaken the clearness by means of lots of internal bookkeeping structures. In order to soften this effect, you can define all internal pieces of your feature in the 'internals' subclass, like this:

```

...

def something_important():
    ...
    internals.something_for_internal_usage(...)
    ...

class internals:

    def something_for_internal_usage(...):
        ...

```

1.11.18 ... provide some documentation for my custom stuff in the console

Use the Python docstring technique for providing documentation about functions and classes. You can use Doxygen notation in order to document function arguments.

```

def something_important(x):
    """
    This function does important things.
    @param x: Some important parameter.
    """
    ...

```

1.11.19 ... create execution scopes for reporting progress details

A program routine can often be seen as an assembly (nested or sequential) of substeps. Technically those are often encapsulated in functions, but from a high-level perspective, the degree of encapsulation can also be broader or finer.

Your program code can specify those blocks as execution scopes. The execution environment will show those blocks to the user together with progress information.

Specify execution scopes this way:

```
from anise.framework.projects import universe

def foo():
    ...
    with universe.executionscope.newsubscope("Doing something hilarious"):
        ...
    ...
```

1.11.20 ... deal with errors

The [anise.framework.exceptions](#) module contains special exception types for usage within Anise and the automation projects. It is a good idea for various reasons to consider using them in own `raise` statements. Sometimes it could also be required to use some of them in `except` clauses in order to react on a specific error situation.

1.11.21 ... work with file structures

Instances of [anise.framework.files.Filestructure](#) are typical for exchanging files or complete file structures, including subdirectories and many files. Many included features use them as arguments or return them. There are also some subclasses which are useful in some cases, like [anise.framework.files.TempDir](#).

You can create such objects this way (alternatively use one of the subclasses):

```
from anise.framework import files

def foo():
    ...
    x = files.Filestructure("/path/to/something/in/filesystem")
    ...
```

And use them by calling some of its methods, like this:

```
def foo(x): # takes a files.Filestructure
    ...
    # this copies the file structure to a destination.
    x.dl(to="/path/to/some/destination/in/filesystem")
    ...
```

1.11.22 ... hide some internal stuff of my feature in the console

In a [project description file](#), a very generic tool are [objectref values](#). They are defined by an arbitrary Python expression, which becomes evaluated in initialization when the project description file is read.

```
<?xml version="1.0" ?>
<aniseprojectdesc>
    ...
    <val k="sometask" oref="some.feature.somemethod"/>
    ...
    <val k="name" oref="'MyPrOjEcTnAmE'.lower()'"/>
    ...
</aniseprojectdesc>
```


1.11.23 ... embed Anise in other tools

Since Anise has a powerful [command-line interface](#), it is very easy to call Anise as external process from your tool.

The Anise engine can also be imported in other Python applications and can be triggered in-process. For details about the latter option, please read the source code of the [anise.framework.engine](#) module.

1.12 Appendix: Shallot Plugin

There is an Anise plugin for the Shallot file manager available under `_meta/shallot_plugin/`. It embeds some Anise functionality (like executing the tasks) directly into the file manager.

Find information about the Shallot project on the same website as Anise. The documentation of it explains how to install this plugin.

1.13 Appendix: Installation

Install Anise via the installation package for your environment, if a suitable one exists for download. This also takes care of installing dependencies and doing preparation (unless mentioned otherwise in the installation procedure). After the installation, you can skip the rest of this section.

1.13.1 Source Code Archive

Use the source code archive as fallback. Extract it to a location which is convenient to you (Windows users need an external archive program; for example the great '7-Zip' tool). Also take a look at the [Dependencies](#) for external stuff you need to install as well.

It is highly recommended to also establish a command line link or alias for [anise/anise.py](#) so you just have to type `anise` (`ln -s ...anise/anise.py /usr/local/bin/anise` on Unix or any other operating system specific way). This is according to what the installation packages do and required for executing the exact same commands as used in this manual (otherwise you must substitute the full name for the short command names in this manual).

1.14 Packages

Here are the packages with brief descriptions (if available):

anise	The Anise program	??
anise.data	Only data files - no Python code here	??
anise.data.icons	Png files	??
anise.data.icons.pde	Icons for project description editor	??
anise.data.licenses	License texts	??
anise.data.PyDepsEngine	??

anise.data.PyDepsEngine.depsengine	??
anise.features	
High level functionality provided by Anise (read documentation for more!)	??
anise.features.base	??
anise.features.build	??
anise.features.build.autotools	??
anise.features.build.make	??
anise.features.build.qmake	??
anise.features.datainjections	??
anise.features.dependencies	??
anise.features.dependencies.python	??
anise.features.diagnostics	??
anise.features.distributables	??
anise.features.documentation	??
anise.features.filesystem	??
anise.features.filetransfer	??
anise.features.gpg	??
anise.features.homepage	??
anise.features.licensing	??
anise.features.mediagalleries	??
anise.features.milieus	??
anise.features.packages	??
anise.features.packages.android	??
anise.features.packages.appc	??
anise.features.packages.debian	??
anise.features.packages.flatpak	??
anise.features.packages.python	??
anise.features.packages.win32	??
anise.features.packagestore	??
anise.features.projectdescriptioneditor	??
anise.features.projectdescriptioneditor.base	??
anise.features.projectdescriptioneditor.datainjections	??
anise.features.projectdescriptioneditor.dependencies	??
anise.features.projectdescriptioneditor.distributables	??
anise.features.projectdescriptioneditor.documentation	??
anise.features.projectdescriptioneditor.filetransfer	??
anise.features.projectdescriptioneditor.homepage	??
anise.features.projectdescriptioneditor.imagegalleries	??
anise.features.projectdescriptioneditor.licensing	??
anise.features.projectdescriptioneditor.packages	??
anise.features.projectdescriptioneditor.testing	??
anise.features.projectdescriptioneditor.versioncontrol	??
anise.features.python	??
anise.features.releasing	??
anise.features.signing	??
anise.features.testing	??
anise.features.testing.generic	??
anise.features.testing.pyunit	??
anise.features.ui	??
anise.features.ui.terminal	??
anise.features.ui.terminal.helpers	??
anise.features.ui.web	??
anise.features.ui.web.helpers	??
anise.features.ui.web.webexec	??
anise.features.versioncontrol	??
anise.features.versioncontrol.git	??
anise.features.versioncontrol.svn	??

anise.framework	
The infrastructure parts of anise	??
anise.framework.engine	
The anise engine initializes all the anise platform stuff and executes a task from a project file	??
anise.framework.exceptions	
Exception subclasses	??
anise.framework.features	
The anise plug-in system	??
anise.framework.files	
File structures, used for data exchange between methods or as result	??
anise.framework.globalvars	
Global variables which control some global aspects of anise behavior	??
anise.framework.imports	
Imports some modules for availability within <code>_projectdesc</code>	??
anise.framework.projectinformations	??
anise.framework.projects	
Implementation for the anise universe object and some services around it	??
anise.framework.report	
The Anise report facility keeps track of the steps of the execution process, the progresses and all output messages	??
anise.test	
Testing Anise; not used for production	??
anise.test.anisetest	
Anise testing backend	??
anise.test.coretests	??
anise.test.coretests.t01base	??
anise.test.coretests.t02featureloader	??
anise.test.coretests.t03hooks	??
anise.test.coretests.t04enginehooks	??
anise.test.coretests.t05hooksdependencies	??
anise.test.coretests.t06filestructure	??
anise.test.coretests.t07sourcetofilestructure	??
anise.test.coretests.t08projectvalueconvenience	??
anise.test.coretests.t09remappedfilestructure	??
anise.test.featuretests	??
anise.test.featuretests.t01homepagepackagerelease	??
anise.test.featuretests.t02homepagedependencies	??
anise.test.featuretests.t03dynamicdocumentation	??
anise.test.featuretests.t04packagestore	??
anise.test.featuretests.t05customreleasetask	??
anise.test.featuretests.t06versioncontrol	??
anise.test.featuretests.t07buildenv	??
anise.test.featuretests.t08buildmake	??
anise.test.featuretests.t09buildautotools_simple	??
anise.utils	
Utility functions from several categories which make life easier	??
anise.utils.basic	
Basic helpers for filesystem and other operating system related stuff	??
anise.utils.data	
Helpers for creation and handling of some data structures	??
anise.utils.logging	
Logging	??
anise.utils.ssh	
Utilities for ssh usage	??
test_custom_processor	??
test_custom_processor_filterobjects	??
test_custom_processor_filterobjects_completelistbyrequirements	??
test_deps_afterbeforerequired	??

test_deps_afterrequired	??
test_deps_afterrequiredoptional	??

1.15 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

<code>anise.framework.projects._UniverseProxy</code>	??
<code>anise.features.versioncontrol.svn.internals._Version</code>	??
<code>anise.features.versioncontrol.git.internals._Version</code>	??
<code>anise.features.projectdescriptioneditor.datainjections.AbstractDataInjectionGenerator</code>	??
<code>anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorC</code>	??
<code>anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorCustom</code>	??
<code>anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorPython</code>	??
<code>anise.features.projectdescriptioneditor.packages.AbstractPackage</code>	??
<code>anise.features.projectdescriptioneditor.packages.DebPackage</code>	??
<code>anise.features.projectdescriptioneditor.packages.PythonWheelPackage</code>	??
<code>anise.features.projectdescriptioneditor.packages.TarPackage</code>	??
<code>anise.features.projectdescriptioneditor.packages.Win32Package</code>	??
<code>ActionAction</code>	
<code>anise.RunAniseTaskAction</code>	??
<code>anise.framework.report.AniseOutputStream</code>	??
<code>AniseWebApplication</code>	
<code>anise.features.testing.internals.TestReportWebModule</code>	??
<code>AniseWebApplication</code>	
<code>anise.features.diagnostics.internals.ConsoleWebModule</code>	??
<code>anise.features.ui.web.webexec.WebexecApplication</code>	??
<code>Application</code>	
<code>anise.features.ui.web.helpers.AniseWebApplication</code>	??
<code>anise.features.diagnostics.internals.ChildrenInformation.Arguments</code>	??
<code>anise.features.licensing.BaseLicense</code>	??
<code>anise.features.licensing.AFLv3</code>	??
<code>anise.features.licensing.AGPLv3</code>	??
<code>anise.features.licensing.Apache2</code>	??
<code>anise.features.licensing.Artistic1</code>	??
<code>anise.features.licensing.BSD2clause</code>	??
<code>anise.features.licensing.BSD3clause</code>	??
<code>anise.features.licensing.Cc0v1</code>	??
<code>anise.features.licensing.CcBy3</code>	??
<code>anise.features.licensing.CcByNc3</code>	??
<code>anise.features.licensing.CcByNcNd3</code>	??
<code>anise.features.licensing.CcByNcSa3</code>	??
<code>anise.features.licensing.CcByNd3</code>	??
<code>anise.features.licensing.CcBySa3</code>	??
<code>anise.features.licensing.GPLv3</code>	??
<code>anise.features.licensing.GPLv3OrHigher</code>	??
<code>anise.features.licensing.LGPLv3</code>	??
<code>anise.features.licensing.MIT</code>	??
<code>anise.features.licensing.MPLv11</code>	??
<code>anise.features.licensing.MPLv2</code>	??
<code>anise.features.licensing.PublicDomain</code>	??
<code>anise.features.packages.debian.Category</code>	??
<code>anise.features.versioncontrol.ChangeLogChange</code>	??
<code>anise.features.versioncontrol.ChangeLogRevision</code>	??
<code>anise.utils.basic.ChDir</code>	??

anise.features.diagnostics.internals.ChildrenInformation	??
anise.features.ui.ChoiceTree	??
anise.features.versioncontrol.CommitMessage	??
anise.features.projectdescriptioneditor.CustomAction	??
anise.features.projectdescriptioneditor.base.ChangeHookCustomAction	??
anise.features.projectdescriptioneditor.base.EditCodeExternalEditorCustomAction	??
anise.features.projectdescriptioneditor.base.EditCodeInlineCustomAction	??
anise.features.projectdescriptioneditor.base.RenameTaskCustomAction	??
anise.features.projectdescriptioneditor.packages.DebPackageAddExecLinkCustomAction	??
anise.features.projectdescriptioneditor.packages.DebPackageAddStartMenuEntry	??
anise.features.projectdescriptioneditor.packages.PythonWheelPackageAddExecLinkCustomAction	??
anise.features.projectdescriptioneditor.packages.Win32PackageAddStartMenuEntryCustomAction	??
anise.features.projectdescriptioneditor.testing.SetReleaseCriticalCustomAction	??
anise.features.projectdescriptioneditor.testing.UnsetReleaseCriticalCustomAction	??
anise.features.dependencies.Dependency	??
anise.features.dependencies.python.DjangoDependency	??
anise.features.dependencies.python.PythonDependency	??
anise.data.PyDepsEngine.depsengine.Engine	??
anise.features.packages.appc.EnvironmentVariableDescription	??
Exception	
anise.data.PyDepsEngine.depsengine.EngineError	??
anise.data.PyDepsEngine.depsengine.CircularDependenciesError	??
anise.data.PyDepsEngine.depsengine.DependencyUnresolvedError	??
anise.framework.exceptions.AniseError	??
anise.framework.exceptions.BadCommunicationError	??
anise.framework.exceptions.BadFormatError	??
anise.framework.projects.IntermediateStructure.ParseProjectDescriptionError	??
anise.framework.exceptions.BadInputError	??
anise.framework.exceptions.ImplementationError	??
anise.framework.exceptions.InternalError	??
anise.framework.exceptions.NotImplementedError	??
anise.framework.projects.IntermediateStructure.ParseProjectDescriptionInternalError	??
anise.framework.exceptions.ProcessExecutionFailedError	??
anise.framework.exceptions.RequirementsMissingError	??
anise.framework.exceptions.UnexpectedSituationError	??
anise.framework.report.ExecutionScopePart	??
anise.framework.report.ExecutionScope	??
anise.framework.report.RootExecutionScope	??
anise.features.documentation.internals.Export	??
anise.features.documentation.HomepageExport	??
anise.features.documentation.internals.AbstractFileExport	??
anise.features.documentation.PackageExport	??
anise.features.documentation.SourceExport	??
anise.features.documentation.ExportFormat	??
anise.features.documentation.internals.Exports	??
anise.framework.features.Feature	??
anise.features.diagnostics.FeatureAction	??
anise.features.diagnostics.internals.ReadManualApplication	??
anise.features.projectdescriptioneditor.internals.EditProjectdescApplication	??
anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication	??
anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication_InBase	??
anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication	??
anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication_InBase	??
anise.features.projectdescriptioneditor.datainjections.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.dependencies.ProjectDescriptionEditorAssistant	??

anise.features.projectdescriptioneditor.distributables.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistantDevdoc	??
anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistantExports	??
anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistantPool	??
anise.features.projectdescriptioneditor.homepage.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.imagegalleries.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.licensing.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.testing.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.versioncontrol.ProjectDescriptionEditorAssistant	??
anise.features.ui.web.internals.UserFeedback.FeedbackRequest	??
anise.features.distributables.FileGroup	??
anise.framework.files.Filestructure	??
anise.features.packages.flatpak.Metafile	??
anise.features.packages.flatpak.FlatpakRef	??
anise.framework.files.TaskExecution	??
anise.features.packages.RawPackage	??
anise.framework.files.AbstractRebuildDirectoriesFilestructure	??
anise.framework.files.MergedFilestructure	??
anise.framework.files.RemappedFilestructure	??
anise.framework.files.TempDir	??
anise.framework.files.TextFileByContent	??
anise.framework.features.Hook	??
anise.framework.features.HookHandler	??
anise.framework.features.HookKind	??
anise.framework.projects.IntermediateStructure	??
anise.features.packages.python.internals	??
anise.features.dependencies.internals	??
anise.features.projectdescriptioneditor.internals	??
anise.features.python.internals	??
anise.features.diagnostics.internals	??
anise.features.releasing.internals	??
anise.features.signing.internals	??
anise.features.testing.internals	??
anise.features.mediagalleries.internals	??
anise.features.ui.terminal.internals	??
anise.features.ui.web.internals	??
anise.features.milieus.internals	??
anise.features.ui.internals	??
anise.features.distributables.internals	??
anise.features.versioncontrol.git.internals	??
anise.features.packages.android.internals	??
anise.features.versioncontrol.svn.internals	??
anise.features.base.internals	??
anise.features.packages.debian.internals	??
anise.features.versioncontrol.internals	??
anise.features.documentation.internals	??
anise.features.packages.appc.internals	??
anise.features.packages.flatpak.internals	??
anise.features.build.autotools.internals	??
anise.features.packages.win32.internals	??
anise.features.build.make.internals	??
anise.features.packages.internals	??
anise.features.packagestore.internals	??
anise.features.filetransfer.internals	??
anise.features.homepage.internals	??
anise.framework.features.internals	??
anise.features.datainjections.internals	??
anise.features.licensing.internals	??

anise.features.build.qmake.internals	??
KeyboardInterrupt	
anise.features.ui.web.webexec.WebexecApplication.AbortedInterrupt	??
anise.features.projectdescriptioneditor.filetransfer.LocalFiletransfer	??
MakeMilieuComposite	
anise.features.build.autotools.AutotoolsMilieuComposite	??
anise.features.base.Maturity	??
anise.features.mediagalleries.Media	??
anise.features.mediagalleries.MediaGallery	??
anise.features.packages.win32.MenuEntry	??
anise.features.packages.debian.MenuEntry	??
MenuEntry	
anise.features.packages.flatpak.MenuEntry	??
anise.features.milieus.Milieu	??
anise.features.milieus.DefaultMilieu	??
anise.features.packages.flatpak.FlatpakMilieu	??
anise.features.milieus.MilieuComposite	??
anise.features.build.make.MakeMilieuComposite	??
anise.features.build.qmake.QMakeMilieuComposite	??
anise.features.ui.Mode	??
anise.utils.basic.Mount	??
anise.utils.ssh.Mount	??
test_custom_processor.MyObject	??
anise.features.ui.ChoiceTree.Node	??
anise.framework.projects.IntermediateStructure.Node	??
anise.framework.projects.IntermediateStructure.Let	??
anise.framework.projects.IntermediateStructure.Value	??
anise.data.PyDepsEngine.depsengine.Processor.Object	??
anise.features.signing.OsslsigncodeConfiguration	??
anise.features.dependencies.internals.Pool	??
anise.features.mediagalleries.internals.Pool	??
anise.features.distributables.internals.Pool	??
anise.features.documentation.internals.Pool	??
anise.features.datainjections.internals.Pool	??
anise.features.packages.appc.PortDescription	??
anise.data.PyDepsEngine.depsengine.Processor	??
Processor	
test_custom_processor.MyProcessor	??
test_custom_processor_filterobjects.MyProcessor	??
test_custom_processor_filterobjects_completelistbyrequirements.MyProcessor	??
anise.framework.report.ExecutionScope.Progress	??
anise.framework.report.ExecutionScope.ProgressDeterminate	??
anise.framework.report.ExecutionScope.ProgressFailed	??
anise.framework.report.ExecutionScope.ProgressFinished	??
anise.framework.report.ExecutionScope.ProgressIndeterminate	??
anise.framework.features.PseudoFeature	??
anise.framework.projects.Universe.RegisteredTask	??
anise.features.releasing.internals.Releasetasks	??
anise.features.homepage.internals.Sections	??
anise.features.packages.debian.ServiceDescription	??
anise.features.ui.internals.SetImplicitStopAllowed	??
anise.utils.logging.Severity	??
anise.features.documentation.internals.ShortSnippets	??
SingleSelectionPanelDetail	
anise.AnisePanelDetailProjectName	??
anise.framework.engine.SpecialPaths	??
anise.features.projectdescriptioneditor.filetransfer.SshFiletransfer	??

SubmenuAction	
anise.RunAniseTasksActionMenu	??
anise.RunAniseTasksMoreActionMenu	??
TestCase	
anise.test.anisetest.AniseTestCase	??
anise.test.coretests.t01base.Test	??
anise.test.coretests.t02featureloader.Test	??
anise.test.coretests.t03hooks.Test	??
anise.test.coretests.t04enginehooks.Test	??
anise.test.coretests.t05hooksdependencies.Test	??
anise.test.coretests.t06filestructure.Test	??
anise.test.coretests.t07sourcetofilestructure.Test	??
anise.test.coretests.t08projectvalueconvenience.Test	??
anise.test.coretests.t09remappedfilestructure.Test	??
anise.test.featuretests.t01homepagepackagerelease.Test	??
anise.test.featuretests.t02homepagedependencies.Test	??
anise.test.featuretests.t03dynamicdocumentation.Test	??
anise.test.featuretests.t04packagestore.Test	??
anise.test.featuretests.t05customreleasetask.Test	??
anise.test.featuretests.t06versioncontrol.Test	??
anise.test.featuretests.t07buildenv.Test	??
anise.test.featuretests.t08buildmake.Test	??
anise.test.featuretests.t09buildautotools_simple.Test	??
test_custom_processor.Test	??
test_custom_processor_filterobjects.Test	??
test_custom_processor_filterobjects_completelistbyrequirements.Test	??
test_deps_afterbeforerequired.Test	??
test_deps_afterrequired.Test	??
test_deps_afterrequiredoptional.Test	??
anise.features.projectdescriptioneditor.testing.TestingImplementation	??
anise.features.projectdescriptioneditor.testing.AutodiscoveredPyUnitTestingImplementation	??
anise.features.projectdescriptioneditor.testing.ExternalProgramTestingImplementation	??
anise.features.projectdescriptioneditor.testing.PyUnitTestingImplementation	??
anise.features.testing.internals.TestRun	??
anise.features.testing.internals.Tests	??
anise.features.dependencies.Type	??
anise.framework.projects.Universe	??
anise.features.ui.internals.UserFeedback	??
anise.features.ui.internals.UserFeedbackProxy	??
anise.features.ui.terminal.internals.UserFeedback	??
anise.features.ui.web.internals.UserFeedback	??
anise.Utils	??
anise.features.projectdescriptioneditor.versioncontrol.Vcs	??
anise.features.versioncontrol.internals.VersionControlSystem	??
anise.features.versioncontrol.git.GitVersionControlSystem	??
anise.features.versioncontrol.internals.PseudoVersionControlSystem	??
anise.features.versioncontrol.svn.SubversionVersionControlSystem	??
anise.features.packages.python.WheelApplicationLink	??

1.16 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

anise.framework.projects._UniverseProxy	
Internally used for the universe object for allowing just to write <code>universe</code> instead of <code>universe()</code>	??

anise.features.versioncontrol.svn.internals._Version	
Version number of an Anise Subversion project	??
anise.features.versioncontrol.git.internals._Version	
Version number of an Anise Git project	??
anise.features.ui.web.webexec.WebexecApplication.AbortedInterrupt	??
anise.features.projectdescriptioneditor.datainjections.AbstractDataInjectionGenerator	??
anise.features.documentation.internals.AbstractFileExport	
Abstract base class for exports, which write documentation content to files	??
anise.features.projectdescriptioneditor.packages.AbstractPackage	??
anise.framework.files.AbstractRebuildDirectoriesFilestructure	
Abstract base class for merging directory structures together	??
anise.features.licensing.AFLv3	??
anise.features.licensing.AGPLv3	??
anise.framework.exceptions.AniseError	
An error in the execution of Anise	??
anise.framework.report.AniseOutputStream	
Proxy for terminal output/error channel, so we can record output messages	??
anise.AnisePanelDetailProjectName	??
anise.test.anisetest.AniseTestCase	??
anise.features.ui.web.helpers.AniseWebApplication	??
anise.features.licensing.Apache2	??
anise.features.diagnostics.internals.ChildrenInformation.Arguments	??
anise.features.licensing.Artistic1	??
anise.features.projectdescriptioneditor.testing.AutodiscoveredPyUnitTestingImplementation	??
anise.features.build.autotools.AutotoolsMilieuComposite	
Build environment composite for autotools builds	??
anise.framework.exceptions.BadCommunicationError	
The communication with some other component failed	??
anise.framework.exceptions.BadFormatError	
Some input is wrongly formatted	??
anise.framework.exceptions.BadInputError	
A bad value was given as some input	??
anise.features.licensing.BaseLicense	
Base class for licenses	??
anise.features.licensing.BSD2clause	??
anise.features.licensing.BSD3clause	??
anise.features.packages.debian.Category	
Enumeration for Debian's categories for software packages	??
anise.features.licensing.Cc0v1	??
anise.features.licensing.CcBy3	??
anise.features.licensing.CcByNc3	??
anise.features.licensing.CcByNcNd3	??
anise.features.licensing.CcByNcSa3	??
anise.features.licensing.CcByNd3	??
anise.features.licensing.CcBySa3	??
anise.features.projectdescriptioneditor.base.ChangeHookCustomAction	??
anise.features.versioncontrol.ChangeLogChange	
A change entry in a change log	??
anise.features.versioncontrol.ChangeLogRevision	
A revision section in a change log	??
anise.utils.basic.ChDir	
Temporarily changes the current working directory in a scope of the <code>with</code> keyword	??
anise.features.diagnostics.internals.ChildrenInformation	??
anise.features.ui.ChoiceTree	
A tree of possible choices	??
anise.data.PyDepsEngine.depsengine.CircularDependenciesError	
Exception for circular dependencies in dependency resolution	??

anise.features.versioncontrol.CommitMessage	
A commit message	??
anise.features.diagnostics.internals.ConsoleWebModule	??
anise.features.projectdescriptioneditor.CustomAction	
A custom action in a ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorC	??
anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorCustom	??
anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorPython	??
anise.features.projectdescriptioneditor.packages.DebPackage	??
anise.features.projectdescriptioneditor.packages.DebPackageAddExecLinkCustomAction	??
anise.features.projectdescriptioneditor.packages.DebPackageAddStartMenuEntry	??
anise.features.milieus.DefaultMilieu	
The default milieu	??
anise.features.dependencies.Dependency	
A dependency	??
anise.data.PyDepsEngine.depsengine.DependencyUnresolvedError	
Exception for unresolvable dependencies in dependency resolution	??
anise.features.dependencies.python.DjangoDependency	
A dependency describing the required Django framework installation	??
anise.features.projectdescriptioneditor.base.EditCodeExternalEditorCustomAction	??
anise.features.projectdescriptioneditor.base.EditCodeInlineCustomAction	??
anise.features.projectdescriptioneditor.internals.EditProjectdescApplication	
The raw mode editor	??
anise.data.PyDepsEngine.depsengine.Engine	
A dependency engine manages a list of entity objects	??
anise.data.PyDepsEngine.depsengine.EngineError	
Exceptions for all kinds of errors happening inside PyDepsEngine	??
anise.features.packages.appc.EnvironmentVariableDescription	
Description for Appc environment variables to be added to the container image	??
anise.framework.report.ExecutionScope	
An execution scope is one piece of work to do in a complete task executions	??
anise.framework.report.ExecutionScopePart	
Base class of ExecutionScope (read there for more)	??
anise.features.documentation.internals.Export	
Defines a documentation export	??
anise.features.documentation.ExportFormat	
Enumeration of output formats	??
anise.features.documentation.internals.Exports	
Storage of all documentation export definitions (i.e	??
anise.features.projectdescriptioneditor.testing.ExternalProgramTestingImplementation	??
anise.framework.features.Feature	
A feature is some encapsuled part of anise high-level functionality	??
anise.features.diagnostics.FeatureAction	
Abstract base class for an action, which can be triggered from the feature overview in the console	??
anise.features.ui.web.internals.UserFeedback.FeedbackRequest	??
anise.features.distributables.FileGroup	
A file group	??
anise.framework.files.Filestructure	
This class represents a file or a directory structure with many subdirectories and files	??
anise.features.packages.flatpak.FlatpakMilieu	
Build environment for a new Flatpak image	??
anise.features.packages.flatpak.FlatpakRef	
Description for creation of a Flatpak application reference (.flatpakref) file	??
anise.features.versioncontrol.git.GitVersionControlSystem	
Git version control system adapter	??
anise.features.licensing.GPLv3	??
anise.features.licensing.GPLv3OrHigher	??

anise.features.documentation.HomepageExport	
Describes an export of an anise documentation directly into the homepage	??
anise.framework.features.Hook	
A hook	??
anise.framework.features.HookHandler	
A hook handler bound to some hook	??
anise.framework.features.HookKind	
Enumeration of hook kinds	??
anise.framework.exceptions.ImplementationError	
A software error in an implementation occurred	??
anise.framework.projects.IntermediateStructure	
A data structure which holds an in-memory representation of the project description	??
anise.framework.exceptions.InternalError	
An internal error in the anise code occurred	??
anise.features.packages.python.internals	??
anise.features.dependencies.internals	??
anise.features.projectdescriptioneditor.internals	??
anise.features.python.internals	??
anise.features.diagnostics.internals	??
anise.features.releasing.internals	??
anise.features.signing.internals	??
anise.features.testing.internals	??
anise.features.mediagalleries.internals	??
anise.features.ui.terminal.internals	??
anise.features.ui.web.internals	??
anise.features.milieus.internals	??
anise.features.ui.internals	??
anise.features.distributables.internals	??
anise.features.versioncontrol.git.internals	??
anise.features.packages.android.internals	??
anise.features.versioncontrol.svn.internals	??
anise.features.base.internals	??
anise.features.packages.debian.internals	??
anise.features.versioncontrol.internals	??
anise.features.documentation.internals	??
anise.features.packages.appc.internals	??
anise.features.packages.flatpak.internals	??
anise.features.build.autotools.internals	??
anise.features.packages.win32.internals	??
anise.features.build.make.internals	??
anise.features.packages.internals	??
anise.features.packagestore.internals	??
anise.features.filetransfer.internals	??
anise.features.homepage.internals	??
anise.framework.features.internals	
Internal stuff	??
anise.features.datainjections.internals	??
anise.features.licensing.internals	??
anise.features.build.qmake.internals	??
anise.framework.projects.IntermediateStructure.Let	
A Let entry in the project description structure	??
anise.features.licensing.LGPLv3	??
anise.features.projectdescriptioneditor.filetransfer.LocalFiletransfer	??
anise.features.build.make.MakeMilieuComposite	
Build environment composite for make builds	??
anise.features.base.Maturity	
Enumeration describing possible project states	??

anise.features.mediagalleries.Media	One image in a gallery	??
anise.features.mediagalleries.MediaGallery	An image gallery	??
anise.features.packages.win32.MenuEntry	Specification for one start menu entry added by a Windows package installer	??
anise.features.packages.debian.MenuEntry	Specification for one menu entry added by a Debian installation package	??
anise.features.packages.flatpak.MenuEntry	Specification for one menu entry added by a Flatpak	??
anise.framework.files.MergedFilestructure	Merges many Filestructure instances together in one tree	??
anise.features.packages.flatpak.Metafile	An abstract description for a Flatpak meta file for easier access (e.g	??
anise.features.milieus.Milieu	Milieux are a functionality which allows a build task to work in different environments	??
anise.features.milieus.MilieuComposite	A milieu composite brings parts of functionality which can be stacked on top of a Milieu	??
anise.features.licensing.MIT	??
anise.features.ui.Mode	Enumeration of user interface modes	??
anise.utils.basic.Mount	Mounts a volume	??
anise.utils.ssh.Mount	Mounts a volume via ssh	??
anise.features.licensing.MPLv11	??
anise.features.licensing.MPLv2	??
test_custom_processor.MyObject	??
test_custom_processor_filterobjects_completelistbyrequirements.MyProcessor	??
test_custom_processor.MyProcessor	??
test_custom_processor_filterobjects.MyProcessor	??
anise.features.ui.ChoiceTree.Node	One node in a tree of possible choices	??
anise.framework.projects.IntermediateStructure.Node	A node in the intermediate structure	??
anise.framework.exceptions.NotImplementedError	Some unimplemented functionality was called	??
anise.data.PyDepsEngine.depsengine.Processor.Object	A processor object holds an entity object together with dependency details	??
anise.features.signing.OsslsigncodeConfiguration	Signing configuration for signing a win32 binary with the 'osslsigncode' tool	??
anise.features.documentation.PackageExport	Describes an export of an anise documentation into the distribution package(s)	??
anise.framework.projects.IntermediateStructure.ParseProjectDescriptionError	Parsing error like bad input xml	??
anise.framework.projects.IntermediateStructure.ParseProjectDescriptionInternalError	Parsing error which seems like a bug	??
anise.features.dependencies.internals.Pool	Storage of all dependencies	??
anise.features.mediagalleries.internals.Pool	Storage of all image galleries	??
anise.features.distributables.internals.Pool	Storage of all dependencies	??
anise.features.documentation.internals.Pool	Storage of all documentation text generators	??
anise.features.datainjections.internals.Pool	??
anise.features.packages.appc.PortDescription	Description for Appc network ports to be added to the container image	??

anise.framework.exceptions.ProcessExecutionFailedError	
The execution of an external process failed	??
anise.data.PyDepsEngine.depsengine.Processor	
A dependency processor manages a list of entity objects, based on top of an Engine , but sorted according to their dependencies	??
anise.framework.report.ExecutionScope.Progress	
Progress info for an execution scope	??
anise.framework.report.ExecutionScope.ProgressDeterminate	
Progress info for a determinate progress	??
anise.framework.report.ExecutionScope.ProgressFailed	
Progress info for a failed execution	??
anise.framework.report.ExecutionScope.ProgressFinished	
Progress info for a finished execution	??
anise.framework.report.ExecutionScope.ProgressIndeterminate	
Progress info for an indeterminate progress	??
anise.features.projectdescriptioneditor.homepage.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.imagegalleries.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.testing.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.versioncontrol.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.licensing.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant	
Interface class for a project description editor assistant	??
anise.features.projectdescriptioneditor.dependencies.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.datainjections.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.distributables.ProjectDescriptionEditorAssistant	??
anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistantDevdoc	??
anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistantExports	??
anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistantPool	??
anise.framework.features.PseudoFeature	
This is an auxiliary class, which fixes the gap which appears, e.g	??
anise.features.versioncontrol.internals.PseudoVersionControlSystem	
A pseudo version control system implementation which does nothing	??
anise.features.licensing.PublicDomain	??
anise.features.dependencies.python.PythonDependency	
A dependency describing the required Python interpreter	??
anise.features.projectdescriptioneditor.packages.PythonWheelPackage	??
anise.features.projectdescriptioneditor.packages.PythonWheelPackageAddExecLinkCustomAction	??
anise.features.projectdescriptioneditor.testing.PyUnitTestingImplementation	??
anise.features.build.qmake.QMakeMilieuComposite	
Build environment composite for Qt and qmake builds	??
anise.features.packages.RawPackage	
A anise.framework.files.Filestructure subclass which contains the raw package, which is the typical starting point for any further packaging steps or for fetching parts of the project file structure for some processing	??
anise.features.diagnostics.internals.ReadManualApplication	??
anise.framework.projects.Universe.RegisteredTask	
A task which is registered to the universe object	??
anise.features.releasing.internals.Releasetasks	
Storage of all actions to execute for releasing	??
anise.framework.files.RemappedFilestructure	
Remaps a Filestructure 's inner directory hierarchy into a new tree	??
anise.features.projectdescriptioneditor.base.RenameTaskCustomAction	??
anise.framework.exceptions.RequirementsMissingError	
Something required is not in place	??
anise.framework.report.RootExecutionScope	
The root execution scope	??
anise.RunAniseTaskAction	??
anise.RunAniseTasksActionMenu	??

anise.RunAniseTasksMoreActionMenu	??
anise.features.homepage.internals.Sections	
Storage of all homepage sections	??
anise.features.packages.debian.ServiceDescription	
Description for Debian services to be included in a package	??
anise.features.ui.internals.SetImplicitStopAllowed	
Controls if implicit application stop is enabled (e.g	??
anise.features.projectdescriptioneditor.testing.SetReleaseCriticalCustomAction	??
anise.utils.logging.Severity	
Enumeration of log message severities	??
anise.features.documentation.internals.ShortSnippets	
Storage of text snippets for use in the documentation texts like @a_foo	??
anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication	??
anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication_InBase	??
anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication	??
anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication_InBase	??
anise.features.documentation.SourceExport	
Describes an export of an anise documentation back into the source tree	??
anise.framework.engine.SpecialPaths	
A data structure of some particular special paths	??
anise.features.projectdescriptioneditor.filetransfer.SshFiletransfer	??
anise.features.versioncontrol.svn.SubversionVersionControlSystem	
Subversion version control system adapter	??
anise.features.projectdescriptioneditor.packages.TarPackage	??
anise.framework.files.TaskExecution	
Holds a function and some parameters for calling later on (like a delegate; or a more explicit, specialized and 'xml-compatible' version of a lambda) in order to get a Filestructure as function result	??
anise.framework.files.TempDir	
A special anise.framework.files.Filestructure which automatically creates a temporary directory in background	??
anise.test.coretests.t03hooks.Test	??
test_deps_afterbeforerequired.Test	??
anise.test.featuretests.t07buildenv.Test	??
anise.test.coretests.t02featureloader.Test	??
anise.test.featuretests.t08buildmake.Test	??
anise.test.coretests.t06filestructure.Test	??
test_custom_processor.Test	??
anise.test.featuretests.t01homepagepackagerelease.Test	??
anise.test.featuretests.t02homepagedependencies.Test	??
test_custom_processor_filterobjects_completelistbyrequirements.Test	??
anise.test.coretests.t05hooksdependencies.Test	??
anise.test.coretests.t01base.Test	??
test_custom_processor_filterobjects.Test	??
anise.test.featuretests.t05customreleasetask.Test	??
anise.test.coretests.t07sourcetofilestructure.Test	??
anise.test.featuretests.t06versioncontrol.Test	??
test_deps_afterrequired.Test	??
anise.test.coretests.t04enginehooks.Test	??
anise.test.featuretests.t04packagestore.Test	??
anise.test.coretests.t09remappedfilestructure.Test	??
anise.test.featuretests.t09buildautotools_simple.Test	??
anise.test.featuretests.t03dynamicdocumentation.Test	??
anise.test.coretests.t08projectvalueconvenience.Test	??
test_deps_afterrequiredoptional.Test	??
anise.features.projectdescriptioneditor.testing.TestingImplementation	??
anise.features.testing.internals.TestReportWebModule	
Anise web module for generating the html test report	??

anise.features.testing.internals.TestRun	
This instance keeps track of the test results	??
anise.features.testing.internals.Tests	
Storage of all tests	??
anise.framework.files.TextFileByContent	
Holds a function and some parameters for calling later on (like a delegate; or a more explicit, specialized and 'xml-compatible' version of a lambda) in order to get a Filestructure as function result	??
anise.features.dependencies.Type	
Enumeration of different types of dependencies	??
anise.framework.exceptions.UnexpectedSituationError	
An unexpected situation led to an error	??
anise.framework.projects.Universe	
Implementation for universe objects	??
anise.features.projectdescriptioneditor.testing.UnsetReleaseCriticalCustomAction	??
anise.features.ui.internals.UserFeedback	
This interface provides mechanisms for asking the user for some kinds of interaction	??
anise.features.ui.web.internals.UserFeedback	
Uses available web browser sessions for user interaction	??
anise.features.ui.terminal.internals.UserFeedback	
Uses the standard input and standard output channel of the terminal for user interaction	??
anise.features.ui.internals.UserFeedbackProxy	
This user feedback implementation forwards requests to another internal UserFeedback , depending on which mode (web, terminal) is selected	??
anise.Utils	??
anise.framework.projects.IntermediateStructure.Value	
A Value entry in the project description structure	??
anise.features.projectdescriptioneditor.versioncontrol.Vcs	??
anise.features.versioncontrol.internals.VersionControlSystem	
A version control system	??
anise.features.ui.web.webexec.WebexecApplication	??
anise.features.packages.python.WheelApplicationLink	
Specification for one program symlink added by a Python wheel package	??
anise.features.projectdescriptioneditor.packages.Win32Package	??
anise.features.projectdescriptioneditor.packages.Win32PackageAddStartMenuEntryCustomAction	??

1.17 anise Namespace Reference

The Anise program.

Namespaces

- [data](#)
 Only data files - no Python code here.
- [features](#)
 High level functionality provided by Anise (read documentation for more!).
- [framework](#)
 The infrastructure parts of anise.
- [test](#)
 Testing Anise; not used for production.
- [utils](#)
 Utility functions from several categories which make life easier.

Classes

- class [AnisePanelDetailProjectName](#)
- class [RunAniseTaskAction](#)
- class [RunAniseTasksActionMenu](#)
- class [RunAniseTasksMoreActionMenu](#)
- class [Utils](#)

Functions

- def [exec_action](#) (cls, a, b)

Variables

- [Strings](#)

1.17.1 Detailed Description

The Anise program.

1.17.2 Function Documentation

1.17.2.1 `exec_action()`

```
def anise.exec_action (
    cls,
    a,
    b )
```

1.17.3 Variable Documentation

1.17.3.1 Strings

`anise.Strings`

Initial value:

```
1 = shallot.IntlStringMap(
2   Cancel={"en": "Cancel", "de": "Abbrechen"},
3   AniseProjectName={"en": "Anise project name", "de": "Anise Projektname"},
4   ProjectNameNone={"en": "(none)", "de": "(keiner)"},
5   RunAnAniseTask={"en": "Run an Anise task", "de": "Anise-Task ausführen"},
6   TaskInParentPattern={"en": "{taskname} (in {parentname})", "de": "{taskname} (in {parentname})"},
7   TasksMore={"en": "more", "de": "mehr"},
8 )
```


1.18 anise.data Namespace Reference

Only data files - no Python code here.

Namespaces

- [icons](#)
png files.
- [licenses](#)
License texts.
- [PyDepsEngine](#)

1.18.1 Detailed Description

Only data files - no Python code here.

1.19 anise.data.icons Namespace Reference

png files.

Namespaces

- [pde](#)
icons for project description editor.

1.19.1 Detailed Description

png files.

1.20 anise.data.icons.pde Namespace Reference

icons for project description editor.

1.20.1 Detailed Description

icons for project description editor.

1.21 anise.data.licenses Namespace Reference

License texts.

1.21.1 Detailed Description

License texts.

1.22 anise.data.PyDepsEngine Namespace Reference

Namespaces

- [depsengine](#)

1.23 anise.data.PyDepsEngine.depsengine Namespace Reference

Classes

- class [CircularDependenciesError](#)
Exception for circular dependencies in dependency resolution.
- class [DependencyUnresolvedError](#)
Exception for unresolvable dependencies in dependency resolution.
- class [Engine](#)
A dependency engine manages a list of entity objects.
- class [EngineError](#)
Exceptions for all kinds of errors happening inside [PyDepsEngine](#).
- class [Processor](#)
A dependency processor manages a list of entity objects, based on top of an [Engine](#), but sorted according to their dependencies.

1.24 anise.features Namespace Reference

High level functionality provided by Anise (read documentation for more!).

Namespaces

- [base](#)
- [build](#)
- [datainjections](#)
- [dependencies](#)
- [diagnostics](#)
- [distributables](#)
- [documentation](#)
- [filesystem](#)
- [filetransfer](#)
- [gpg](#)
- [homepage](#)
- [licensing](#)
- [mediagalleries](#)
- [milieus](#)

- [packages](#)
- [packagestore](#)
- [projectdescriptioneditor](#)
- [python](#)
- [releasing](#)
- [signing](#)
- [testing](#)
- [ui](#)
- [versioncontrol](#)

1.24.1 Detailed Description

High level functionality provided by Anise (read documentation for more!).

Anise features are a plugin system. All high-level functionality of anise is implemented in one of the features which are bundled with anise. They are loaded by the feature loader [anise.framework.features](#) at runtime.

1.25 anise.features.base Namespace Reference

Classes

- class [internals](#)
- class [Maturity](#)

Enumeration describing possible project states.

Functions

- def [set_basecolor](#) (r, g, b)
Sets the project base color.
- def [pyscript](#) (code, hook, provides=None, requires=None, prepares=None)
Adds a Python function given as source code text to a hook (see Anise documentation for details about hooks).
- def [pytask](#) (code, taskname, decorators=None, label=None)
Adds a task implementation to the universe object, so it is available for execution, e.g.

1.25.1 Function Documentation

1.25.1.1 pyscript()

```
def anise.features.base.pyscript (
    code,
    hook,
    provides = None,
    requires = None,
    prepares = None )
```

Adds a Python function given as source code text to a hook (see Anise documentation for details about hooks).

Note

This call only has an effect, if this hook will be triggered afterwards. So, usage in a project description file has no effect for hooks, which are triggered in very early stages!

Parameters

<i>code</i>	The Python code block to register as a hook handler.
<i>hook</i>	The hook name (i.e. the event). This is typically a variable, which begins with <code>HOOK_</code> .
<i>provides</i>	Used for dependency ordering of hooks. A list of string symbols, which this method provides.
<i>requires</i>	Used for dependency ordering of hooks. A list of string symbols, which this method requires. Each method, which provides one of those symbols is guaranteed to be executed before.
<i>prepares</i>	Used for dependency ordering of hooks. A list of string symbols, which this method prepares. Each method, which provides one of those symbols is guaranteed to be executed afterwards.

1.25.1.2 pytask()

```
def anise.features.base.pytask (
    code,
    taskname,
    decorators = None,
    label = None )
```

Adds a task implementation to the universe object, so it is available for execution, e.g.

directly from command line or from the graphical task chooser.

Parameters

<i>code</i>	The Python code block to register as task implementation.
<i>taskname</i>	The name for this new task.
<i>decorators</i>	Additional Python function decorators to apply (as list of strings).
<i>label</i>	An optional label text.

1.25.1.3 set_basecolor()

```
def anise.features.base.set_basecolor (
    r,
    g,
    b )
```

Sets the project base color.

This color is used for all kinds of styling.

Parameters

<i>r</i>	Red value (0.0 to 1.0).
<i>g</i>	Green value (0.0 to 1.0).
<i>b</i>	Blue value (0.0 to 1.0).

1.26 anise.features.build Namespace Reference

Namespaces

- [autotools](#)
- [make](#)
- [qmake](#)

1.27 anise.features.build.autotools Namespace Reference

Classes

- class [AutotoolsMilieuComposite](#)
Build environment composite for autotools builds.
- class [internals](#)

Functions

- def [build](#) (source, milieu=None, autoreconf_args=(), configure_args=(), make_args=(), makeinstall_args=())
Builds with autotools (and autoreconf).

Variables

- [buildmake](#) = [anise.framework.features.loadfeature](#)("build.make").featuremodule

1.27.1 Function Documentation

1.27.1.1 build()

```
def anise.features.build.autotools.build (
    source,
    milieu = None,
    autoreconf_args = (),
    configure_args = (),
    make_args = (),
    makeinstall_args = () )
```

Builds with autotools (and autoreconf).

Parameters

<i>source</i>	The anise.framework.files.Filestructure to build.
<i>milieu</i>	The milieu to use. See milieus.Milieu for details.
<i>autoreconf_args</i>	Additional arguments for the 'autoreconf' call (as list of strings).
<i>configure_args</i>	Additional arguments for the './configure' call (as list of strings).
<i>make_args</i>	Additional arguments for the 'make' call (as list of strings).
<i>makeinstall_args</i>	Additional arguments for the 'make install' call (as list of strings).

Returns

: A [anise.framework.files.Filestructure](#).

1.27.2 Variable Documentation**1.27.2.1 buildmake**

```
anise.features.build.autotools.buildmake = anise.framework.features.loadfeature("build.↵
make").featuremodule
```

1.28 anise.features.build.make Namespace Reference**Classes**

- class [internals](#)
- class [MakeMilieuComposite](#)

Build environment composite for make builds.

Functions

- def [build](#) (source, milieu=None, targetname=None, targetfile=None)
Builds with make.

Variables

- [milieus](#) = [anise.framework.features.loadfeature](#)("milieus").featuremodule

1.28.1 Function Documentation**1.28.1.1 build()**

```
def anise.features.build.make.build (
    source,
    milieu = None,
    targetname = None,
    targetfile = None )
```

Builds with make.

Parameters

<i>source</i>	The anise.framework.files.Filestructure to build.
<i>milieu</i>	The milieu to use. See milieus.Milieu for details.
<i>targetname</i>	The name of the make target for creating.
<i>targetfile</i>	The file name of the target file to create.

Returns

: A [anise.framework.files.Filestructure](#).

1.28.2 Variable Documentation

1.28.2.1 milieus

```
anise.features.build.make.milieus = anise.framework.features.loadfeature("milieus").featuremodule
```

1.29 anise.features.build.qmake Namespace Reference

Classes

- class [internals](#)
- class [QMakeMilieuComposite](#)

Build environment composite for Qt and qmake builds.

Functions

- def [basedependencies](#) ()
Returns a list of dependencies, which are typical for Qt projects.
- def [build](#) (project, source, milieu=None, targetname=None, targetfile=None, qmakeoptions=None)
Builds with qmake.

Variables

- [make](#) = [anise.framework.features.loadfeature](#)("build.make").featuremodule

1.29.1 Function Documentation

1.29.1.1 basedependencies()

```
def anise.features.build.qmake.basedependencies ( )
```

Returns a list of dependencies, which are typical for Qt projects.

Returns

: List of [dependencies.Dependency](#).

1.29.1.2 build()

```
def anise.features.build.qmake.build (
    project,
    source,
    milieu = None,
    targetname = None,
    targetfile = None,
    qmakeoptions = None )
```

Builds with qmake.

Parameters

<i>project</i>	The qmake project file path relative to <code>sources</code> root.
<i>source</i>	The anise.framework.files.Filestructure to build.
<i>milieu</i>	The milieu to use. See milieus.Milieu for details.
<i>targetname</i>	The name of the make target for creating.
<i>targetfile</i>	The file name of the target file to create.
<i>qmakeoptions</i>	Additional options for running 'qmake'.

Returns

: A [anise.framework.files.Filestructure](#).

1.29.2 Variable Documentation

1.29.2.1 make

```
anise.features.build.qmake.make = anise.framework.features.loadfeature("build.make").featuremodule
```

1.30 anise.features.datainjections Namespace Reference

Classes

- class [internals](#)

Functions

- def [getessentials](#) (proprnames=None)
Returns essential project data for injecting it into some file.
- def [inject_python](#) (outfile=None, data=None, asclass=None)
Injects project data into a Python file, so it is available at program's runtime.
- def [inject_c](#) (outfile=None, data=None)
Injects project data into a C++ header file, so it is available at program's runtime.

1.30.1 Function Documentation

1.30.1.1 [getessentials\(\)](#)

```
def anise.features.datainjections.getessentials (
    proprnames = None )
```

Returns essential project data for injecting it into some file.

Parameters

<i>proprnames</i>	Additional properties to inject on top of the very fundamental ones.
-------------------	--

Returns

: a list of name/value tuples containing the data for injections.

1.30.1.2 [inject_c\(\)](#)

```
def anise.features.datainjections.inject_c (
    outfile = None,
    data = None )
```

Injects project data into a C++ header file, so it is available at program's runtime.

Parameters

<i>outfile</i>	The target file for injection.
<i>data</i>	The data to inject.

1.30.1.3 inject_python()

```
def anise.features.datainjections.inject_python (
    outfile = None,
    data = None,
    asclass = None )
```

Injects project data into a Python file, so it is available at program's runtime.

Parameters

<i>outfile</i>	The target file for injection.
<i>data</i>	The data to inject.
<i>asclass</i>	Optional class name. If set, the data is written as members into a class with this name.

1.31 anise.features.dependencies Namespace Reference

Namespaces

- [python](#)

Classes

- class [Dependency](#)
A dependency.
- class [internals](#)
- class [Type](#)
Enumeration of different types of dependencies.

1.32 anise.features.dependencies.python Namespace Reference

Classes

- class [DjangoDependency](#)
A dependency describing the required Django framework installation.
- class [PythonDependency](#)
A dependency describing the required Python interpreter.

Variables

- [dependencies](#) = `anise.framework.features.loadfeature("dependencies").featuremodule`

1.32.1 Variable Documentation

1.32.1.1 dependencies

```
anise.features.dependencies.python.dependencies = anise.framework.features.loadfeature("dependencies").feature
```

1.33 anise.features.diagnostics Namespace Reference

Classes

- class [FeatureAction](#)
Abstract base class for an action, which can be triggered from the feature overview in the console.
- class [internals](#)

Functions

- def [console](#) ()
Executes the diagnostics console.

Variables

- [ui](#) = [anise.framework.features.loadfeature\("ui"\).featuremodule](#)
- [webhelpers](#) = [anise.framework.features.loadfeature\("ui.web.helpers"\).featuremodule](#)
- [HOOK_GET_FEATURE_ACTIONS](#)

1.33.1 Function Documentation

1.33.1.1 console()

```
def anise.features.diagnostics.console ( )
```

Executes the diagnostics console.

Either call this task directly from command line or from within another task implementation in order to inspect the items of the universe object.

1.33.2 Variable Documentation

1.33.2.1 HOOK_GET_FEATURE_ACTIONS

`anise.features.diagnostics.HOOK_GET_FEATURE_ACTIONS`

Initial value:

```
1 = anise.framework.features.Hook(
2   anise.framework.features.HookKind.CLASS, wizard_classbase="_diagnostics.FeatureAction",
   wizard_classbody=, wizard_loadfeatures=["diagnostics"],
3   doc="This hook collects all implementations for feature actions, as offered in the console.",
   interesting=3)
```

1.33.2.2 ui

`anise.features.diagnostics.ui = anise.framework.features.loadfeature("ui").featuremodule`

1.33.2.3 webhelpers

`anise.features.diagnostics.webhelpers = anise.framework.features.loadfeature("ui.web.helpers").featuremodule`

1.34 anise.features.distributables Namespace Reference

Classes

- class [FileGroup](#)
A file group.
- class [internals](#)

Variables

- [HOOK_PREPARE_DISTRIBUTABLES_POOL](#)

1.34.1 Variable Documentation

1.34.1.1 HOOK_PREPARE_DISTRIBUTABLES_POOL

`anise.features.distributables.HOOK_PREPARE_DISTRIBUTABLES_POOL`

Initial value:

```
1 = anise.framework.features.Hook(
2   doc="This hook is called just before the distributables pool is prepared for further processing.")
```

1.35 anise.features.documentation Namespace Reference

Classes

- class [ExportFormat](#)
Enumeration of output formats.
- class [HomepageExport](#)
Describes an export of an anise documentation directly into the homepage.
- class [internals](#)
- class [PackageExport](#)
Describes an export of an anise documentation into the distribution package(s).
- class [SourceExport](#)
Describes an export of an anise documentation back into the source tree.

Functions

- def [has_doxygen_developerdocumentation](#) (also_include_in_readme=False, kwargs)
Configures a project for creating a Doxygen developer documentation.
- def [markdown_downgrade_headings](#) (docsrc, levels)
Rewrites a Markdown documentation for shifting the heading levels down.

1.35.1 Function Documentation

1.35.1.1 [has_doxygen_developerdocumentation\(\)](#)

```
def anise.features.documentation.has_doxygen_developerdocumentation (
    also_include_in_readme = False,
    kwargs )
```

Configures a project for creating a Doxygen developer documentation.

Parameters

<i>also_include_in_readme</i>	If a developer documentation shall be included in the README files.
<i>kwargs</i>	Additional arguments for the export definition.

1.35.1.2 [markdown_downgrade_headings\(\)](#)

```
def anise.features.documentation.markdown_downgrade_headings (
    docsrc,
    levels )
```

Rewrites a Markdown documentation for shifting the heading levels down.

Parameters

<i>docsrc</i>	The Markdown source as string.
<i>levels</i>	Number of levels to downshift the headings.

Returns

: The result Markdown documentation.

1.36 anise.features.filesystem Namespace Reference

Functions

- def [filterbyfunction](#) (acceptorfunction, source)
Filters a filestructure by a path acceptor function.
- def [filterbyextension](#) (source, exclude=None, include=None, casesensitive=False)
Filters a filestructure filename extension.
- def [filterbyname](#) (source, exclude=None, include=None, casesensitive=False)
Filters a filestructure by the file name.
- def [filterbynamepattern](#) (source, exclude=None, include=None, casesensitive=False)
Filters a filestructure by a file name pattern.
- def [subtree](#) (source, root="", copy=False)
Gets a subtree of a filestructure.

1.36.1 Function Documentation

1.36.1.1 filterbyextension()

```
def anise.features.filesystem.filterbyextension (
    source,
    exclude = None,
    include = None,
    casesensitive = False )
```

Filters a filestructure filename extension.

Parameters

<i>source</i>	An anise.framework.files.Filestructure .
<i>exclude</i>	A list of extensions to filter away.
<i>include</i>	A list of extensions to not filter away.
<i>casesensitive</i>	If the check is to be applied in a case sensitive manner.

Returns

: A filtered [anise.framework.files.Filestructure](#).

1.36.1.2 filterbyfunction()

```
def anise.features.filesystem.filterbyfunction (
    acceptorfunction,
    source )
```

Filters a filestructure by a path acceptor function.

Parameters

<i>acceptorfunction</i>	A function returning a bool for a relative file path.
<i>source</i>	An anise.framework.files.Filestructure .

Returns

: A filtered [anise.framework.files.Filestructure](#).

1.36.1.3 filterbyname()

```
def anise.features.filesystem.filterbyname (
    source,
    exclude = None,
    include = None,
    casesensitive = False )
```

Filters a filestructure by the file name.

Parameters

<i>source</i>	An anise.framework.files.Filestructure .
<i>exclude</i>	A list of file names to filter away.
<i>include</i>	A list of file names to not filter away.
<i>casesensitive</i>	If the check is to be applied in a case sensitive manner.

Returns

: A filtered [anise.framework.files.Filestructure](#).

1.36.1.4 filterbynamepattern()

```
def anise.features.filesystem.filterbynamepattern (
    source,
    exclude = None,
    include = None,
    casesensitive = False )
```

Filters a filestructure by a file name pattern.

Parameters

<i>source</i>	An anise.framework.files.Filestructure .
<i>exclude</i>	A list of file name patterns to filter away.
<i>include</i>	A list of file name patterns to not filter away.
<i>casesensitive</i>	If the check is to be applied in a case sensitive manner.

Returns

: A filtered [anise.framework.files.Filestructure](#).

1.36.1.5 subtree()

```
def anise.features.filesystem.subtree (
    source,
    root = "",
    copy = False )
```

Gets a subtree of a filestructure.

Parameters

<i>source</i>	An anise.framework.files.Filestructure .
<i>root</i>	The relative path to the new root directory.
<i>copy</i>	If the file structure is to be copied (instead of using the original tree).

Returns

: An [anise.framework.files.Filestructure](#).

1.37 anise.features.filetransfer Namespace Reference

Classes

- class [internals](#)

Functions

- def [sshupload](#) (host, username, destination, source, identityfile=None, indexpattern="{name}\", port=22, transferstrategy=None, contentdescription="content", collectlinkshook=[homepage.HOOK_UPLOAD_HOMEPA↵GE_COLLECTLINKS](#))
Deploys content to a server (e.g.
- def [localupload](#) (destination, source, indexpattern="{name}\", contentdescription="content", collectlinkshook=[homepage.↵HOOK_UPLOAD_HOMEPAGE_COLLECTLINKS](#))
Deploys content to a local place (e.g.

Variables

- [homepage](#) = [anise.framework.features.loadfeature](#)("homepage").featuremodule

1.37.1 Function Documentation

1.37.1.1 localupload()

```
def anise.features.filetransfer.localupload (
    destination,
    source,
    indexpattern = "{name}\n",
    contentdescription = "content",
    collectlinkshook = homepage.HOOK\_UPLOAD\_HOMEPAGE\_COLLECTLINKS )
```

Deploys content to a local place (e.g.

for testing).

Parameters

<i>destination</i>	The destination file path.
<i>source</i>	An anise.framework.files.Filestructure .
<i>indexpattern</i>	Pattern for index entries.
<i>contentdescription</i>	Textual description of the kind of content.
<i>collectlinkshook</i>	Which hook to trigger for collecting a list of symlinks to create/refresh after upload.

1.37.1.2 sshupload()

```
def anise.features.filetransfer.sshupload (
    host,
    username,
    destination,
    source,
```

```

identityfile = None,
indexpattern = "{name}\n",
port = 22,
transferstrategy = None,
contentdescription = "content",
collectlinkshook = homepage.HOOK_UPLOAD_HOMEPAGE_COLLECTLINKS )

```

Deploys content to a server (e.g.

a web server) via ssh.

Parameters

<i>host</i>	The server host name or address.
<i>username</i>	The login username.
<i>destination</i>	The destination file path.
<i>source</i>	An anise.framework.files.Filestructure .
<i>identityfile</i>	Path to a ssh identity file for login via public-key.
<i>indexpattern</i>	Pattern for index entries.
<i>port</i>	Port number.
<i>transferstrategy</i>	File transfer strategy.
<i>contentdescription</i>	Textual description of the kind of content.
<i>collectlinkshook</i>	Which hook to trigger for collecting a list of symlinks to create/refresh after upload.

1.37.2 Variable Documentation

1.37.2.1 homepage

```
anise.features.filetransfer.homepage = anise.framework.features.loadfeature("homepage").featuremodule
```

1.38 anise.features.gpg Namespace Reference

Functions

- def [call_gpg](#) (args, raise_on_errors="unable to call gpg", kwargs)
Calls gpg with some arguments.
- def [get_key](#) (keyid, homedir=None, witharmor=True, withbase64=True)
Returns a pgp key.

1.38.1 Function Documentation

1.38.1.1 call_gpg()

```
def anise.features.gpg.call_gpg (
    args,
    raise_on_errors = "unable to call gpg",
    kwargs )
```

Calls gpg with some arguments.

1.38.1.2 get_key()

```
def anise.features.gpg.get_key (
    keyid,
    homedir = None,
    witharmor = True,
    withbase64 = True )
```

Returns a pgp key.

Parameters

<i>keyid</i>	The key id to look for.
<i>homedir</i>	The pgp home directory.
<i>witharmor</i>	If to apply an armor to the output.
<i>withbase64</i>	If to apply additional base64 encoding to the output.

Returns

: The key as bytes.

1.39 anise.features.homepage Namespace Reference

Classes

- class [internals](#)

Functions

- def [makehomepage](#) ()
Creates the project homepage from the registered sections.

Variables

- [HOOK_UPLOAD_HOMEPAGE_COLLECTLINKS](#)

1.39.1 Function Documentation

1.39.1.1 makehomepage()

```
def anise.features.homepage.makehomepage ( )
```

Creates the project homepage from the registered sections.

Returns

: The homepage as [framework.files.Filestructure](#).

1.39.2 Variable Documentation

1.39.2.1 HOOK_UPLOAD_HOMEPAGE_COLLECTLINKS

```
anise.features.homepage.HOOK_UPLOAD_HOMEPAGE_COLLECTLINKS
```

Initial value:

```
1 = anise.framework.features.Hook(
2     wizard_functionbody="return [['sourcepath', 'destinationpath']]",
3     doc="This hook is called for collecting symlinks to deploy together with the homepage downloads.")
```

1.40 anise.features.licensing Namespace Reference

Classes

- class [AFLv3](#)
- class [AGPLv3](#)
- class [Apache2](#)
- class [Artistic1](#)
- class [BaseLicense](#)
 - Base class for licenses.*
- class [BSD2clause](#)
- class [BSD3clause](#)
- class [Cc0v1](#)
- class [CcBy3](#)
- class [CcByNc3](#)
- class [CcByNcNd3](#)
- class [CcByNcSa3](#)
- class [CcByNd3](#)
- class [CcBySa3](#)
- class [GPLv3](#)
- class [GPLv3OrHigher](#)
- class [internals](#)
- class [LGPLv3](#)
- class [MIT](#)
- class [MPLv11](#)
- class [MPLv2](#)
- class [PublicDomain](#)

Variables

- `packages` = `anise.framework.features.loadfeature("packages").featuremodule`

1.40.1 Variable Documentation

1.40.1.1 packages

`anise.features.licensing.packages` = `anise.framework.features.loadfeature("packages").featuremodule`

1.41 anise.features.mediagalleries Namespace Reference

Classes

- class `internals`
- class `Media`
One image in a gallery.
- class `MediaGallery`
An image gallery.

1.42 anise.features.milieux Namespace Reference

Classes

- class `DefaultMilieu`
The default milieu.
- class `internals`
- class `Milieu`
Milieux are a functionality which allows a build task to work in different environments.
- class `MilieuComposite`
A milieu composite brings parts of functionality which can be stacked on top of a `Milieu`.

1.43 anise.features.packages Namespace Reference

Namespaces

- `android`
- `appc`
- `debian`
- `flatpak`
- `python`
- `win32`

Classes

- class [internals](#)
- class [RawPackage](#)

A [anise.framework.files.Filestructure](#) subclass which contains the raw package, which is the typical starting point for any further packaging steps or for fetching parts of the project file structure for some processing.

Functions

- def [tarpackage](#) (source, namepostfix=None, pkgdescription="")
Builds a tarball which can be distributed.

Variables

- [ui](#) = [anise.framework.features.loadfeature](#)("ui").featuremodule
- [HOOK_ENRICHRAWPACKAGE](#)

1.43.1 Function Documentation

1.43.1.1 [tarpackage\(\)](#)

```
def anise.features.packages.tarpackage (
    source,
    namepostfix = None,
    pkgdescription = "" )
```

Builds a tarball which can be distributed.

This is mostly a source package and is possibly not the most convenient package type for an end-user.

Parameters

<i>source</i>	An anise.framework.files.Filestructure .
<i>namepostfix</i>	Postfix for the package root directory name.
<i>pkgdescription</i>	The package description text.

Returns

: A list containing new tarball as [framework.files.Filestructure](#).

1.43.2 Variable Documentation

1.43.2.1 HOOK_ENRICHRAWPACKAGE

`anise.features.packages.HOOK_ENRICHRAWPACKAGE`

Initial value:

```
1 = anise.framework.features.Hook(
2     wizard_functionparams=["destdir"], doc="This hook is called for inserting additional files to the raw
    package.")
```

1.43.2.2 ui

`anise.features.packages.ui = anise.framework.features.loadfeature("ui").featuremodule`

1.44 anise.features.packages.android Namespace Reference

Classes

- class [internals](#)

Functions

- def [package_with_ant](#) (androidpath, source, relpath="", targetname="debug")
Builds an Android package with Ant.
- def [inject_projectdata_to_androidmanifest](#) (outfile, data)
Updates some project data (like version number) in the AndroidManifest.xml.

1.44.1 Function Documentation

1.44.1.1 inject_projectdata_to_androidmanifest()

```
def anise.features.packages.android.inject_projectdata_to_androidmanifest (
    outfile,
    data )
```

Updates some project data (like version number) in the `AndroidManifest.xml`.

Parameters

<i>outfile</i>	The folder which holds the <code>AndroidManifest.xml</code> . Can be absolute or relative to the anise project directory.
<i>data</i>	Additional data to inject.

1.44.1.2 package_with_ant()

```
def anise.features.packages.android.package_with_ant (
    androidpath,
    source,
    relpath = "",
    targetname = "debug" )
```

Builds an Android package with Ant.

Parameters

<i>androidpath</i>	Full path to the <code>android</code> tool (something like <code>.../android-sdk/tools/android</code>).
<i>source</i>	An anise.framework.files.Filestructure .
<i>relpath</i>	The root path of the android project, relative to <code>sources</code> root.
<i>targetname</i>	Name of the Ant target to build.

Returns

: An [anise.framework.files.Filestructure](#).

1.45 anise.features.packages.appc Namespace Reference

Classes

- class [EnvironmentVariableDescription](#)
Description for Appc environment variables to be added to the container image.
- class [internals](#)
- class [PortDescription](#)
Description for Appc network ports to be added to the container image.

Functions

- def [appcdebianpackage](#) (debianpackage, fullname=None, ports=None, environment_variables=None, exec-command=None, run=None, run_late=None)
Builds an appc container image based on Debian.

1.45.1 Function Documentation

1.45.1.1 appcdebianpackage()

```
def anise.features.packages.appc.appcdebianpackage (
    debianpackage,
    fullname = None,
    ports = None,
    environment_variables = None,
    execcommand = None,
    run = None,
    run_late = None )
```

Builds an appc container image based on Debian.

You need an existing Debian package source.

Parameters

<i>debianpackage</i>	An anise.framework.files.Filestructure containing the debian package.
<i>fullname</i>	The full container name.
<i>ports</i>	A list of PortDescription port descriptions to add to the container image.
<i>environment_variables</i>	A list of EnvironmentVariableDescription descriptions to add to the container image.
<i>execcommand</i>	A list of strings to specify the command to execute inside the container after boot.
<i>run</i>	A list of command-line lists to execute for preparation.
<i>run_late</i>	A list of command-line lists to execute for preparation after installing the package.

Returns

: An [anise.framework.files.Filestructure](#).

1.46 anise.features.packages.debian Namespace Reference

Classes

- class [Category](#)
Enumeration for Debian's categories for software packages.
- class [internals](#)
- class [MenuEntry](#)
Specification for one menu entry added by a Debian installation package.
- class [ServiceDescription](#)
Description for Debian services to be included in a package.

Functions

- def [debpackage](#) (source, dependencies=None, executablelinks=None, menuentries=None, services=None, prerm="", postinst="", architecture="all")
Builds a Debian package.
- def [only_programfiles](#) (source)
Returns a filtered set of files for a Debian program directory (no documentation, license, tests, ...).

1.46.1 Function Documentation

1.46.1.1 `debpackage()`

```
def anise.features.packages.debian.debpackage (
    source,
    dependencies = None,
    executablelinks = None,
    menuentries = None,
    services = None,
    prerm = "",
    postinst = "",
    architecture = "all" )
```

Builds a Debian package.

Parameters

<i>source</i>	An anise.framework.files.Filestructure .
<i>dependencies</i>	A list of Debian dependencies for installation on target machines.
<i>executablelinks</i>	A dict for links to executables for deployment on target machines.
<i>menuentries</i>	A list of menu entries for creation on target machines (list of MenuEntry).
<i>services</i>	A list of services for deployment on target machines.
<i>prerm</i>	Piece of bash script executed before removal of the package on target machines.
<i>postinst</i>	Piece of bash script executed after installation of the package on the target machines.
<i>architecture</i>	The target architecture name.

Returns

: An [anise.framework.files.Filestructure](#).

1.46.1.2 `only_programfiles()`

```
def anise.features.packages.debian.only_programfiles (
    source )
```

Returns a filtered set of files for a Debian program directory (no documentation, license, tests, ...).

Parameters

<i>source</i>	An anise.framework.files.Filestructure .
---------------	--

Returns

: A filtered [anise.framework.files.Filestructure](#).

1.47 anise.features.packages.flatpak Namespace Reference

Classes

- class [FlatpakMilieu](#)
Build environment for a new Flatpak image.
- class [FlatpakRef](#)
Description for creation of a Flatpak application reference (.flatpakref) file.
- class [internals](#)
- class [MenuEntry](#)
Specification for one menu entry added by a Flatpak.
- class [Metafile](#)
An abstract description for a Flatpak meta file for easier access (e.g.

Functions

- def [flatpakpackage](#) (source, fullname=None, repository=None, repository_external_baseurl=None, metafiles=None, gpg_signkeyid="", gpg_homedir="", gpg_exportto="", milieuopts)
Builds a Flatpak and upload it to a repository.

Variables

- [milieus](#) = [anise.framework.features.loadfeature](#)("milieus").featuremodule
- [packagesdebian](#) = [anise.framework.features.loadfeature](#)("packages.debian").featuremodule

1.47.1 Function Documentation

1.47.1.1 flatpakpackage()

```
def anise.features.packages.flatpak.flatpakpackage (
    source,
    fullname = None,
    repository = None,
    repository_external_baseurl = None,
    metafiles = None,
    gpg_signkeyid = "",
    gpg_homedir = "",
    gpg_exportto = "",
    milieuopts )
```

Builds a Flatpak and upload it to a repository.

See also [FlatpakMilieu.__init__](#) for more parameters.

Parameters

<i>source</i>	An anise.framework.files.Filestructure .
<i>fullname</i>	The full image name (dotted string).

Parameters

<i>repository</i>	The repository to upload the flatpak to instead of deploying it inside the homepage data.
<i>repository_external_baseurl</i>	The external base url of the repository.
<i>metafiles</i>	Optional list of Metafile meta files to create for easier access.
<i>gpg_signkeyid</i>	The gpg key id used for signing products.
<i>gpg_homedir</i>	For gpg_sign, this optionally specifies the directory which contains that key.
<i>gpg_exportto</i>	The optional target file name for making the gpg key available for download.
<i>milieuopts</i>	Additional options for the Flatpak milieu (see FlatpakMilieu).

Returns

: An [anise.framework.files.Filestructure](#).

1.47.2 Variable Documentation

1.47.2.1 milieus

```
anise.features.packages.flatpak.milieus = anise.framework.features.loadfeature("milieus").featuremodule
```

1.47.2.2 packagesdebian

```
anise.features.packages.flatpak.packagesdebian = anise.framework.features.loadfeature("packages.↔
debian").featuremodule
```

1.48 anise.features.packages.python Namespace Reference

Classes

- class [internals](#)
- class [WheelApplicationLink](#)

Specification for one program symlink added by a Python wheel package.

Functions

- def [wheelpackage](#) (source, applicationlinks=None, requirements=None)
Builds a python wheel for distribution of Python applications.

Variables

- `packageswin32` = `anise.framework.features.loadfeature("packages.win32").featuremodule`

1.48.1 Function Documentation

1.48.1.1 `wheelpackage()`

```
def anise.features.packages.python.wheelpackage (
    source,
    applicationlinks = None,
    requirements = None )
```

Builds a python wheel for distribution of Python applications.

Parameters

<i>source</i>	An anise.framework.files.Filestructure .
<i>applicationlinks</i>	List of WheelApplicationLink .
<i>requirements</i>	List of wheel dependencies to add (list of strings).

Returns

: An [anise.framework.files.Filestructure](#).

1.48.2 Variable Documentation

1.48.2.1 `packageswin32`

```
anise.features.packages.python.packageswin32 = anise.framework.features.loadfeature("packages.↵
win32").featuremodule
```

1.49 anise.features.packages.win32 Namespace Reference

Classes

- class [internals](#)
- class [MenuEntry](#)

Specification for one start menu entry added by a Windows package installer.

Functions

- def [win32exepackage](#) (source, architecture="all", menuentries=None, projecticon=None, oninitscript="", oninstallscrip="", usemodernui2=True)
Builds a Win32 installation executable (as .exe file).
- def [deployqt5dlls](#) (milieu, binsource=None, pluginssource=None)
Returns the DLLs needed for Qt deployment on Windows.

Variables

- [HOOK_WINDOWS_PACKAGE_NSIS_ONINIT](#)
- [HOOK_WINDOWS_PACKAGE_NSIS_ONINSTALL](#)
- [HOOK_WINDOWS_PACKAGE_NSIS_GLOBAL](#)

1.49.1 Function Documentation

1.49.1.1 [deployqt5dlls\(\)](#)

```
def anise.features.packages.win32.deployqt5dlls (
    milieu,
    binsource = None,
    pluginssource = None )
```

Returns the DLLs needed for Qt deployment on Windows.

Parameters

<i>milieu</i>	The milieu to use. See milieus.Milieu for details.
<i>binsource</i>	The directory where the Qt library binaries are stored.
<i>pluginssource</i>	The directory where the Qt plugins are stored.

Returns

: A [anise.framework.files.Filestructure](#).

1.49.1.2 [win32exepackage\(\)](#)

```
def anise.features.packages.win32.win32exepackage (
    source,
    architecture = "all",
    menuentries = None,
    projecticon = None,
    oninitscript = "",
    oninstallscrip = "",
    usemodernui2 = True )
```

Builds a Win32 installation executable (as .exe file).

Parameters

<i>source</i>	An anise.framework.files.Filestructure .
<i>architecture</i>	The target architecture name.
<i>menuentries</i>	A list of menu entries for creation on target machines. See packages.win32.MenuEntry .
<i>projecticon</i>	Path to a project icon. Relative to the project root directory.
<i>oninitscript</i>	Additional NSIS on-init script.
<i>oninstallscript</i>	Additional NSIS on-install script.
<i>usemodernui2</i>	If Modern UI 2 shall be used (brings a more fresh look).

Returns

: An [anise.framework.files.Filestructure](#).

1.49.2 Variable Documentation

1.49.2.1 HOOK_WINDOWS_PACKAGE_NSIS_GLOBAL

`anise.features.packages.win32.HOOK_WINDOWS_PACKAGE_NSIS_GLOBAL`

Initial value:

```
1 = anise.framework.features.Hook(
2     doc="This hook is called for collecting additional nsis win32 installer global scripts.",
3     wizard_functionbody="return ''")
```

1.49.2.2 HOOK_WINDOWS_PACKAGE_NSIS_ONINIT

`anise.features.packages.win32.HOOK_WINDOWS_PACKAGE_NSIS_ONINIT`

Initial value:

```
1 = anise.framework.features.Hook(
2     doc="This hook is called for collecting additional nsis win32 installer oninit scripts.",
3     wizard_functionbody="return ''")
```

1.49.2.3 HOOK_WINDOWS_PACKAGE_NSIS_ONINSTALL

`anise.features.packages.win32.HOOK_WINDOWS_PACKAGE_NSIS_ONINSTALL`

Initial value:

```
1 = anise.framework.features.Hook(
2     doc="This hook is called for collecting additional nsis win32 installer oninstall scripts.",
3     wizard_functionbody="return ''")
```

1.50 anise.features.packagestore Namespace Reference

Classes

- class [internals](#)

Functions

- def [enable](#) ()
Enables using the package store.

Variables

- [distributables](#) = [anise.framework.features.loadfeature](#)("distributables").featuremodule

1.50.1 Function Documentation

1.50.1.1 [enable\(\)](#)

```
def anise.features.packagestore.enable ( )
```

Enables using the package store.

1.50.2 Variable Documentation

1.50.2.1 [distributables](#)

```
anise.features.packagestore.distributables = anise.framework.features.loadfeature("distributables").featuremod
```

1.51 anise.features.projectdescriptioneditor Namespace Reference

Namespaces

- [base](#)
- [datainjections](#)
- [dependencies](#)
- [distributables](#)
- [documentation](#)
- [filetransfer](#)
- [homepage](#)
- [imagegalleries](#)
- [licensing](#)
- [packages](#)
- [testing](#)
- [versioncontrol](#)

Classes

- class [CustomAction](#)
A custom action in a [ProjectDescriptionEditorAssistant](#).
- class [internals](#)
- class [ProjectDescriptionEditorAssistant](#)
Interface class for a project description editor assistant.

Variables

- [diagnostics](#)
- [ui](#)
- [HOOK_GET_FEATURE_ACTIONS](#)
- [HOOK_GET_PROJECTDESCRIPTIONNODE_CUSTOMACTION_IMPLEMENTATIONS](#)
- [hide](#)
- [True](#)
- [doc](#)

1.51.1 Variable Documentation

1.51.1.1 diagnostics

`anise.features.projectdescriptioneditor.diagnostics`

1.51.1.2 doc

`anise.features.projectdescriptioneditor.doc`

1.51.1.3 hide

`anise.features.projectdescriptioneditor.hide`

1.51.1.4 HOOK_GET_FEATURE_ACTIONS

`anise.features.projectdescriptioneditor.HOOK_GET_FEATURE_ACTIONS`

1.51.1.5 HOOK_GET_PROJECTDESCRIPTIONNODE_CUSTOMACTION_IMPLEMENTATIONS

```
anise.features.projectdescriptioneditor.HOOK_GET_PROJECTDESCRIPTIONNODE_CUSTOMACTION_IMPLEMENTATIONS
```

1.51.1.6 True

```
anise.features.projectdescriptioneditor.True
```

1.51.1.7 ui

```
anise.features.projectdescriptioneditor.ui
```

1.52 anise.features.projectdescriptioneditor.base Namespace Reference

Classes

- class [ChangeHookCustomAction](#)
- class [EditCodeExternalEditorCustomAction](#)
- class [EditCodeInlineCustomAction](#)
- class [RenameTaskCustomAction](#)
- class [SimpleScriptAssistantApplication](#)
- class [SimpleScriptAssistantApplication_InBase](#)
- class [SimpleTaskAssistantApplication](#)
- class [SimpleTaskAssistantApplication_InBase](#)

Functions

- def [formatcodefrom](#) (code)
- def [formatcodeto](#) (code)

Variables

- [diagnostics](#) = [anise.framework.features.loadfeature](#)("diagnostics").featuremodule
- [projectdescriptioneditor](#) = [anise.framework.features.loadfeature](#)("projectdescriptioneditor").featuremodule

1.52.1 Function Documentation

1.52.1.1 formatcodefrom()

```
def anise.features.projectdescriptioneditor.base.formatcodefrom (
    code )
```

1.52.1.2 formatcodeto()

```
def anise.features.projectdescriptioneditor.base.formatcodeto (
    code )
```

1.52.2 Variable Documentation

1.52.2.1 diagnostics

```
anise.features.projectdescriptioneditor.base.diagnostics = anise.framework.features.loadfeature("diagnostics")
```

1.52.2.2 projectdescriptioneditor

```
anise.features.projectdescriptioneditor.base.projectdescriptioneditor = anise.framework.↔
features.loadfeature("projectdescriptioneditor").featuremodule
```

1.53 anise.features.projectdescriptioneditor.datainjections Namespace Reference

Classes

- class [AbstractDataInjectionGenerator](#)
- class [DataInjectionGeneratorC](#)
- class [DataInjectionGeneratorCustom](#)
- class [DataInjectionGeneratorPython](#)
- class [ProjectDescriptionEditorAssistant](#)

Variables

- [projectdescriptioneditor](#)
- [HOOK_GET_DATAINJECTION_GENERATORS](#)
- [hide](#)
- [True](#)
- [doc](#)

1.53.1 Variable Documentation

1.53.1.1 doc

```
anise.features.projectdescriptioneditor.datainjections.doc
```

1.53.1.2 hide

```
anise.features.projectdescriptioneditor.datainjections.hide
```

1.53.1.3 HOOK_GET_DATAINJECTION_GENERATORS

```
anise.features.projectdescriptioneditor.datainjections.HOOK_GET_DATAINJECTION_GENERATORS
```

1.53.1.4 projectdescriptioneditor

```
anise.features.projectdescriptioneditor.datainjections.projectdescriptioneditor
```

1.53.1.5 True

```
anise.features.projectdescriptioneditor.datainjections.True
```

1.54 anise.features.projectdescriptioneditor.dependencies Namespace Reference

Classes

- class [ProjectDescriptionEditorAssistant](#)

Variables

- [projectdescriptioneditor](#)

1.54.1 Variable Documentation

1.54.1.1 projectdescriptioneditor

`anise.features.projectdescriptioneditor.dependencies.projectdescriptioneditor`

1.55 anise.features.projectdescriptioneditor.distributables Namespace Reference

Classes

- class [ProjectDescriptionEditorAssistant](#)

Variables

- [projectdescriptioneditor](#)
- [HOOK_GET_DISTRIBUTABLES_PROVIDERS](#)
- [hide](#)
- [True](#)
- [doc](#)

1.55.1 Variable Documentation

1.55.1.1 doc

`anise.features.projectdescriptioneditor.distributables.doc`

1.55.1.2 hide

`anise.features.projectdescriptioneditor.distributables.hide`

1.55.1.3 HOOK_GET_DISTRIBUTABLES_PROVIDERS

`anise.features.projectdescriptioneditor.distributables.HOOK_GET_DISTRIBUTABLES_PROVIDERS`

1.55.1.4 projectdescriptioneditor

`anise.features.projectdescriptioneditor.distributables.projectdescriptioneditor`

1.55.1.5 True

`anise.features.projectdescriptioneditor.distributables.True`

1.56 anise.features.projectdescriptioneditor.documentation Namespace Reference

Classes

- class [ProjectDescriptionEditorAssistantDevdoc](#)
- class [ProjectDescriptionEditorAssistantExports](#)
- class [ProjectDescriptionEditorAssistantPool](#)

Variables

- [projectdescriptioneditor](#)

1.56.1 Variable Documentation

1.56.1.1 projectdescriptioneditor

`anise.features.projectdescriptioneditor.documentation.projectdescriptioneditor`

1.57 anise.features.projectdescriptioneditor.filetransfer Namespace Reference

Classes

- class [LocalFiletransfer](#)
- class [SshFiletransfer](#)

Functions

- def [ask_new_filetransfer](#) (msg)

Variables

- [HOOK_GET_FILETRANSFER_PROVIDERS](#)
- [hide](#)
- [True](#)
- [doc](#)

1.57.1 Function Documentation

1.57.1.1 ask_new_filetransfer()

```
def anise.features.projectdescriptioneditor.filetransfer.ask_new_filetransfer (
    msg )
```

1.57.2 Variable Documentation

1.57.2.1 doc

```
anise.features.projectdescriptioneditor.filetransfer.doc
```

1.57.2.2 hide

```
anise.features.projectdescriptioneditor.filetransfer.hide
```

1.57.2.3 HOOK_GET_FILETRANSFER_PROVIDERS

```
anise.features.projectdescriptioneditor.filetransfer.HOOK_GET_FILETRANSFER_PROVIDERS
```

1.57.2.4 True

```
anise.features.projectdescriptioneditor.filetransfer.True
```

1.58 anise.features.projectdescriptioneditor.homepage Namespace Reference

Classes

- class [ProjectDescriptionEditorAssistant](#)

Variables

- [projectdescriptioneditor](#)
- [projectdescriptioneditorfiletransfer](#)

1.58.1 Variable Documentation

1.58.1.1 projectdescriptioneditor

`anise.features.projectdescriptioneditor.homepage.projectdescriptioneditor`

1.58.1.2 projectdescriptioneditorfiletransfer

`anise.features.projectdescriptioneditor.homepage.projectdescriptioneditorfiletransfer`

1.59 anise.features.projectdescriptioneditor.imagegalleries Namespace Reference

Classes

- class [ProjectDescriptionEditorAssistant](#)

Variables

- [projectdescriptioneditor](#)

1.59.1 Variable Documentation

1.59.1.1 projectdescriptioneditor

`anise.features.projectdescriptioneditor.imagegalleries.projectdescriptioneditor`

1.60 anise.features.projectdescriptioneditor.licensing Namespace Reference

Classes

- class [ProjectDescriptionEditorAssistant](#)

Variables

- [projectdescriptioneditor](#)

1.60.1 Variable Documentation

1.60.1.1 projectdescriptioneditor

`anise.features.projectdescriptioneditor.licensing.projectdescriptioneditor`

1.61 anise.features.projectdescriptioneditor.packages Namespace Reference

Classes

- class [AbstractPackage](#)
- class [DebPackage](#)
- class [DebPackageAddExecLinkCustomAction](#)
- class [DebPackageAddStartMenuEntry](#)
- class [PythonWheelPackage](#)
- class [PythonWheelPackageAddExecLinkCustomAction](#)
- class [TarPackage](#)
- class [Win32Package](#)
- class [Win32PackageAddStartMenuEntryCustomAction](#)

Variables

- [pde_distributables](#)
- [projectdescriptioneditor](#)

1.61.1 Variable Documentation

1.61.1.1 pde_distributables

`anise.features.projectdescriptioneditor.packages.pde_distributables`

1.61.1.2 projectdescriptioneditor

`anise.features.projectdescriptioneditor.packages.projectdescriptioneditor`

1.62 anise.features.projectdescriptioneditor.testing Namespace Reference

Classes

- class [AutodiscoveredPyUnitTestingImplementation](#)
- class [ExternalProgramTestingImplementation](#)
- class [ProjectDescriptionEditorAssistant](#)
- class [PyUnitTestingImplementation](#)
- class [SetReleaseCriticalCustomAction](#)
- class [TestingImplementation](#)
- class [UnsetReleaseCriticalCustomAction](#)

Variables

- [projectdescriptioneditor](#)
- [HOOK_GET_TESTING_CONFIGUIDES](#)
- [CLASS](#)
- [hide](#)
- [doc](#)

1.62.1 Variable Documentation

1.62.1.1 CLASS

`anise.features.projectdescriptioneditor.testing.CLASS`

1.62.1.2 doc

`anise.features.projectdescriptioneditor.testing.doc`

1.62.1.3 hide

`anise.features.projectdescriptioneditor.testing.hide`

1.62.1.4 HOOK_GET_TESTING_CONFIGUIDES

`anise.features.projectdescriptioneditor.testing.HOOK_GET_TESTING_CONFIGUIDES`

1.62.1.5 projectdescriptioneditor

`anise.features.projectdescriptioneditor.testing.projectdescriptioneditor`

1.63 anise.features.projectdescriptioneditor.versioncontrol Namespace Reference

Classes

- class [ProjectDescriptionEditorAssistant](#)
- class [Vcs](#)

Functions

- def [getversioncontrolimplementations](#) ()

Variables

- [projectdescriptioneditor](#)
- [HOOK_GET_VERSIONCONTROLSYSTEM_IMPLEMENTATIONS](#)
Called in order to get version control system implementations.
- [hide](#)
- [True](#)
- [doc](#)

1.63.1 Function Documentation

1.63.1.1 getversioncontrolimplementations()

```
def anise.features.projectdescriptioneditor.versioncontrol.getversioncontrolimplementations (
)
```

1.63.2 Variable Documentation

1.63.2.1 doc

`anise.features.projectdescriptioneditor.versioncontrol.doc`

1.63.2.2 hide

`anise.features.projectdescriptioneditor.versioncontrol.hide`

1.63.2.3 HOOK_GET_VERSIONCONTROLSYSTEM_IMPLEMENTATIONS

`anise.features.projectdescriptioneditor.versioncontrol.HOOK_GET_VERSIONCONTROLSYSTEM_IMPLEMENTATIONS`

Called in order to get version control system implementations.

1.63.2.4 projectdescriptioneditor

`anise.features.projectdescriptioneditor.versioncontrol.projectdescriptioneditor`

1.63.2.5 True

`anise.features.projectdescriptioneditor.versioncontrol.True`

1.64 anise.features.python Namespace Reference

Classes

- class [internals](#)

Functions

- def [is_django_project](#) (relpath="")
Configures a project to be a Django project.

1.64.1 Function Documentation

1.64.1.1 is_django_project()

```
def anise.features.python.is_django_project (
    relpath = "" )
```

Configures a project to be a Django project.

Parameters

<code>relpath</code>	Path to the Django application. Relative to the project root directory.
----------------------	---

1.65 anise.features.releasing Namespace Reference

Classes

- class [internals](#)

Variables

- [HOOK_RELEASE](#)
- [interesting](#)
- [doc](#)

1.65.1 Variable Documentation

1.65.1.1 doc

`anise.features.releasing.doc`

1.65.1.2 HOOK_RELEASE

`anise.features.releasing.HOOK_RELEASE`

1.65.1.3 interesting

`anise.features.releasing.interesting`

1.66 anise.features.signing Namespace Reference

Classes

- class [internals](#)
- class [OsslsigncodeConfiguration](#)

Signing configuration for signing a win32 binary with the 'osslsigncode' tool.

Functions

- def [signwin32_osslsigncode](#) (signconfig, source, required=True)
Signs a win32 binary with a cryptographic code-signing certificate.

1.66.1 Function Documentation

1.66.1.1 signwin32_osslsigncode()

```
def anise.features.signing.signwin32_osslsigncode (
    signconfig,
    source,
    required = True )
```

Signs a win32 binary with a cryptographic code-signing certificate.

Parameters

<i>signconfig</i>	An instance of OsslsigncodeConfiguration specifying the signing details.
<i>source</i>	An anise.framework.files.Filestructure .
<i>required</i>	If a failure is critical or can be ignored.

1.67 anise.features.testing Namespace Reference

Namespaces

- [generic](#)
- [pyunit](#)

Classes

- class [internals](#)

Functions

- def [test](#) (runtestpackage=None, runtest=None)
Runs tests.

Variables

- [releasing](#)
- [uiwebhelpers](#)

1.67.1 Function Documentation

1.67.1.1 test()

```
def anise.features.testing.test (
    runtestpackage = None,
    runtest = None )
```

Runs tests.

It either obeys runtestpackage and runtest, checks the universe, or it asks the user which tests to run.

Parameters

<i>runtestpackage</i>	Name of the test package to run ('*': all).
<i>runtest</i>	Name of the test to run ('*': all).

1.67.2 Variable Documentation

1.67.2.1 releasing

```
anise.features.testing.releasing
```

1.67.2.2 uiwebhelpers

```
anise.features.testing.uiwebhelpers
```

1.68 anise.features.testing.generic Namespace Reference

Functions

- def [generictest](#) (testrun, command)
Runs an external command as test and interpret it as successful, if the return value is 0.

Variables

- [testing](#)

1.68.1 Function Documentation

1.68.1.1 generictest()

```
def anise.features.testing.generic.generictest (
    testrun,
    command )
```

Runs an external command as test and interpret it as successful, if the return value is 0.

Parameters

<i>testrun</i>	A testing.internals.TestRun instance.
<i>command</i>	The external command (as list of strings).

1.68.2 Variable Documentation

1.68.2.1 testing

```
anise.features.testing.generic.testing
```

1.69 anise.features.testing.pyunit Namespace Reference

Functions

- def [pyunittest](#) (testrun, testclass, testdir=None)
Runs an pyunit test.
- def [pyunittests](#) (testrun, testdir=None, filepattern="*.py")
Runs all pyunit tests in the project (or all which fulfill specified patterns).

Variables

- [testinggeneric](#)

1.69.1 Function Documentation

1.69.1.1 pyunittest()

```
def anise.features.testing.pyunit.pyunittest (
    testrun,
    testclass,
    testdir = None )
```

Runs an pyunit test.

Parameters

<i>testrun</i>	A testing.internals.TestRun instance.
<i>testdir</i>	The root directory where to look for tests.
<i>testclass</i>	The name of the test class.

1.69.1.2 pyunittests()

```
def anise.features.testing.pyunit.pyunittests (
    testrun,
    testdir = None,
    filepattern = "*.py" )
```

Runs all pyunit tests in the project (or all which fulfill specified patterns).

The discovery is done automatically.

Parameters

<i>testrun</i>	A testing.internals.TestRun instance.
<i>testdir</i>	The root directory where to look for tests.
<i>filepattern</i>	A file pattern describing which files to inspect.

1.69.2 Variable Documentation

1.69.2.1 testinggeneric

```
anise.features.testing.pyunit.testinggeneric
```

1.70 anise.features.ui Namespace Reference

Namespaces

- [terminal](#)
- [web](#)

Classes

- class [ChoiceTree](#)
A tree of possible choices.
- class [internals](#)
- class [Mode](#)
Enumeration of user interface modes.

Functions

- def `chooser_hide_showall` (`hide=True`)
Sets the availability of the 'show all' button in the task chooser, which also shows the unimportant tasks.
- def `enforce_terminal_mode` ()
Force usage of terminal mode.

Variables

- `HOOK_UIMODE_SELECTED`
- `doc`
- `wizard_functionadditionallog`
- `HOOK_GET_USERFEEDBACK_IMPLEMENTATION`
- `hide`
- `HOOK_GET_ALLOWIMPLICITCLOSE_IMPLEMENTATION`

1.70.1 Function Documentation

1.70.1.1 `chooser_hide_showall()`

```
def anise.features.ui.chooser_hide_showall (
    hide = True )
```

Sets the availability of the 'show all' button in the task chooser, which also shows the unimportant tasks.

Parameters

<code>hide</code>	If the 'show all' button is to be hidden.
-------------------	---

1.70.1.2 `enforce_terminal_mode()`

```
def anise.features.ui.enforce_terminal_mode ( )
```

Force usage of terminal mode.

1.70.2 Variable Documentation

1.70.2.1 `doc`

```
anise.features.ui.doc
```

1.70.2.2 hide

`anise.features.ui.hide`

1.70.2.3 HOOK_GET_ALLOWIMPLICITCLOSE_IMPLEMENTATION

`anise.features.ui.HOOK_GET_ALLOWIMPLICITCLOSE_IMPLEMENTATION`

1.70.2.4 HOOK_GET_USERFEEDBACK_IMPLEMENTATION

`anise.features.ui.HOOK_GET_USERFEEDBACK_IMPLEMENTATION`

1.70.2.5 HOOK_UIMODE_SELECTED

`anise.features.ui.HOOK_UIMODE_SELECTED`

1.70.2.6 wizard_functionadditionallog

`anise.features.ui.wizard_functionadditionallog`

1.71 anise.features.ui.terminal Namespace Reference

Namespaces

- [helpers](#)

Classes

- class [internals](#)

Variables

- [ui](#)

1.71.1 Variable Documentation

1.71.1.1 ui

```
anise.features.ui.terminal.ui
```

1.72 anise.features.ui.terminal.helpers Namespace Reference

Functions

- def [showhead](#) (head, msg, withterm=True)
- def [prompt](#) (head, msg)

1.72.1 Function Documentation

1.72.1.1 prompt()

```
def anise.features.ui.terminal.helpers.prompt (
    head,
    msg )
```

1.72.1.2 showhead()

```
def anise.features.ui.terminal.helpers.showhead (
    head,
    msg,
    withterm = True )
```

1.73 anise.features.ui.web Namespace Reference

Namespaces

- [helpers](#)
- [webexec](#)

Classes

- class [internals](#)

Variables

- [ui](#)
- [HOOK_SEND_USERFEEDBACK_WEB_REQUEST](#)
- [hide](#)
- [True](#)
- [doc](#)
- [HOOK_GET_USERFEEDBACK_WEB_ANSWERS](#)

1.73.1 Variable Documentation

1.73.1.1 doc

`anise.features.ui.web.doc`

1.73.1.2 hide

`anise.features.ui.web.hide`

1.73.1.3 HOOK_GET_USERFEEDBACK_WEB_ANSWERS

`anise.features.ui.web.HOOK_GET_USERFEEDBACK_WEB_ANSWERS`

1.73.1.4 HOOK_SEND_USERFEEDBACK_WEB_REQUEST

`anise.features.ui.web.HOOK_SEND_USERFEEDBACK_WEB_REQUEST`

1.73.1.5 True

`anise.features.ui.web.True`

1.73.1.6 ui

`anise.features.ui.web.ui`

1.74 anise.features.ui.web.helpers Namespace Reference

Classes

- class [AniseWebApplication](#)

Variables

- [uiweb](#)

1.74.1 Variable Documentation

1.74.1.1 uiweb

`anise.features.ui.web.helpers.uiweb`

1.75 anise.features.ui.web.webexec Namespace Reference

Classes

- class [WebexecApplication](#)

Functions

- def [initproject](#) ()

Variables

- [ui](#)
- [webhelpers](#)
- [val](#)
- [o](#)

1.75.1 Function Documentation

1.75.1.1 initproject()

```
def anise.features.ui.web.webexec.initproject ( )
```

1.75.2 Variable Documentation

1.75.2.1 o

`anise.features.ui.web.webexec.o`

1.75.2.2 ui

`anise.features.ui.web.webexec.ui`

1.75.2.3 val

`anise.features.ui.web.webexec.val`

1.75.2.4 webhelpers

`anise.features.ui.web.webexec.webhelpers`

1.76 anise.features.versioncontrol Namespace Reference

Namespaces

- [git](#)
- [svn](#)

Classes

- class [ChangeLogChange](#)
A change entry in a change log.
- class [ChangeLogRevision](#)
A revision section in a change log.
- class [CommitMessage](#)
A commit message.
- class [internals](#)

Functions

- def [setvcs](#) (vcs)
Sets the version control system for this project.

Variables

- [ui](#)

1.76.1 Function Documentation

1.76.1.1 setvcs()

```
def anise.features.versioncontrol.setvcs (
    vcs )
```

Sets the version control system for this project.

Parameters

vcs	Instance of a internals.VersionControlSystem subclass, which implements the interaction with the version control system.
-----	--

1.76.2 Variable Documentation

1.76.2.1 ui

```
anise.features.versioncontrol.ui
```

1.77 anise.features.versioncontrol.git Namespace Reference

Classes

- class [GitVersionControlSystem](#)
Git version control system adapter.
- class [internals](#)

Variables

- [pdeversioncontrol](#)
- [ui](#)
- [versioncontrol](#)

1.77.1 Variable Documentation

1.77.1.1 pdeversioncontrol

`anise.features.versioncontrol.git.pdeversioncontrol`

1.77.1.2 ui

`anise.features.versioncontrol.git.ui`

1.77.1.3 versioncontrol

`anise.features.versioncontrol.git.versioncontrol`

1.78 anise.features.versioncontrol.svn Namespace Reference

Classes

- class [internals](#)
- class [SubversionVersionControlSystem](#)
Subversion version control system adapter.

Variables

- [pdeversioncontrol](#)
- [ui](#)
- [versioncontrol](#)

1.78.1 Variable Documentation

1.78.1.1 pdeversioncontrol

`anise.features.versioncontrol.svn.pdeversioncontrol`

1.78.1.2 ui

`anise.features.versioncontrol.svn.ui`

1.78.1.3 versioncontrol

`anise.features.versioncontrol.svn.versioncontrol`

1.79 anise.framework Namespace Reference

The infrastructure parts of anise.

Namespaces

- [engine](#)
The anise engine initializes all the anise platform stuff and executes a task from a project file.
- [exceptions](#)
Exception subclasses.
- [features](#)
The anise plug-in system.
- [files](#)
File structures, used for data exchange between methods or as result.
- [globalvars](#)
Global variables which control some global aspects of anise behavior.
- [imports](#)
Imports some modules for availability within `_projectdesc`.
- [projectinformations](#)
- [projects](#)
Implementation for the anise universe object and some services around it.
- [report](#)
The Anise report facility keeps track of the steps of the execution process, the progresses and all output messages.

1.79.1 Detailed Description

The infrastructure parts of anise.

The infrastructure executes Python code in a `_projectdesc` file and makes some syntactical tricks leading to shorter (and better readable) files. This tricks include the direct access to the universe object and the automatic imports it makes before executing `_projectdesc`.

In most places, the anise infrastructure tries to be as thin a layer above Python as possible instead of reinventing the wheel again and again. Although on some places anise actually has to do some more own stuff in order to bring some functionality.

Read about the individual infrastructure parts in the subpackages here.

The starting point is [anise.framework.engine](#), which is the active part responsible for executing tasks.

1.80 anise.framework.engine Namespace Reference

The anise engine initializes all the anise platform stuff and executes a task from a project file.

Classes

- class [SpecialPaths](#)
A data structure of some particular special paths.

Functions

- def [execute](#) (taskname, projectfile, projectdir, values, loadfeaturesfrom)
Executes a task.
- def [_parsevalue](#) (l)
Parses a commandline universe object value assignment like `s : name=foo`.
- def [main](#) ()
The main method when called from command line.

Variables

- [HOOK_AFTER_PREPARATION](#)
- [doc](#)
- [interesting](#)
- [HOOK_TASK_RESULT_AVAILABLE](#)
- [HOOK_SHUTDOWN](#)

1.80.1 Detailed Description

The anise engine initializes all the anise platform stuff and executes a task from a project file.

1.80.2 Function Documentation

1.80.2.1 `_parsevalue()`

```
def anise.framework.engine._parsevalue (
    l ) [private]
```

Parses a commandline universe object value assignment like `s : name=foo`.

1.80.2.2 `execute()`

```
def anise.framework.engine.execute (
    taskname,
    projectfile,
    projectdir,
    values,
    loadfeaturesfrom )
```

Executes a task.

Parameters

<i>taskname</i>	The name of the task to be executed.
<i>projectfile</i>	The path to the anise project description file. Their name is usually <code>_projectdesc</code> .
<i>projectdir</i>	The path to the project root directory.
<i>values</i>	Additional values to be set to the universe object.
<i>loadfeaturesfrom</i>	List of paths where anise features shall be loaded from.

1.80.2.3 `main()`

```
def anise.framework.engine.main ( )
```

The main method when called from command line.

Parses command line arguments and executes the chosen task.

1.80.3 Variable Documentation

1.80.3.1 doc

`anise.framework.engine.doc`

1.80.3.2 HOOK_AFTER_PREPARATION

`anise.framework.engine.HOOK_AFTER_PREPARATION`

1.80.3.3 HOOK_SHUTDOWN

`anise.framework.engine.HOOK_SHUTDOWN`

1.80.3.4 HOOK_TASK_RESULT_AVAILABLE

`anise.framework.engine.HOOK_TASK_RESULT_AVAILABLE`

1.80.3.5 interesting

`anise.framework.engine.interesting`

1.81 anise.framework.exceptions Namespace Reference

Exception subclasses.

Classes

- class [AniseError](#)
An error in the execution of Anise.
- class [BadCommunicationError](#)
The communication with some other component failed.
- class [BadFormatError](#)
Some input is wrongly formatted.
- class [BadInputError](#)
A bad value was given as some input.
- class [ImplementationError](#)
A software error in an implementation occurred.
- class [InternalError](#)
An internal error in the anise code occurred.
- class [NotImplementedError](#)
Some unimplemented functionality was called.
- class [ProcessExecutionFailedError](#)
The execution of an external process failed.
- class [RequirementsMissingError](#)
Something required is not in place.
- class [UnexpectedSituationError](#)
An unexpected situation led to an error.

1.81.1 Detailed Description

Exception subclasses.

1.82 anise.framework.features Namespace Reference

The anise plug-in system.

Classes

- class [Feature](#)
A feature is some encapsuled part of anise high-level functionality.
- class [Hook](#)
A hook.
- class [HookHandler](#)
A hook handler bound to some hook.
- class [HookKind](#)
Enumeration of hook kinds.
- class [internals](#)
Internal stuff.
- class [PseudoFeature](#)
This is an auxiliary class, which fixes the gap which appears, e.g.

Functions

- def [loadfeature](#) (name)
Load an [anise.framework.features.Feature](#) by name from the predefined sources (as optional command-line argument).
- def [hookhandler](#) (name, provides=None, requires=None, prepares=None)
Used by a feature module to register a handler for a hook globally.
- def [getfeatures](#) ()
Returns a list of all features (as [Feature](#)).

Variables

- [_features](#)
- [_featuresources](#)
- [_hooks](#)
- [HOOK_BEFORE_DEFINITION](#)
This hook gets executed by the anise engine right before the `_projectdesc` file is loaded.
- [doc](#)
- [HOOK_BEFORE_EXECUTION](#)
This hook gets executed by the anise engine after the `_projectdesc` file is loaded and right before the chosen task itself is called.

1.82.1 Detailed Description

The anise plug-in system.

This package does not include the anise high-level features itself. You would find them in [anise.features](#). This package is the part of the anise infrastructure which loads and manages those features.

For each anise run, it automatically loads the built-in [anise.features](#) as well as custom features from `~/.anise/features` and `/var/lib/anise/features`. Due to their locations, the latter ones are useful for per-machine and per-user customization.

In an arbitrary anise execution context, you may call [anise.framework.features.loadfeature](#) in order to access functionality from other features, like here:

```
from anise.framework import features
_somefeature = features.loadfeature("somefeature").featuremodule
_somefeature.some_method()
x = _somefeature.some_variable
```

1.82.2 Function Documentation

1.82.2.1 `getfeatures()`

```
def anise.framework.features.getfeatures ( )
```

Returns a list of all features (as [Feature](#)).

1.82.2.2 `hookhandler()`

```
def anise.framework.features.hookhandler (
    name,
    provides = None,
    requires = None,
    prepares = None )
```

Used by a feature module to register a handler for a hook globally.

The result should be used as Python function decorator.

Parameters

<i>name</i>	The hook which you want to register a handler for.
<i>provides</i>	A list of symbol names this hook participates to provide (dependency provider).
<i>requires</i>	A list of symbol names this hook requires to be executed before.
<i>prepares</i>	A list of symbol names this hook must prepend.

1.82.2.3 loadfeature()

```
def anise.framework.features.loadfeature (
    name )
```

Load an [anise.framework.features.Feature](#) by name from the predefined sources (as optional command-line argument).

If it is already loaded, it returns the loaded feature again.

1.82.3 Variable Documentation

1.82.3.1 _features

```
anise.framework.features._features [private]
```

1.82.3.2 _featuresources

```
anise.framework.features._featuresources [private]
```

1.82.3.3 _hooks

```
anise.framework.features._hooks [private]
```

1.82.3.4 doc

```
anise.framework.features.doc
```

1.82.3.5 HOOK_BEFORE_DEFINITION

```
anise.framework.features.HOOK_BEFORE_DEFINITION
```

This hook gets executed by the anise engine right before the `_projectdesc` file is loaded.

1.82.3.6 HOOK_BEFORE_EXECUTION

```
anise.framework.features.HOOK_BEFORE_EXECUTION
```

This hook gets executed by the anise engine after the `_projectdesc` file is loaded and right before the chosen task itself is called.

At this time all the assignments and global statements in the `_projectdesc` are already applied.

1.83 anise.framework.files Namespace Reference

File structures, used for data exchange between methods or as result.

Classes

- class [AbstractRebuildDirectoriesFilestructure](#)
Abstract base class for merging directory structures together.
- class [Filestructure](#)
This class represents a file or a directory structure with many subdirectories and files.
- class [MergedFilestructure](#)
Merges many [Filestructure](#) instances together in one tree.
- class [RemappedFilestructure](#)
Remaps a [Filestructure](#)'s inner directory hierarchy into a new tree.
- class [TaskExecution](#)
Holds a function and some parameters for calling later on (like a delegate; or a more explicit, specialized and 'xml-compatible' version of a lambda) in order to get a [Filestructure](#) as function result.
- class [TempDir](#)
A special [anise.framework.files.Filestructure](#) which automatically creates a temporary directory in background.
- class [TextFileByContent](#)
Holds a function and some parameters for calling later on (like a delegate; or a more explicit, specialized and 'xml-compatible' version of a lambda) in order to get a [Filestructure](#) as function result.

1.83.1 Detailed Description

File structures, used for data exchange between methods or as result.

1.84 anise.framework.globalvars Namespace Reference

Global variables which control some global aspects of anise behavior.

Variables

- [logdebug](#)
Log debugging stuff as well?
- [removetempdirs](#)
Remove all the temporary dirs with parts of the results after the run?

1.84.1 Detailed Description

Global variables which control some global aspects of anise behavior.

They are often influenced by command line parameters.

1.84.2 Variable Documentation

1.84.2.1 logdebug

```
anise.framework.globalvars.logdebug
```

Log debugging stuff as well?

1.84.2.2 removetempdirs

```
anise.framework.globalvars.removetempdirs
```

Remove all the temporary dirs with parts of the results after the run?

1.85 `anise.framework.imports` Namespace Reference

Imports some modules for availability within `_projectdesc`.

1.85.1 Detailed Description

Imports some modules for availability within `_projectdesc`.

Used internally.

1.86 `anise.framework.projectinformations` Namespace Reference

Variables

- [version](#)
- [buildtimeutc](#)
- [homepage](#)

1.86.1 Variable Documentation

1.86.1.1 buildtimeutc

`anise.framework.projectinformations.buildtimeutc`

1.86.1.2 homepage

`anise.framework.projectinformations.homepage`

1.86.1.3 version

`anise.framework.projectinformations.version`

1.87 anise.framework.projects Namespace Reference

Implementation for the anise universe object and some services around it.

Classes

- class [_UniverseProxy](#)
Internally used for the universe object for allowing just to write `universe` instead of `universe()`.
- class [IntermediateStructure](#)
A data structure which holds an in-memory representation of the project description.
- class [Universe](#)
Implementation for universe objects.

Functions

- def [setcurrentuniverse](#) (project)
Sets the current universe object.
- def [unsetcurrentuniverse](#) ()
Unsets the current universe object.
- def [getvalue](#) (key, defaultval=None)
A convenience method returning the value of a certain property for the current universe object.
- def [setvalue](#) (key, value)
A convenience method setting the value of a certain property for the current universe object.
- def [currentuniverse](#) ()
Returns the current universe object.

Variables

- [_currentuniverse](#)
- [universe](#)

1.87.1 Detailed Description

Implementation for the anise universe object and some services around it.

1.87.2 Function Documentation

1.87.2.1 `currentuniverse()`

```
def anise.framework.projects.currentuniverse ( )
```

Returns the current universe object.

1.87.2.2 `getvalue()`

```
def anise.framework.projects.getvalue (
    key,
    defaultval = None )
```

A convenience method returning the value of a certain property for the current universe object.

1.87.2.3 `setcurrentuniverse()`

```
def anise.framework.projects.setcurrentuniverse (
    project )
```

Sets the current universe object.

This is part of a particular piece of internal infrastructure and is typically not required to be used.

1.87.2.4 setvalue()

```
def anise.framework.projects.setvalue (
    key,
    value )
```

A convenience method setting the value of a certain property for the current universe object.

1.87.2.5 unsetcurrentuniverse()

```
def anise.framework.projects.unsetcurrentuniverse ( )
```

Unsets the current universe object.

This is part of a particular piece of internal infrastructure and is typically not required to be used.

1.87.3 Variable Documentation

1.87.3.1 _currentuniverse

```
anise.framework.projects._currentuniverse [private]
```

1.87.3.2 universe

```
anise.framework.projects.universe
```

1.88 anise.framework.report Namespace Reference

The Anise report facility keeps track of the steps of the execution process, the progresses and all output messages.

Classes

- class [AniseOutputStream](#)
Proxy for terminal output/error channel, so we can record output messages.
- class [ExecutionScope](#)
An execution scope is one piece of work to do in a complete task executions.
- class [ExecutionScopePart](#)
Base class of [ExecutionScope](#) (read there for more).
- class [RootExecutionScope](#)
The root execution scope.

Functions

- `def install_proxies ()`
Installs stdout/stderr proxies, so we can record all output made.
- `def _logmessage (text, severity, terminalonly, tofile, textescaper)`
A message logger implementation which also records the logging output to the report.

Variables

- `HOOK_EXECUTION_SCOPEPART_ADDED`
- `doc`
- `wizard_functionparams`
- `rootExecutionScope`

1.88.1 Detailed Description

The Anise report facility keeps track of the steps of the execution process, the progresses and all output messages.

1.88.2 Function Documentation

1.88.2.1 _logmessage()

```
def anise.framework.report._logmessage (  
    text,  
    severity,  
    terminalonly,  
    tofile,  
    textescaper ) [private]
```

A message logger implementation which also records the logging output to the report.

1.88.2.2 install_proxies()

```
def anise.framework.report.install_proxies ( )
```

Installs stdout/stderr proxies, so we can record all output made.

1.88.3 Variable Documentation

1.88.3.1 doc

`anise.framework.report.doc`

1.88.3.2 HOOK_EXECUTION_SCOPEPART_ADDED

`anise.framework.report.HOOK_EXECUTION_SCOPEPART_ADDED`

1.88.3.3 rootExecutionScope

`anise.framework.report.rootExecutionScope`

1.88.3.4 wizard_functionparams

`anise.framework.report.wizard_functionparams`

1.89 anise.test Namespace Reference

Testing Anise; not used for production.

Namespaces

- [anisetest](#)
Anise testing backend.
- [coretests](#)
- [featuretests](#)

1.89.1 Detailed Description

Testing Anise; not used for production.

1.90 anise.test.anisetest Namespace Reference

Anise testing backend.

Classes

- class [AniseTestCase](#)

1.90.1 Detailed Description

Anise testing backend.

1.91 anise.test.coretests Namespace Reference

Namespaces

- [t01base](#)
- [t02featureloader](#)
- [t03hooks](#)
- [t04enginehooks](#)
- [t05hooksdependencies](#)
- [t06filestructure](#)
- [t07sourcetofilestructure](#)
- [t08projectvalueconvenience](#)
- [t09remappedfilestructure](#)

1.92 anise.test.coretests.t01base Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)

1.92.1 Variable Documentation

1.92.1.1 _projectdesc

`anise.test.coretests.t01base._projectdesc` [private]

1.93 anise.test.coretests.t02featureloader Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)
- [_myfeature](#)

1.93.1 Variable Documentation

1.93.1.1 [_myfeature](#)

`anise.test.coretests.t02featureloader._myfeature` [private]

1.93.1.2 [_projectdesc](#)

`anise.test.coretests.t02featureloader._projectdesc` [private]

1.94 anise.test.coretests.t03hooks Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)
- [_myfeature](#)

1.94.1 Variable Documentation

1.94.1.1 [_myfeature](#)

`anise.test.coretests.t03hooks._myfeature` [private]

1.94.1.2 [_projectdesc](#)

`anise.test.coretests.t03hooks._projectdesc` [private]

1.95 anise.test.coretests.t04enginehooks Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)
- [_myfeature](#)

1.95.1 Variable Documentation

1.95.1.1 [_myfeature](#)

`anise.test.coretests.t04enginehooks._myfeature` [private]

1.95.1.2 [_projectdesc](#)

`anise.test.coretests.t04enginehooks._projectdesc` [private]

1.96 anise.test.coretests.t05hooksdependencies Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)
- [_myfeature](#)

1.96.1 Variable Documentation

1.96.1.1 `_myfeature`

`anise.test.coretests.t05hooksdependencies._myfeature` [private]

1.96.1.2 `_projectdesc`

`anise.test.coretests.t05hooksdependencies._projectdesc` [private]

1.97 anise.test.coretests.t06filestructure Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)

1.97.1 Variable Documentation

1.97.1.1 `_projectdesc`

`anise.test.coretests.t06filestructure._projectdesc` [private]

1.98 anise.test.coretests.t07sourcetofilestructure Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)

1.98.1 Variable Documentation

1.98.1.1 `_projectdesc`

`anise.test.coretests.t07sourcetofilestructure._projectdesc` [private]

1.99 `anise.test.coretests.t08projectvalueconvenience` Namespace Reference

Classes

- class `Test`

Variables

- `_projectdesc`

1.99.1 Variable Documentation

1.99.1.1 `_projectdesc`

`anise.test.coretests.t08projectvalueconvenience._projectdesc` [private]

1.100 `anise.test.coretests.t09remappedfilestructure` Namespace Reference

Classes

- class `Test`

Variables

- `_projectdesc`

1.100.1 Variable Documentation

1.100.1.1 `_projectdesc`

`anise.test.coretests.t09remappedfilestructure._projectdesc` [private]

1.101 anise.test.featuretests Namespace Reference

Namespaces

- [t01homepagepackagerelease](#)
- [t02homepagedependencies](#)
- [t03dynamicdocumentation](#)
- [t04packagestore](#)
- [t05customreleasetask](#)
- [t06versioncontrol](#)
- [t07buildenv](#)
- [t08buildmake](#)
- [t09buildautotools_simple](#)

1.102 anise.test.featuretests.t01homepagepackagerelease Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)

1.102.1 Variable Documentation

1.102.1.1 [_projectdesc](#)

`anise.test.featuretests.t01homepagepackagerelease._projectdesc` [private]

1.103 anise.test.featuretests.t02homepagedependencies Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)

1.103.1 Variable Documentation

1.103.1.1 `_projectdesc`

```
anise.test.featuretests.t02homepagedependencies._projectdesc [private]
```

1.104 `anise.test.featuretests.t03dynamicdocumentation` Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)

1.104.1 Variable Documentation

1.104.1.1 `_projectdesc`

```
anise.test.featuretests.t03dynamicdocumentation._projectdesc [private]
```

1.105 `anise.test.featuretests.t04packagestore` Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)
- [_myfeature](#)

1.105.1 Variable Documentation

1.105.1.1 `_myfeature`

`anise.test.featuretests.t04packagestore._myfeature` [private]

1.105.1.2 `_projectdesc`

`anise.test.featuretests.t04packagestore._projectdesc` [private]

1.106 anise.test.featuretests.t05customreleasetask Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)
- [_myfeature](#)

1.106.1 Variable Documentation

1.106.1.1 `_myfeature`

`anise.test.featuretests.t05customreleasetask._myfeature` [private]

1.106.1.2 `_projectdesc`

`anise.test.featuretests.t05customreleasetask._projectdesc` [private]

1.107 anise.test.featuretests.t06versioncontrol Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)
- [_myfeature](#)

1.107.1 Variable Documentation

1.107.1.1 [_myfeature](#)

```
anise.test.featuretests.t06versioncontrol._myfeature [private]
```

1.107.1.2 [_projectdesc](#)

```
anise.test.featuretests.t06versioncontrol._projectdesc [private]
```

1.108 [anise.test.featuretests.t07buildenv](#) Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)
- [_myfeature](#)

1.108.1 Variable Documentation

1.108.1.1 [_myfeature](#)

```
anise.test.featuretests.t07buildenv._myfeature [private]
```

1.108.1.2 [_projectdesc](#)

```
anise.test.featuretests.t07buildenv._projectdesc [private]
```


1.109 anise.test.featuretests.t08buildmake Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)
- [_myfeature](#)

1.109.1 Variable Documentation

1.109.1.1 _myfeature

`anise.test.featuretests.t08buildmake._myfeature` [private]

1.109.1.2 _projectdesc

`anise.test.featuretests.t08buildmake._projectdesc` [private]

1.110 anise.test.featuretests.t09buildautotools_simple Namespace Reference

Classes

- class [Test](#)

Variables

- [_projectdesc](#)
- [_myfeature](#)

1.110.1 Variable Documentation

1.110.1.1 `_myfeature`

```
anise.test.featuretests.t09buildautotools_simple._myfeature [private]
```

1.110.1.2 `_projectdesc`

```
anise.test.featuretests.t09buildautotools_simple._projectdesc [private]
```

1.111 `anise.utils` Namespace Reference

Utility functions from several categories which make life easier.

Namespaces

- [basic](#)
Basic helpers for filesystem and other operating system related stuff.
- [data](#)
Helpers for creation and handling of some data structures.
- [logging](#)
Logging.
- [ssh](#)
Utilities for ssh usage.

1.111.1 Detailed Description

Utility functions from several categories which make life easier.

All stuff here is not tightly bound to anise infrastructure but can easily be understood separately. Most stuff here is used by deeper parts of anise.

1.112 `anise.utils.basic` Namespace Reference

Basic helpers for filesystem and other operating system related stuff.

Classes

- class [ChDir](#)
Temporarily changes the current working directory in a scope of the `with` keyword.
- class [Mount](#)
Mounts a volume.

Functions

- def `writetofile` (filename, content)
Writes some content to a file.
- def `readfromfile` (filename, onestring=True, notexist_default=`_throw`, binary=False)
Reads the content from a file.
- def `makedirs` (dname)
Creates a directory path recursively if it does not yet exist.
- def `maketarball` (rootdir, tarname)
Makes a tar archive from a given directory structure.
- def `call` (cmdline, shell=False, decode=True, raise_on_errors=None, split_output_channels=False)
Executes a command line.
- def `callanise` (args)
Calls the anise process (in a platform independent way).
- def `execwithretry` (fct, args, maxtries=10, delaysecs=0.2, kwargs)
Executes a function with a retry mechanism (executing it again when an Exception was raised).
- def `openurl` (url)
Opens an URL (or file path) in the operating system's default way.
- def `openfile` (filepath)
Opens a file path in the operating system's default way.
- def `getdriveletters` ()
Returns a list of drive letters, which exist on this system (only win32).

Variables

- `_throw`

1.112.1 Detailed Description

Basic helpers for filesystem and other operating system related stuff.

1.112.2 Function Documentation

1.112.2.1 `call()`

```
def anise.utils.basic.call (
    cmdline,
    shell = False,
    decode = True,
    raise_on_errors = None,
    split_output_channels = False )
```

Executes a command line.

Parameters

<i>cmdline</i>	The command line as list of strings.
<i>shell</i>	If the command shall be interpreted by a shell (only the first part of cmdline will apply!).
<i>decode</i>	If the result is to be decoded to a string.
<i>raise_on_errors</i>	If errors shall raise Exceptions (also specifies the exception text in this case).
<i>split_output_channels</i>	If to separately record output and error stream.

Returns

: Tuple of returncode, program output.

1.112.2.2 callanise()

```
def anise.utils.basic.callanise (
    args )
```

Calls the anise process (in a platform independent way).

Parameters

<i>args</i>	The command line arguments to pass.
-------------	-------------------------------------

1.112.2.3 execwithretry()

```
def anise.utils.basic.execwithretry (
    fct,
    args,
    maxtries = 10,
    delaysecs = 0.2,
    kwargs )
```

Executes a function with a retry mechanism (executing it again when an Exception was raised).

Parameters

<i>fct</i>	The function to execute.
<i>args</i>	Arguments for the function call.
<i>maxtries</i>	Maximum try count.
<i>delaysecs</i>	Delay (in seconds) between tries.
<i>kwargs</i>	Keyword arguments for the function call.

1.112.2.4 getdriveletters()

```
def anise.utils.basic.getdriveletters ( )
```

Returns a list of drive letters, which exist on this system (only win32).

1.112.2.5 maketarball()

```
def anise.utils.basic.maketarball (
    rootdir,
    tarname )
```

Makes a tar archive from a given directory structure.

Parameters

<i>rootdir</i>	The root directory of the new tarball (becomes a directory in the archive as typical for tarballs).
<i>tarname</i>	The destination path for the tar archive.

1.112.2.6 mkdirs()

```
def anise.utils.basic.mkdirs (
    dname )
```

Creates a directory path recursively if it does not yet exist.

Parameters

<i>dname</i>	The directory path.
--------------	---------------------

1.112.2.7 openfile()

```
def anise.utils.basic.openfile (
    filepath )
```

Opens a file path in the operating system's default way.

Parameters

<i>filepath</i>	A file path (as string).
-----------------	--------------------------

1.112.2.8 `openurl()`

```
def anise.utils.basic.openurl (
    url )
```

Opens an URL (or file path) in the operating system's default way.

Parameters

<i>url</i>	A url or file path (as string).
------------	---------------------------------

1.112.2.9 `readfromfile()`

```
def anise.utils.basic.readfromfile (
    filename,
    onestring = True,
    notexist_default = _throw,
    binary = False )
```

Reads the content from a file.

Parameters

<i>filename</i>	The file path.
<i>onestring</i>	If the content is to be read in one single string (instead of a list of lines).
<i>notexist_default</i>	The default content when the file does not exist.
<i>binary</i>	If this is binary content instead of text.

1.112.2.10 `writetofile()`

```
def anise.utils.basic.writetofile (
    filename,
    content )
```

Writes some content to a file.

Parameters

<i>filename</i>	The file path.
<i>content</i>	The content to write.

1.112.3 Variable Documentation

1.112.3.1 `_throw`

```
anise.utils.basic._throw [private]
```

1.113 anise.utils.data Namespace Reference

Helpers for creation and handling of some data structures.

Functions

- def `constructobject` (d)
Creates an object with properties from the given keyword arguments.
- def `createobjectstructure` (root, newnames, objclass=None)
Creates a chain of new objects, so you can access property chains like `foo.bar.baz.bam` on the root object.
- def `getuniqueidentifier` ()
Returns a random and (statistically nearly) unique string.
- def `stripname` (name, lowercase=True)
Strips a name to something very bare with only alphabetic letters.

1.113.1 Detailed Description

Helpers for creation and handling of some data structures.

1.113.2 Function Documentation

1.113.2.1 `constructobject()`

```
def anise.utils.data.constructobject (  
    d )
```

Creates an object with properties from the given keyword arguments.

1.113.2.2 `createobjectstructure()`

```
def anise.utils.data.createobjectstructure (  
    root,  
    newnames,  
    objclass = None )
```

Creates a chain of new objects, so you can access property chains like `foo.bar.baz.bam` on the root object.

Parameters

<i>root</i>	The object to populate with the new structure.
<i>newnames</i>	The complete property chain (as string).
<i>objclass</i>	Which object class to use for the cascaded new members.

Returns

: Tuple of the deepest object in the cascade (corresponding to the last part of the chain) and its parent.

1.113.2.3 getuniqueidentifier()

```
def anise.utils.data.getuniqueidentifier ( )
```

Returns a random and (statistically nearly) unique string.

1.113.2.4 stripname()

```
def anise.utils.data.stripname (
    name,
    lowercase = True )
```

Strips a name to something very bare with only alphabetic letters.

Parameters

<i>name</i>	The input name.
<i>lowercase</i>	If only lowercase characters are allowed.

1.114 anise.utils.logging Namespace Reference

Logging.

Classes

- class [Severity](#)

Enumeration of log message severities.

Functions

- def `logmessage` (text, severity, printcaller=True, skiplevels=1, terminalonly=False)
Logs a text message.
- def `_logmessage_default` (text, severity, terminalonly, tofile, textescaper)
This is the fallback lowlevel logging implementation.
- def `setlowlevellogger` (fct)
Sets the lowlevel logging implementation.
- def `logdebug` (text, printcaller=True, skiplevels=1, terminalonly=False)
Logs a text message with debug severity.
- def `loginfo` (text, printcaller=True, skiplevels=1, terminalonly=False)
Logs a text message with info severity.
- def `logwarning` (text, printcaller=True, skiplevels=1, terminalonly=False)
Logs a text message with warning severity.
- def `logerror` (text, printcaller=True, skiplevels=1, terminalonly=False)
Logs a text message with error severity.
- def `logexception` (exception, prolog="Error: ")
Logs an exception.

Variables

- `_logmessage`

1.114.1 Detailed Description

Logging.

1.114.2 Function Documentation

1.114.2.1 `_logmessage_default()`

```
def anise.utils.logging._logmessage_default (  
    text,  
    severity,  
    terminalonly,  
    tofile,  
    textescaper ) [private]
```

This is the fallback lowlevel logging implementation.

Typically it gets overridden by a different one in early initialization steps.

1.114.2.2 `logdebug()`

```
def anise.utils.logging.logdebug (  
    text,  
    printcaller = True,  
    skiplevels = 1,  
    terminalonly = False )
```

Logs a text message with debug severity.

Parameters

<i>text</i>	The log message.
<i>printcaller</i>	Also logs the caller name.
<i>skiplevels</i>	If given, it controls the position on the call stack which is used for logging the caller name.
<i>terminalonly</i>	If this log message shall just go to the terminal and not to the report, the web interface, etc. Only used by the infrastructure.

1.114.2.3 logerror()

```
def anise.utils.logging.logerror (
    text,
    printcaller = True,
    skiplevels = 1,
    terminalonly = False )
```

Logs a text message with error severity.

Parameters

<i>text</i>	The log message.
<i>printcaller</i>	Also logs the caller name.
<i>skiplevels</i>	If given, it controls the position on the call stack which is used for logging the caller name.
<i>terminalonly</i>	If this log message shall just go to the terminal and not to the report, the web interface, etc. Only used by the infrastructure.

1.114.2.4 logexception()

```
def anise.utils.logging.logexception (
    exception,
    prolog = "Error: " )
```

Logs an exception.

Parameters

<i>exception</i>	The occurred exception. This only works from within the <code>except</code> -handler of this exception.
<i>prolog</i>	This is prepended to the exception text in the output.

1.114.2.5 loginfo()

```
def anise.utils.logging.loginfo (
    text,
```

```
printcaller = True,  
skiplevels = 1,  
terminalonly = False )
```

Logs a text message with info severity.

Parameters

<i>text</i>	The log message.
<i>printcaller</i>	Also logs the caller name.
<i>skiplevels</i>	If given, it controls the position on the call stack which is used for logging the caller name.
<i>terminalonly</i>	If this log message shall just go to the terminal and not to the report, the web interface, etc. Only used by the infrastructure.

1.114.2.6 logmessage()

```
def anise.utils.logging.logmessage (  
    text,  
    severity,  
    printcaller = True,  
    skiplevels = 1,  
    terminalonly = False )
```

Logs a text message.

Parameters

<i>text</i>	The log message.
<i>severity</i>	The message severity. One of the values from anise.utils.logging.Severity .
<i>printcaller</i>	Also logs the caller name.
<i>skiplevels</i>	If given, it controls the position on the call stack which is used for logging the caller name.
<i>terminalonly</i>	If this log message shall just go to the terminal and not to the report, the web interface, etc. Only used by the infrastructure.

1.114.2.7 logwarning()

```
def anise.utils.logging.logwarning (  
    text,  
    printcaller = True,  
    skiplevels = 1,  
    terminalonly = False )
```

Logs a text message with warning severity.

Parameters

<i>text</i>	The log message.
-------------	------------------

Parameters

<i>printcaller</i>	Also logs the caller name.
<i>skiplevels</i>	If given, it controls the position on the call stack which is used for logging the caller name.
<i>terminalonly</i>	If this log message shall just go to the terminal and not to the report, the web interface, etc. Only used by the infrastructure.

1.114.2.8 setlowlevellogger()

```
def anise.utils.logging.setlowlevellogger (
    fct )
```

Sets the lowlevel logging implementation.

Parameters

<i>fct</i>	The logger function.
------------	----------------------

1.114.3 Variable Documentation**1.114.3.1 _logmessage**

```
anise.utils.logging._logmessage [private]
```

1.115 anise.utils.ssh Namespace Reference

Utilities for ssh usage.

Classes

- class [Mount](#)
Mounts a volume via ssh.

Functions

- def [call](#) (server, command, options=None, port=22, identityfile=None)
Executes a command via ssh.
- def [copy](#) (src, dst, options=None, port=22, identityfile=None)
Copies a local directory to a remote place via ssh.

1.115.1 Detailed Description

Utilities for ssh usage.

1.115.2 Function Documentation

1.115.2.1 call()

```
def anise.utils.ssh.call (
    server,
    command,
    options = None,
    port = 22,
    identityfile = None )
```

Executes a command via ssh.

Parameters

<i>server</i>	The ssh server hostname.
<i>command</i>	The command to execute.
<i>options</i>	Additional ssh options (as string list).
<i>port</i>	Port number.
<i>identityfile</i>	Optional identity file for authentication.

1.115.2.2 copy()

```
def anise.utils.ssh.copy (
    src,
    dst,
    options = None,
    port = 22,
    identityfile = None )
```

Copies a local directory to a remote place via ssh.

Parameters

<i>src</i>	The source.
<i>dst</i>	The destination.
<i>options</i>	Optional additional mount options.
<i>port</i>	Optional tcp port.
<i>identityfile</i>	Optional identity file for authentication.

1.116 test_custom_processor Namespace Reference

Classes

- class [MyObject](#)
- class [MyProcessor](#)
- class [Test](#)

Variables

- [mydir](#) = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

1.116.1 Variable Documentation

1.116.1.1 mydir

```
test_custom_processor.mydir = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

1.117 test_custom_processor_filterobjects Namespace Reference

Classes

- class [MyProcessor](#)
- class [Test](#)

Variables

- [mydir](#) = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

1.117.1 Variable Documentation

1.117.1.1 mydir

```
test_custom_processor_filterobjects.mydir = os.path.dirname(os.path.dirname(os.path.abspath(↵  
__file__)))
```

1.118 test_custom_processor_filterobjects_completelistbyrequirements Namespace Reference

Classes

- class [MyProcessor](#)
- class [Test](#)

Variables

- [mydir](#) = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

1.118.1 Variable Documentation

1.118.1.1 mydir

```
test_custom_processor_filterobjects_completelistbyrequirements.mydir = os.path.dirname(os.↵  
path.dirname(os.path.abspath(__file__)))
```

1.119 test_deps_afterbeforerequired Namespace Reference

Classes

- class [Test](#)

Variables

- [mydir](#) = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

1.119.1 Variable Documentation

1.119.1.1 mydir

```
test_deps_afterbeforerequired.mydir = os.path.dirname(os.path.dirname(os.path.abspath(__file↵  
__)))
```

1.120 test_deps_afterrequired Namespace Reference

Classes

- class [Test](#)

Variables

- [mydir](#) = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

1.120.1 Variable Documentation

1.120.1.1 mydir

```
test_deps_afterrequired.mydir = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

1.121 test_deps_afterrequiredoptional Namespace Reference

Classes

- class [Test](#)

Variables

- [mydir](#) = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

1.121.1 Variable Documentation

1.121.1.1 mydir

```
test_deps_afterrequiredoptional.mydir = os.path.dirname(os.path.dirname(os.path.abspath(__↵
file__)))
```

1.122 anise.framework.projects._UniverseProxy Class Reference

Internally used for the universe object for allowing just to write `universe` instead of `universe\(\)`.

Public Member Functions

- def [__getattr__](#) (self, item)
- def [__setattr__](#) (self, key, value)
- def [__call__](#) (self, args, kwargs)

1.122.1 Detailed Description

Internally used for the universe object for allowing just to write `universe` instead of `universe()`.

1.122.2 Member Function Documentation

1.122.2.1 [__call__\(\)](#)

```
def anise.framework.projects._UniverseProxy.__call__ (
    self,
    args,
    kwargs )
```

1.122.2.2 [__getattr__\(\)](#)

```
def anise.framework.projects._UniverseProxy.__getattr__ (
    self,
    item )
```

1.122.2.3 [__setattr__\(\)](#)

```
def anise.framework.projects._UniverseProxy.__setattr__ (
    self,
    key,
    value )
```

The documentation for this class was generated from the following file:

- [anise/framework/projects.py](#)

1.123 anise.features.versioncontrol.svn.internals._Version Class Reference

Version number of an Anise Subversion project.

Public Member Functions

- def `__init__` (self, `build`, `tag`)
- def `isvalid` (self)
- def `format` (self)

Public Attributes

- `build`
- `tag`

1.123.1 Detailed Description

Version number of an Anise Subversion project.

1.123.2 Constructor & Destructor Documentation

1.123.2.1 `__init__()`

```
def anise.features.versioncontrol.svn.internals._Version.__init__ (  
    self,  
    build,  
    tag )
```

1.123.3 Member Function Documentation

1.123.3.1 `format()`

```
def anise.features.versioncontrol.svn.internals._Version.format (  
    self )
```

1.123.3.2 `isvalid()`

```
def anise.features.versioncontrol.svn.internals._Version.isvalid (  
    self )
```

1.123.4 Member Data Documentation

1.123.4.1 build

```
anise.features.versioncontrol.svn.internals._Version.build
```

1.123.4.2 tag

```
anise.features.versioncontrol.svn.internals._Version.tag
```

The documentation for this class was generated from the following file:

- [anise/features/versioncontrol/svn.py](#)

1.124 anise.features.versioncontrol.git.internals._Version Class Reference

Version number of an Anise Git project.

Public Member Functions

- `def __init__ (self, build, tag)`
- `def isvalid (self)`
- `def format (self)`

Public Attributes

- [build](#)
- [tag](#)

1.124.1 Detailed Description

Version number of an Anise Git project.

1.124.2 Constructor & Destructor Documentation

1.124.2.1 `__init__()`

```
def anise.features.versioncontrol.git.internals._Version.__init__ (
    self,
    build,
    tag )
```

1.124.3 Member Function Documentation

1.124.3.1 `format()`

```
def anise.features.versioncontrol.git.internals._Version.format (
    self )
```

1.124.3.2 `isvalid()`

```
def anise.features.versioncontrol.git.internals._Version.isvalid (
    self )
```

1.124.4 Member Data Documentation

1.124.4.1 `build`

```
anise.features.versioncontrol.git.internals._Version.build
```

1.124.4.2 `tag`

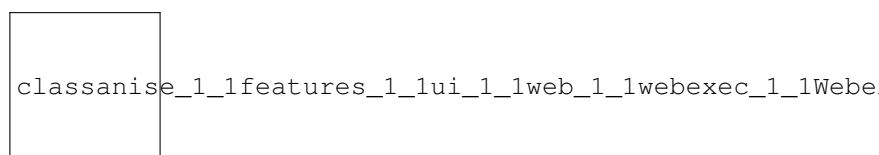
```
anise.features.versioncontrol.git.internals._Version.tag
```

The documentation for this class was generated from the following file:

- [anise/features/versioncontrol/git.py](#)

1.125 `anise.features.ui.web.webexec.WebexecApplication.AbortedInterrupt` Class Reference

Inheritance diagram for `anise.features.ui.web.webexec.WebexecApplication.AbortedInterrupt`:



Public Member Functions

- def [__init__](#) (self)
- def [__str__](#) (self)

1.125.1 Constructor & Destructor Documentation

1.125.1.1 [__init__\(\)](#)

```
def anise.features.ui.web.webexec.WebexecApplication.AbortedInterrupt.__init__ (
    self )
```

1.125.2 Member Function Documentation

1.125.2.1 [__str__\(\)](#)

```
def anise.features.ui.web.webexec.WebexecApplication.AbortedInterrupt.__str__ (
    self )
```

The documentation for this class was generated from the following file:

- anise/features/ui/web/[webexec.py](#)

1.126 anise.features.projectdescriptioneditor.datainjections.AbstractDataInjectionGenerator Class Reference

Inheritance diagram for anise.features.projectdescriptioneditor.datainjections.AbstractDataInjectionGenerator:

```
class anise_1_1features_1_1projectdescriptioneditor_
```

Public Member Functions

- def [__init__](#) (self)
- def [getfctref](#) (self)

Public Attributes

- [label](#)

1.126.1 Constructor & Destructor Documentation

1.126.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.datainjections.AbstractDataInjectionGenerator.__init__(
    self )
```

1.126.2 Member Function Documentation

1.126.2.1 `getfctref()`

```
def anise.features.projectdescriptioneditor.datainjections.AbstractDataInjectionGenerator.getfctref(
    self )
```

1.126.3 Member Data Documentation

1.126.3.1 `label`

```
anise.features.projectdescriptioneditor.datainjections.AbstractDataInjectionGenerator.label
```

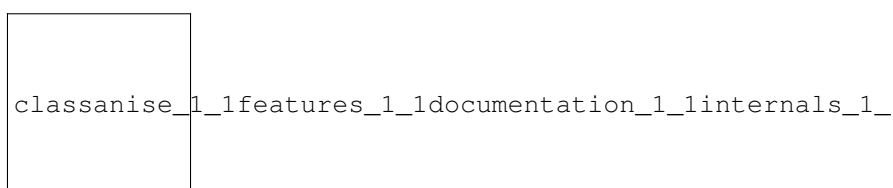
The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/datainjections.py](#)

1.127 `anise.features.documentation.internals.AbstractFileExport` Class Reference

Abstract base class for exports, which write documentation content to files.

Inheritance diagram for `anise.features.documentation.internals.AbstractFileExport`:



Public Member Functions

- def `__init__` (self, `key`, `outputfile`, `format`, `name=None`, `heading=None`, `includedeveloperdoc=False`, `extractall=True`, `extractprivate=True`, `hideundocumented=False`, `imagepaths=None`, `additionaltext=""`, `shownavbar=False`, `excludepatterns=None`, `predefineddocmacros=None`, `head3=None`, `head4=None`, `source=None`, `sources=None`)
- def `generate` (self, `targetpath`)
Generates the content.

Public Attributes

- `outputfile`
- `key`
- `format`
- `name`
- `heading`
- `includedeveloperdoc`
- `extractall`
- `extractprivate`
- `hideundocumented`
- `imagepaths`
- `additionaltext`
- `shownavbar`
- `excludepatterns`
- `predefineddocmacros`
- `head3`
- `head4`
- `source`
- `sources`

1.127.1 Detailed Description

Abstract base class for exports, which write documentation content to files.

1.127.2 Constructor & Destructor Documentation

1.127.2.1 `__init__()`

```
def anise.features.documentation.internals.AbstractFileExport.__init__ (
    self,
    key,
    outputfile,
    format,
    name = None,
    heading = None,
    includedeveloperdoc = False,
    extractall = True,
    extractprivate = True,
```

```
hideundocumented = False,  
imagepaths = None,  
additionaltext = "",  
shownavbar = False,  
excludepatterns = None,  
predefineddocmacros = None,  
head3 = None,  
head4 = None,  
source = None,  
sources = None )
```

1.127.3 Member Function Documentation

1.127.3.1 generate()

```
def anise.features.documentation.internals.Export.generate (  
    self,  
    targetpath ) [inherited]
```

Generates the content.

Override this method in custom subclasses or leave the default implementation.

1.127.4 Member Data Documentation

1.127.4.1 additionaltext

```
anise.features.documentation.internals.Export.additionaltext [inherited]
```

1.127.4.2 excludepatterns

```
anise.features.documentation.internals.Export.excludepatterns [inherited]
```

1.127.4.3 extractall

```
anise.features.documentation.internals.Export.extractall [inherited]
```


1.127.4.4 extractprivate

`anise.features.documentation.internals.Export.extractprivate` [inherited]

1.127.4.5 format

`anise.features.documentation.internals.Export.format` [inherited]

1.127.4.6 head3

`anise.features.documentation.internals.Export.head3` [inherited]

1.127.4.7 head4

`anise.features.documentation.internals.Export.head4` [inherited]

1.127.4.8 heading

`anise.features.documentation.internals.Export.heading` [inherited]

1.127.4.9 hideundocumented

`anise.features.documentation.internals.Export.hideundocumented` [inherited]

1.127.4.10 imagepaths

`anise.features.documentation.internals.Export.imagepaths` [inherited]

1.127.4.11 includedeveloperdoc

`anise.features.documentation.internals.Export.includedeveloperdoc` [inherited]

1.127.4.12 key

`anise.features.documentation.internals.Export.key` [inherited]

1.127.4.13 name

`anise.features.documentation.internals.Export.name` [inherited]

1.127.4.14 outputfile

`anise.features.documentation.internals.AbstractFileExport.outputfile`

1.127.4.15 predefineddocmacros

`anise.features.documentation.internals.Export.predefineddocmacros` [inherited]

1.127.4.16 shownavbar

`anise.features.documentation.internals.Export.shownavbar` [inherited]

1.127.4.17 source

`anise.features.documentation.internals.Export.source` [inherited]

1.127.4.18 sources

`anise.features.documentation.internals.Export.sources` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/documentation.py](#)

1.128 anise.features.projectdescriptioneditor.packages.AbstractPackage Class Reference

Inheritance diagram for anise.features.projectdescriptioneditor.packages.AbstractPackage:

```
class anise_1_1features_1_1projectdescriptioneditor_
```

Public Member Functions

- def [__init__](#) (self)
- def [gettaskref](#) (self)
- def [ask](#) (self)

Public Attributes

- [label](#)
- [taskref](#)

1.128.1 Constructor & Destructor Documentation

1.128.1.1 [__init__\(\)](#)

```
def anise.features.projectdescriptioneditor.packages.AbstractPackage.__init__ (  
    self )
```

1.128.2 Member Function Documentation

1.128.2.1 [ask\(\)](#)

```
def anise.features.projectdescriptioneditor.packages.AbstractPackage.ask (  
    self )
```

1.128.2.2 [gettaskref\(\)](#)

```
def anise.features.projectdescriptioneditor.packages.AbstractPackage.gettaskref (  
    self )
```

1.128.3 Member Data Documentation

1.128.3.1 label

```
anise.features.projectdescriptioneditor.packages.AbstractPackage.label
```

1.128.3.2 taskref

```
anise.features.projectdescriptioneditor.packages.AbstractPackage.taskref
```

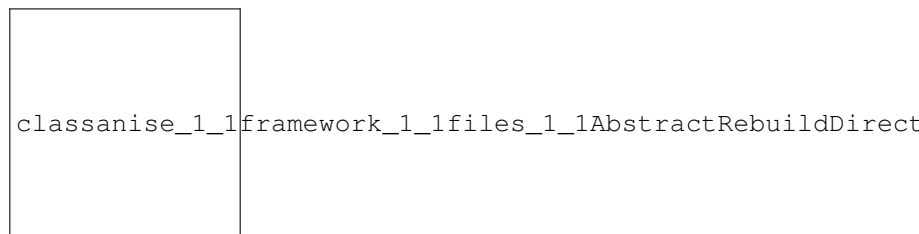
The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/packages.py](#)

1.129 anise.framework.files.AbstractRebuildDirectoriesFilestructure Class Reference

Abstract base class for merging directory structures together.

Inheritance diagram for `anise.framework.files.AbstractRebuildDirectoriesFilestructure`:



Public Member Functions

- `def __init__ (self)`
- `def getdests (self)`
Returns a list of transfer destinations.
- `def getsource (self, desttuple)`
Returns the transfer source (as [anise.framework.files.Filestructure](#)) for a destination tuple (as it was part of `getdests`' result).
- `def getmerged (self)`
- `def datakeys (self)`
Returns a list of keys for all stored metadata properties.
- `def setdata (self, k, v)`
Sets a metadata property.
- `def getdata (self, k, default)`
Gets a metadata property.

- def `initialize` (self)
Initializes this [Filestructure](#).
- def `path` (self)
The path to this structure in the filesystem.
- def `dl` (self, subpath="", to=None, progresscallback=None)
Copies the complete filestructure or a subdirectory to a new destination.
- def `with_modified_rootname` (self, newname)
Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.
- def `mv` (self, newname)
Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.
- def `setautoopen` (self, `path`, interterminal=False)
Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Public Attributes

- `task`
- `params`
- `inner`

1.129.1 Detailed Description

Abstract base class for merging directory structures together.

See the subclasses.

1.129.2 Constructor & Destructor Documentation

1.129.2.1 `__init__()`

```
def anise.framework.files.AbstractRebuildDirectoriesFilestructure.__init__ (
    self )
```

1.129.3 Member Function Documentation

1.129.3.1 `datakeys()`

```
def anise.framework.files.Filestructure.datakeys (
    self ) [inherited]
```

Returns a list of keys for all stored metadata properties.

1.129.3.2 dl()

```
def anise.framework.files.Filestructure.dl (
    self,
    subpath = "",
    to = None,
    progresscallback = None ) [inherited]
```

Copies the complete filestructure or a subdirectory to a new destination.

Parameters

<i>subpath</i>	If given, only this subdirectory will be copied.
<i>to</i>	If given, it specifies the destination path; if not given, a new directory in the current working directory will be used.
<i>progresscallback</i>	Callback function for retrieving progress information.

1.129.3.3 getdata()

```
def anise.framework.files.Filestructure.getdata (
    self,
    k,
    default ) [inherited]
```

Gets a metadata property.

Parameters

<i>k</i>	The key name.
<i>default</i>	The default value (returned if the key doesn't exist).

1.129.3.4 getdests()

```
def anise.framework.files.AbstractRebuildDirectoriesFilestructure.getdests (
    self )
```

Returns a list of transfer destinations.

Returns

: A list of tuples, each beginning with the destination path (and other custom ones).

1.129.3.5 getmerged()

```
def anise.framework.files.AbstractRebuildDirectoriesFilestructure.getmerged (
    self )
```

1.129.3.6 getsource()

```
def anise.framework.files.AbstractRebuildDirectoriesFilestructure.getsource (
    self,
    desttuple )
```

Returns the transfer source (as [anise.framework.files.Filestructure](#)) for a destination tuple (as it was part of getdests' result).

1.129.3.7 initialize()

```
def anise.framework.files.Filestructure.initialize (
    self ) [inherited]
```

Initializes this [Filestructure](#).

When this is called, the dynamic content (if any) is created and written to disc. It is automatically called for typical usage and is not required to be called manually.

1.129.3.8 mv()

```
def anise.framework.files.Filestructure.mv (
    self,
    newname ) [inherited]
```

Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.

Parameters

<i>newname</i>	The new name (absolute or relative).
----------------	--------------------------------------

1.129.3.9 path()

```
def anise.framework.files.Filestructure.path (
    self ) [inherited]
```

The path to this structure in the filesystem.

1.129.3.10 setautoopen()

```
def anise.framework.files.Filestructure.setautoopen (
    self,
    path,
    interterminal = False ) [inherited]
```

Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Parameters

<i>path</i>	The relative path to the file which should automatically be opened.
<i>interterminal</i>	If this hint is for terminal (non-graphical) usage.

1.129.3.11 setdata()

```
def anise.framework.files.Filestructure.setdata (
    self,
    k,
    v ) [inherited]
```

Sets a metadata property.

Parameters

<i>k</i>	The key name.
<i>v</i>	The value.

1.129.3.12 with_modified_rootname()

```
def anise.framework.files.Filestructure.with_modified_rootname (
    self,
    newname ) [inherited]
```

Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.

Parameters

<i>newname</i>	The name which the new root directory should get.
----------------	---

1.129.4 Member Data Documentation

1.129.4.1 inner

`anise.framework.files.TaskExecution.inner` [inherited]

1.129.4.2 params

`anise.framework.files.TaskExecution.params` [inherited]

1.129.4.3 task

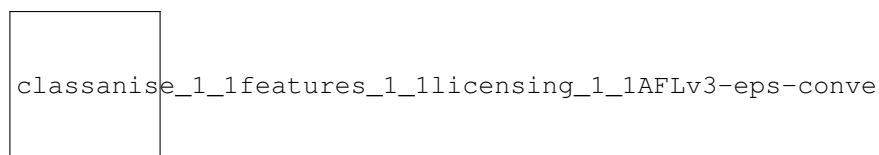
`anise.framework.files.TaskExecution.task` [inherited]

The documentation for this class was generated from the following file:

- `anise/framework/`[files.py](#)

1.130 anise.features.licensing.AFLv3 Class Reference

Inheritance diagram for `anise.features.licensing.AFLv3`:



Public Member Functions

- `def __init__(self)`
- `def getfilename(self)`
Returns the full path of the license text file.
- `def getlicensetext(self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.130.1 Constructor & Destructor Documentation

1.130.1.1 `__init__()`

```
def anise.features.licensing.AFLv3.__init__ (
    self )
```

1.130.2 Member Function Documentation

1.130.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.130.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.130.3 Member Data Documentation

1.130.3.1 `filename`

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.130.3.2 `name`

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.130.3.3 namedebian

`anise.features.licensing.BaseLicense.namedebian` [inherited]

1.130.3.4 namepythonsetuputils

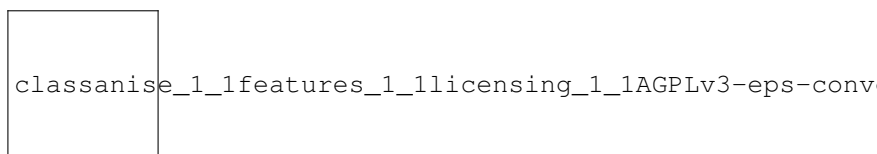
`anise.features.licensing.BaseLicense.namepythonsetuputils` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.131 anise.features.licensing.AGPLv3 Class Reference

Inheritance diagram for `anise.features.licensing.AGPLv3`:



Public Member Functions

- `def __init__ (self)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.131.1 Constructor & Destructor Documentation

1.131.1.1 `__init__()`

```
def anise.features.licensing.AGPLv3.__init__ (
    self )
```

1.131.2 Member Function Documentation

1.131.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.131.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.131.3 Member Data Documentation

1.131.3.1 `filename`

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.131.3.2 `name`

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.131.3.3 `namedebian`

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.131.3.4 namepythonsetuputils

`anise.features.licensing.BaseLicense.namepythonsetuputils` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.132 anise.framework.exceptions.AniseError Class Reference

An error in the execution of Anise.

Inheritance diagram for `anise.framework.exceptions.AniseError`:

```
class anise_1_1framework_1_1exceptions_1_1AniseError
```

Public Member Functions

- `def __call__ (self, args)`

1.132.1 Detailed Description

An error in the execution of Anise.

1.132.2 Member Function Documentation

1.132.2.1 __call__()

```
def anise.framework.exceptions.AniseError.__call__ (
    self,
    args )
```

The documentation for this class was generated from the following file:

- [anise/framework/exceptions.py](#)

1.133 anise.framework.report.AniseOutputStream Class Reference

Proxy for terminal output/error channel, so we can record output messages.

Public Member Functions

- def `__init__` (self, `originalstream`)
- def `write` (self, b)
- def `__getattr__` (self, item)
- def `__dir__` (self)

Public Attributes

- `result`
- `originalstream`

1.133.1 Detailed Description

Proxy for terminal output/error channel, so we can record output messages.

1.133.2 Constructor & Destructor Documentation

1.133.2.1 `__init__()`

```
def anise.framework.report.AniseOutputStream.__init__ (
    self,
    originalstream )
```

1.133.3 Member Function Documentation

1.133.3.1 `__dir__()`

```
def anise.framework.report.AniseOutputStream.__dir__ (
    self )
```

1.133.3.2 `__getattr__()`

```
def anise.framework.report.AniseOutputStream.__getattr__ (
    self,
    item )
```

1.133.3.3 write()

```
def anise.framework.report.AniseOutputStream.write (
    self,
    b )
```

1.133.4 Member Data Documentation

1.133.4.1 originalstream

```
anise.framework.report.AniseOutputStream.originalstream
```

1.133.4.2 result

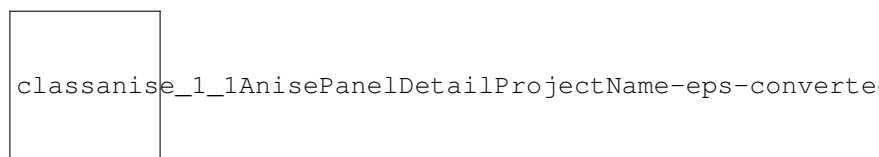
```
anise.framework.report.AniseOutputStream.result
```

The documentation for this class was generated from the following file:

- anise/framework/[report.py](#)

1.134 anise.AnisePanelDetailProjectName Class Reference

Inheritance diagram for anise.AnisePanelDetailProjectName:



Public Member Functions

- def [__init__](#) (self)
- def [set_value](#) (self, detail, node, op)

1.134.1 Constructor & Destructor Documentation

1.134.1.1 `__init__()`

```
def anise.AnisePanelDetailProjectName.__init__ (
    self )
```

1.134.2 Member Function Documentation

1.134.2.1 `set_value()`

```
def anise.AnisePanelDetailProjectName.set_value (
    self,
    detail,
    node,
    op )
```

The documentation for this class was generated from the following file:

- `_meta/shallot_plugin/anise.py`

1.135 `anise.test.anisetest.AniseTestCase` Class Reference

Inheritance diagram for `anise.test.anisetest.AniseTestCase`:



classanise_1_1test_1_1anisetest_1_1AniseTestCase-eps-converted-to.pdf

Public Member Functions

- def `__init__` (self, a, `projectdesc`="raise Exception('test has not specified a `projectdesc`')", kwa)
- def `setUp` (self)
- def `runanise` (self, taskname, values=None, loadfeaturesfrom=None)
- def `tearDown` (self)

Public Attributes

- `projectdesc`
- `rootdir`
- `packagestoredir`
- `mydir`
- `projdescfile`

1.135.1 Constructor & Destructor Documentation

1.135.1.1 `__init__()`

```
def anise.test.anisetest.AniseTestCase.__init__ (
    self,
    a,
    projectdesc = "raise Exception('test has not specified a projectdesc')",
    kwa )
```

1.135.2 Member Function Documentation

1.135.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None )
```

1.135.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self )
```

1.135.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self )
```

1.135.3 Member Data Documentation

1.135.3.1 `mydir`

```
anise.test.anisetest.AniseTestCase.mydir
```

1.135.3.2 `packagestoredir`

```
anise.test.anisetest.AniseTestCase.packagestoredir
```

1.135.3.3 `projdescfile`

```
anise.test.anisetest.AniseTestCase.projdescfile
```

1.135.3.4 `projectdesc`

```
anise.test.anisetest.AniseTestCase.projectdesc
```

1.135.3.5 `rootdir`

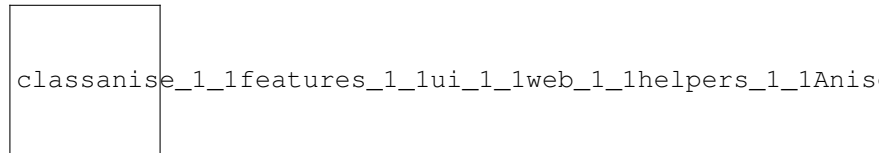
```
anise.test.anisetest.AniseTestCase.rootdir
```

The documentation for this class was generated from the following file:

- [anise/test/anisetest.py](#)

1.136 anise.features.ui.web.helpers.AniseWebApplication Class Reference

Inheritance diagram for anise.features.ui.web.helpers.AniseWebApplication:



Public Member Functions

- def `__init__` (self, title, parentid=None, returntoparent=[True](#))
- def `oninitialize` (self)
- def `start` (self)
- def `stop` (self)
- def `onprocessrequesterror` (self, request, error)
- def `onopenbrowsererror` (self)
- def `onbrowserreopened` (self)
- def `onopenbrowserinformationoutput` (self, kind, message)
- def `clientstaticfilelocations` (self)
- def `do_getprogresses` (self, request, since)
- def `do_getuserfeedback` (self, request)
- def `do_answeruserfeedback` (self, request, index, answer)

Public Attributes

- [startmessageonlyterminal](#)
- [userfeedbackanswers](#)
- [userfeedbackrequest](#)

Private Member Functions

- def `_execution_scopepart_added` (self, x)
- def `_send_userfeedback_web_request` (self, request)
- def `_get_userfeedback_web_answers` (self, requestindex)
- def `_shutdown` (self)

Private Attributes

- [_wh_hooks](#)
- [_pyhtmlview_additionalargs](#)

1.136.1 Constructor & Destructor Documentation

1.136.1.1 `__init__()`

```
def anise.features.ui.web.helpers.AniseWebApplication.__init__ (
    self,
    title,
    parentid = None,
    returntoparent = True )
```

1.136.2 Member Function Documentation

1.136.2.1 `_execution_scopepart_added()`

```
def anise.features.ui.web.helpers.AniseWebApplication._execution_scopepart_added (
    self,
    x ) [private]
```

1.136.2.2 `_get_userfeedback_web_answers()`

```
def anise.features.ui.web.helpers.AniseWebApplication._get_userfeedback_web_answers (
    self,
    requestindex ) [private]
```

1.136.2.3 `_send_userfeedback_web_request()`

```
def anise.features.ui.web.helpers.AniseWebApplication._send_userfeedback_web_request (
    self,
    request ) [private]
```

1.136.2.4 `_shutdown()`

```
def anise.features.ui.web.helpers.AniseWebApplication._shutdown (
    self ) [private]
```

1.136.2.5 `clientstaticfilelocations()`

```
def anise.features.ui.web.helpers.AniseWebApplication.clientstaticfilelocations (
    self )
```

1.136.2.6 do_answeruserfeedback()

```
def anise.features.ui.web.helpers.AniseWebApplication.do_answeruserfeedback (
    self,
    request,
    index,
    answer )
```

1.136.2.7 do_getprogresses()

```
def anise.features.ui.web.helpers.AniseWebApplication.do_getprogresses (
    self,
    request,
    since )
```

1.136.2.8 do_getuserfeedback()

```
def anise.features.ui.web.helpers.AniseWebApplication.do_getuserfeedback (
    self,
    request )
```

1.136.2.9 onbrowserreopened()

```
def anise.features.ui.web.helpers.AniseWebApplication.onbrowserreopened (
    self )
```

1.136.2.10 oninitialize()

```
def anise.features.ui.web.helpers.AniseWebApplication.oninitialize (
    self )
```

1.136.2.11 onopenbrowsererror()

```
def anise.features.ui.web.helpers.AniseWebApplication.onopenbrowsererror (
    self )
```

1.136.2.12 onopenbrowserinformationoutput()

```
def anise.features.ui.web.helpers.AniseWebApplication.onopenbrowserinformationoutput (
    self,
    kind,
    message )
```

1.136.2.13 onprocessrequesterror()

```
def anise.features.ui.web.helpers.AniseWebApplication.onprocessrequesterror (
    self,
    request,
    error )
```

1.136.2.14 start()

```
def anise.features.ui.web.helpers.AniseWebApplication.start (
    self )
```

1.136.2.15 stop()

```
def anise.features.ui.web.helpers.AniseWebApplication.stop (
    self )
```

1.136.3 Member Data Documentation

1.136.3.1 _pyhtmlview_additionalargs

```
anise.features.ui.web.helpers.AniseWebApplication._pyhtmlview_additionalargs [private]
```

1.136.3.2 _wh_hooks

```
anise.features.ui.web.helpers.AniseWebApplication._wh_hooks [private]
```

1.136.3.3 startmessageonlyterminal

```
anise.features.ui.web.helpers.AniseWebApplication.startmessageonlyterminal
```

1.136.3.4 userfeedbackanswers

```
anise.features.ui.web.helpers.AniseWebApplication.userfeedbackanswers
```

1.136.3.5 userfeedbackrequest

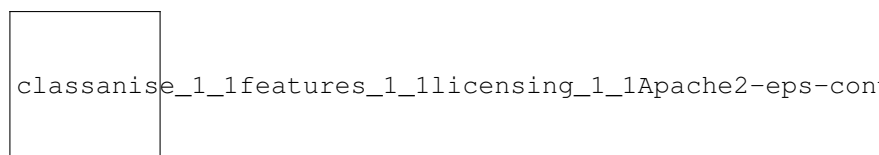
```
anise.features.ui.web.helpers.AniseWebApplication.userfeedbackrequest
```

The documentation for this class was generated from the following file:

- [anise/features/ui/web/helpers.py](#)

1.137 anise.features.licensing.Apache2 Class Reference

Inheritance diagram for anise.features.licensing.Apache2:



Public Member Functions

- `def __init__ (self)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.137.1 Constructor & Destructor Documentation

1.137.1.1 `__init__()`

```
def anise.features.licensing.Apache2.__init__ (
    self )
```

1.137.2 Member Function Documentation

1.137.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.137.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.137.3 Member Data Documentation

1.137.3.1 `filename`

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.137.3.2 `name`

```
anise.features.licensing.BaseLicense.name [inherited]
```


1.137.3.3 namedebian

`anise.features.licensing.BaseLicense.namedebian` [inherited]

1.137.3.4 namepythonsetuputils

`anise.features.licensing.BaseLicense.namepythonsetuputils` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.138 anise.features.diagnostics.internals.ChildrenInformation.Arguments Class Reference

Public Member Functions

- `def __init__(self, args, varargs, varkw, doc)`

Public Attributes

- [args](#)
- [varargs](#)
- [varkw](#)
- [doc](#)

1.138.1 Constructor & Destructor Documentation

1.138.1.1 __init__()

```
def anise.features.diagnostics.internals.ChildrenInformation.Arguments.__init__(
    self,
    args,
    varargs,
    varkw,
    doc )
```

1.138.2 Member Data Documentation

1.138.2.1 args

```
anise.features.diagnostics.internals.ChildrenInformation.Arguments.args
```

1.138.2.2 doc

```
anise.features.diagnostics.internals.ChildrenInformation.Arguments.doc
```

1.138.2.3 varargs

```
anise.features.diagnostics.internals.ChildrenInformation.Arguments.varargs
```

1.138.2.4 varkw

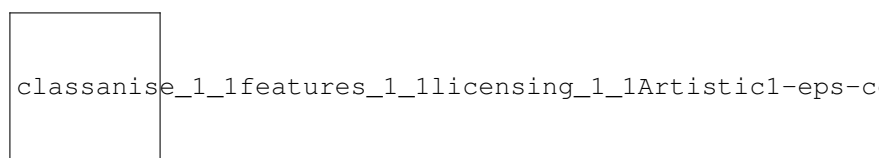
```
anise.features.diagnostics.internals.ChildrenInformation.Arguments.varkw
```

The documentation for this class was generated from the following file:

- [anise/features/diagnostics.py](#)

1.139 anise.features.licensing.Artistic1 Class Reference

Inheritance diagram for `anise.features.licensing.Artistic1`:

**Public Member Functions**

- `def __init__ (self)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.139.1 Constructor & Destructor Documentation

1.139.1.1 `__init__()`

```
def anise.features.licensing.Artistic1.__init__ (  
    self )
```

1.139.2 Member Function Documentation

1.139.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (  
    self ) [inherited]
```

Returns the full path of the license text file.

1.139.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (  
    self ) [inherited]
```

Returns the full license text.

1.139.3 Member Data Documentation

1.139.3.1 `filename`

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.139.3.2 name

`anise.features.licensing.BaseLicense.name` [inherited]

1.139.3.3 namedebian

`anise.features.licensing.BaseLicense.namedebian` [inherited]

1.139.3.4 namepythonsetuputils

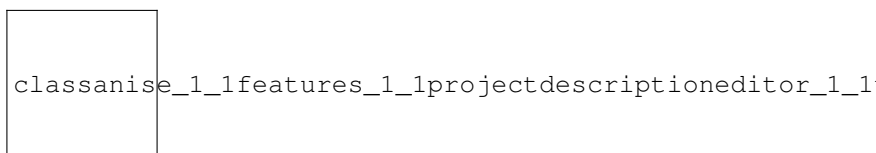
`anise.features.licensing.BaseLicense.namepythonsetuputils` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.140 anise.features.projectdescriptioneditor.testing.AutodiscoveredPyUnitTestingImplementation Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.testing.AutodiscoveredPyUnitTestingImplementation`↵:



Public Member Functions

- `def __init__ (self)`
- `def execute (self, params)`

Public Attributes

- [label](#)

1.140.1 Constructor & Destructor Documentation

1.140.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.testing.AutodiscoveredPyUnitTestingImplementation.  
__init__ (   
    self )
```

1.140.2 Member Function Documentation

1.140.2.1 `execute()`

```
def anise.features.projectdescriptioneditor.testing.AutodiscoveredPyUnitTestingImplementation.  
execute (   
    self,  
    params )
```

1.140.3 Member Data Documentation

1.140.3.1 `label`

`anise.features.projectdescriptioneditor.testing.TestingImplementation.label` [inherited]

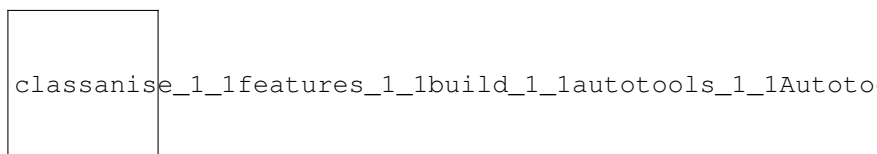
The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/testing.py](#)

1.141 `anise.features.build.autotools.AutotoolsMilieuComposite` Class Reference

Build environment composite for autotools builds.

Inheritance diagram for `anise.features.build.autotools.AutotoolsMilieuComposite`:



Public Member Functions

- def `__init__` (self, [autoreconfpath](#)="autoreconf", makepath="make", processcount=1)
- def [call_autoreconf](#) (self, args, raise_on_errors="unable to run autoreconf", kwargs)

Public Attributes

- [autoreconfpath](#)
- [autoreconfcmd](#)

1.141.1 Detailed Description

Build environment composite for autotools builds.

See [milieus.MilieuComposite](#) for general infos.

1.141.2 Constructor & Destructor Documentation

1.141.2.1 __init__()

```
def anise.features.build.autotools.AutotoolsMilieuComposite.__init__ (
    self,
    autoreconfpath = "autoreconf",
    makepath = "make",
    processcount = 1 )
```

Parameters

<i>autoreconfpath</i>	The path to the <code>autoreconf</code> tool.
<i>makepath</i>	The path to the <code>make</code> tool.
<i>processcount</i>	The maximum number of concurrent processes to execute.

1.141.3 Member Function Documentation

1.141.3.1 call_autoreconf()

```
def anise.features.build.autotools.AutotoolsMilieuComposite.call_autoreconf (
    self,
    args,
    raise_on_errors = "unable to run autoreconf",
    kwargs )
```

1.141.4 Member Data Documentation

1.141.4.1 autoreconfcmd

```
anise.features.build.autotools.AutotoolsMilieuComposite.autoreconfcmd
```

1.141.4.2 autoreconfpath

```
anise.features.build.autotools.AutotoolsMilieuComposite.autoreconfpath
```

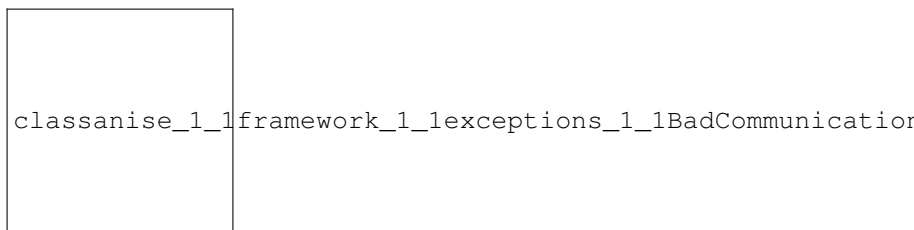
The documentation for this class was generated from the following file:

- anise/features/build/[autotools.py](#)

1.142 anise.framework.exceptions.BadCommunicationError Class Reference

The communication with some other component failed.

Inheritance diagram for anise.framework.exceptions.BadCommunicationError:



Public Member Functions

- def [__call__](#) (self, args)

1.142.1 Detailed Description

The communication with some other component failed.

1.142.2 Member Function Documentation

1.142.2.1 [__call__](#)()

```
def anise.framework.exceptions.AniseError.__call__ (
    self,
    args ) [inherited]
```

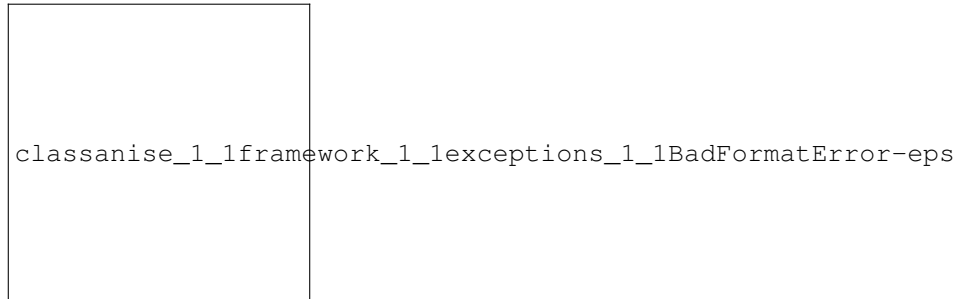
The documentation for this class was generated from the following file:

- anise/framework/[exceptions.py](#)

1.143 `anise.framework.exceptions.BadFormatError` Class Reference

Some input is wrongly formatted.

Inheritance diagram for `anise.framework.exceptions.BadFormatError`:



Public Member Functions

- `def __call__ (self, args)`

1.143.1 Detailed Description

Some input is wrongly formatted.

1.143.2 Member Function Documentation

1.143.2.1 `__call__()`

```
def anise.framework.exceptions.AniseError.__call__ (  
    self,  
    args ) [inherited]
```

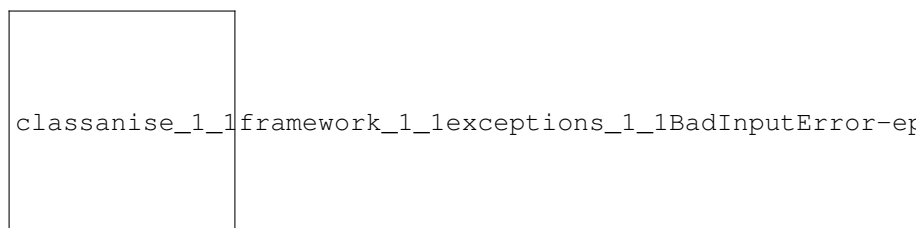
The documentation for this class was generated from the following file:

- `anise/framework/exceptions.py`

1.144 `anise.framework.exceptions.BadInputError` Class Reference

A bad value was given as some input.

Inheritance diagram for `anise.framework.exceptions.BadInputError`:



Public Member Functions

- `def __call__(self, args)`

1.144.1 Detailed Description

A bad value was given as some input.

1.144.2 Member Function Documentation

1.144.2.1 `__call__()`

```
def anise.framework.exceptions.AniseError.__call__ (
    self,
    args ) [inherited]
```

The documentation for this class was generated from the following file:

- `anise/framework/exceptions.py`

1.145 anise.features.licensing.BaseLicense Class Reference

Base class for licenses.

Inheritance diagram for `anise.features.licensing.BaseLicense`:



Public Member Functions

- `def __init__ (self, namedebian=None, namepythonsetuputils=None, name=None, filename=None)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- `name`
- `namedebian`
- `namepythonsetuputils`
- `filename`

1.145.1 Detailed Description

Base class for licenses.

This interface is probably used by many features.

1.145.2 Constructor & Destructor Documentation

1.145.2.1 __init__()

```
def anise.features.licensing.BaseLicense.__init__ (
    self,
    namedebian = None,
    namepythonsetuputils = None,
    name = None,
    filename = None )
```

Parameters

<code>namedebian</code>	The name of the license in Debian packages.
<code>namepythonsetuputils</code>	The name of the license on Python packages.
<code>name</code>	The name of the license.
<code>filename</code>	The path to the file with the license text.

1.145.3 Member Function Documentation

1.145.3.1 getfilename()

```
def anise.features.licensing.BaseLicense.getfilename (
    self )
```

Returns the full path of the license text file.

1.145.3.2 getlicensetext()

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self )
```

Returns the full license text.

1.145.4 Member Data Documentation

1.145.4.1 filename

```
anise.features.licensing.BaseLicense.filename
```

1.145.4.2 name

```
anise.features.licensing.BaseLicense.name
```

1.145.4.3 namedebian

```
anise.features.licensing.BaseLicense.namedebian
```

1.145.4.4 namepythonsetuputils

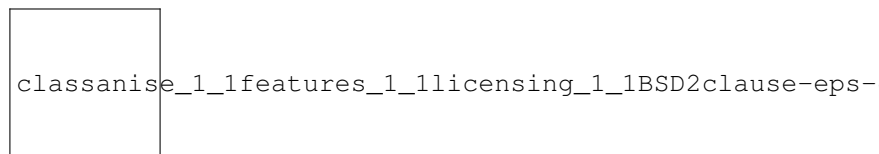
```
anise.features.licensing.BaseLicense.namepythonsetuputils
```

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.146 anise.features.licensing.BSD2clause Class Reference

Inheritance diagram for anise.features.licensing.BSD2clause:



Public Member Functions

- `def __init__ (self)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- `name`
- `namedebian`
- `namepythonsetuputils`
- `filename`

1.146.1 Constructor & Destructor Documentation

1.146.1.1 __init__()

```
def anise.features.licensing.BSD2clause.__init__ (
    self )
```

1.146.2 Member Function Documentation

1.146.2.1 getfilename()

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.146.2.2 getlicensetext()

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.146.3 Member Data Documentation

1.146.3.1 filename

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.146.3.2 name

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.146.3.3 namedebian

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.146.3.4 namepythonsetuputils

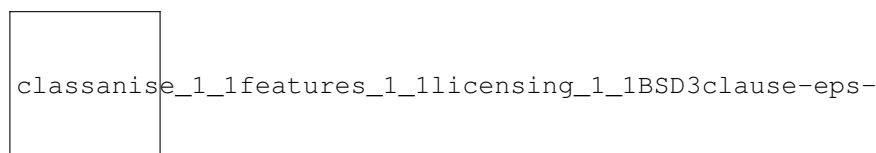
```
anise.features.licensing.BaseLicense.namepythonsetuputils [inherited]
```

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.147 anise.features.licensing.BSD3clause Class Reference

Inheritance diagram for anise.features.licensing.BSD3clause:



Public Member Functions

- def `__init__` (self)
 - def `getfilename` (self)
Returns the full path of the license text file.
- def `getlicensetext` (self)
Returns the full license text.

Public Attributes

- `name`
- `namedebian`
- `namepythonsetuputils`
- `filename`

1.147.1 Constructor & Destructor Documentation

1.147.1.1 `__init__()`

```
def anise.features.licensing.BSD3clause.__init__ (  
    self )
```

1.147.2 Member Function Documentation

1.147.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (  
    self ) [inherited]
```

Returns the full path of the license text file.

1.147.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (  
    self ) [inherited]
```

Returns the full license text.

1.147.3 Member Data Documentation

1.147.3.1 `filename`

`anise.features.licensing.BaseLicense.filename` [inherited]

1.147.3.2 `name`

`anise.features.licensing.BaseLicense.name` [inherited]

1.147.3.3 `namedebian`

`anise.features.licensing.BaseLicense.namedebian` [inherited]

1.147.3.4 `namepythonsetuputils`

`anise.features.licensing.BaseLicense.namepythonsetuputils` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.148 `anise.features.packages.debian.Category` Class Reference

Enumeration for Debian's categories for software packages.

Static Public Attributes

- tuple `ApplicationsAccessibility` = ("Applications/Accessibility", "System")
- tuple `ApplicationsAmateurradio` = ("Applications/Amateur Radio", "Utility")
- tuple `ApplicationsDatamanagement` = ("Applications/Data Management", "System")
- tuple `ApplicationsEditors` = ("Applications/Editors", "Utility")
- tuple `ApplicationsEducation` = ("Applications/Education", "Education")
- tuple `ApplicationsEmulators` = ("Applications/Emulators", "System")
- tuple `ApplicationsFilemanagement` = ("Applications/File Management", "System")
- tuple `ApplicationsGraphics` = ("Applications/Graphics", "Graphics")
- tuple `ApplicationsMobiledevices` = ("Applications/Mobile Devices", "Utility")
- tuple `ApplicationsNetwork` = ("Applications/Network", "Network")
- tuple `ApplicationsNetworkCommunication` = ("Applications/Network/Communication", "Network")
- tuple `ApplicationsNetworkFiletransfer` = ("Applications/Network/File Transfer", "Network")
- tuple `ApplicationsNetworkMonitoring` = ("Applications/Network/Monitoring", "Network")
- tuple `ApplicationsNetworkWebbrowsing` = ("Applications/Network/Web Browsing", "Network")
- tuple `ApplicationsNetworkWebnews` = ("Applications/Network/Web News", "Network")
- tuple `ApplicationsOffice` = ("Applications/Office", "Office")
- tuple `ApplicationsProgramming` = ("Applications/Programming", "Development")
- tuple `ApplicationsProjectmanagement` = ("Applications/Project Management", "Development")
- tuple `ApplicationsScience` = ("Applications/Science", "Education")
- tuple `ApplicationsScienceAstronomy` = ("Applications/Science/Astronomy", "Education")
- tuple `ApplicationsScienceBiology` = ("Applications/Science/Biology", "Education")
- tuple `ApplicationsScienceChemistry` = ("Applications/Science/Chemistry", "Education")
- tuple `ApplicationsScienceDataanalysis` = ("Applications/Science/Data Analysis", "Education")
- tuple `ApplicationsScienceElectronics` = ("Applications/Science/Electronics", "Education")
- tuple `ApplicationsScienceEngineering` = ("Applications/Science/Engineering", "Education")
- tuple `ApplicationsScienceGeoscience` = ("Applications/Science/Geoscience", "Education")
- tuple `ApplicationsScienceMathematics` = ("Applications/Science/Mathematics", "Education")
- tuple `ApplicationsScienceMedicine` = ("Applications/Science/Medicine", "Education")
- tuple `ApplicationsSciencePhysics` = ("Applications/Science/Physics", "Education")
- tuple `ApplicationsScienceSocial` = ("Applications/Science/Social", "Education")
- tuple `ApplicationsShells` = ("Applications/Shells", "System")
- tuple `ApplicationsSounds` = ("Applications/Sound", "AudioVideo")
- tuple `ApplicationsSystem` = ("Applications/System", "System")
- tuple `ApplicationsSystemAdministration` = ("Applications/System/Administration", "System")
- tuple `ApplicationsSystemHardware` = ("Applications/System/Hardware", "System")
- tuple `ApplicationsSystemLanguageenvironment` = ("Applications/System/Language Environment", "System")
- tuple `ApplicationsSystemMonitoring` = ("Applications/System/Monitoring", "System")
- tuple `ApplicationsSystemPackagemanagement` = ("Applications/System/Package Management", "System")
- tuple `ApplicationsSystemSecurity` = ("Applications/System/Security", "System")
- tuple `ApplicationsTerminalemulators` = ("Applications/Terminal Emulators", "System")
- tuple `ApplicationsText` = ("Applications/Text", "Utility")
- tuple `ApplicationsTvandrado` = ("Applications/TV and Radio", "AudioVideo")
- tuple `ApplicationsViewers` = ("Applications/Viewers", "Utility")
- tuple `ApplicationsVideo` = ("Applications/Video", "AudioVideo")
- tuple `ApplicationsWebdevelopment` = ("Applications/Web Development", "Development")
- tuple `GamesAction` = ("Games/Action", "Game")
- tuple `GamesAdventure` = ("Games/Adventure", "Game")
- tuple `GamesBlocks` = ("Games/Blocks", "Game")
- tuple `GamesBoard` = ("Games/Board", "Game")
- tuple `GamesCard` = ("Games/Card", "Game")
- tuple `GamesPuzzles` = ("Games/Puzzles", "Game")
- tuple `GamesSimulation` = ("Games/Simulation", "Game")
- tuple `GamesStrategy` = ("Games/Strategy", "Game")

- tuple [GamesTools](#) = ("Games/Tools", "Game")
- tuple [GamesToys](#) = ("Games/Toys", "Game")
- tuple [Help](#) = ("Help", "System")
- tuple [ScreenSaving](#) = ("Screen/Saving", "System")
- tuple [ScreenLocking](#) = ("Screen/Locking", "System")

1.148.1 Detailed Description

Enumeration for Debian's categories for software packages.

1.148.2 Member Data Documentation

1.148.2.1 ApplicationsAccessibility

```
tuple anise.features.packages.debian.Category.ApplicationsAccessibility = ("Applications/Accessibility",  
"System") [static]
```

1.148.2.2 ApplicationsAmateurradio

```
tuple anise.features.packages.debian.Category.ApplicationsAmateurradio = ("Applications/Amateur  
Radio", "Utility") [static]
```

1.148.2.3 ApplicationsDatamanagement

```
tuple anise.features.packages.debian.Category.ApplicationsDatamanagement = ("Applications/Data  
Management", "System") [static]
```

1.148.2.4 ApplicationsEditors

```
tuple anise.features.packages.debian.Category.ApplicationsEditors = ("Applications/Editors",  
"Utility") [static]
```

1.148.2.5 ApplicationsEducation

```
tuple anise.features.packages.debian.Category.ApplicationsEducation = ("Applications/Education",  
"Education") [static]
```

1.148.2.6 ApplicationsEmulators

```
tuple anise.features.packages.debian.Category.ApplicationsEmulators = ("Applications/Emulators",  
"System") [static]
```

1.148.2.7 ApplicationsFilemanagement

```
tuple anise.features.packages.debian.Category.ApplicationsFilemanagement = ("Applications/File  
Management", "System") [static]
```

1.148.2.8 ApplicationsGraphics

```
tuple anise.features.packages.debian.Category.ApplicationsGraphics = ("Applications/Graphics",  
"Graphics") [static]
```

1.148.2.9 ApplicationsMobiledevices

```
tuple anise.features.packages.debian.Category.ApplicationsMobiledevices = ("Applications/Mobile  
Devices", "Utility") [static]
```

1.148.2.10 ApplicationsNetwork

```
tuple anise.features.packages.debian.Category.ApplicationsNetwork = ("Applications/Network",  
"Network") [static]
```

1.148.2.11 ApplicationsNetworkCommunication

```
tuple anise.features.packages.debian.Category.ApplicationsNetworkCommunication = ("Applications/Network/Commun  
"Network") [static]
```

1.148.2.12 ApplicationsNetworkFiletransfer

```
tuple anise.features.packages.debian.Category.ApplicationsNetworkFiletransfer = ("Applications/Network/File  
Transfer", "Network") [static]
```

1.148.2.13 ApplicationsNetworkMonitoring

```
tuple anise.features.packages.debian.Category.ApplicationsNetworkMonitoring = ("Applications/Network/Monitoring", "Network") [static]
```

1.148.2.14 ApplicationsNetworkWebbrowsing

```
tuple anise.features.packages.debian.Category.ApplicationsNetworkWebbrowsing = ("Applications/Network/Web Browsing", "Network") [static]
```

1.148.2.15 ApplicationsNetworkWebnews

```
tuple anise.features.packages.debian.Category.ApplicationsNetworkWebnews = ("Applications/Network/Web News", "Network") [static]
```

1.148.2.16 ApplicationsOffice

```
tuple anise.features.packages.debian.Category.ApplicationsOffice = ("Applications/Office", "Office") [static]
```

1.148.2.17 ApplicationsProgramming

```
tuple anise.features.packages.debian.Category.ApplicationsProgramming = ("Applications/Programming", "Development") [static]
```

1.148.2.18 ApplicationsProjectmanagement

```
tuple anise.features.packages.debian.Category.ApplicationsProjectmanagement = ("Applications/Project Management", "Development") [static]
```

1.148.2.19 ApplicationsScience

```
tuple anise.features.packages.debian.Category.ApplicationsScience = ("Applications/Science", "Education") [static]
```

1.148.2.20 ApplicationsScienceAstronomy

```
tuple anise.features.packages.debian.Category.ApplicationsScienceAstronomy = ("Applications/Science/Astronomy",  
"Education") [static]
```

1.148.2.21 ApplicationsScienceBiology

```
tuple anise.features.packages.debian.Category.ApplicationsScienceBiology = ("Applications/Science/Biology",  
"Education") [static]
```

1.148.2.22 ApplicationsScienceChemistry

```
tuple anise.features.packages.debian.Category.ApplicationsScienceChemistry = ("Applications/Science/Chemistry",  
"Education") [static]
```

1.148.2.23 ApplicationsScienceDataanalysis

```
tuple anise.features.packages.debian.Category.ApplicationsScienceDataanalysis = ("Applications/Science/Data  
Analysis", "Education") [static]
```

1.148.2.24 ApplicationsScienceElectronics

```
tuple anise.features.packages.debian.Category.ApplicationsScienceElectronics = ("Applications/Science/Electronics",  
"Education") [static]
```

1.148.2.25 ApplicationsScienceEngineering

```
tuple anise.features.packages.debian.Category.ApplicationsScienceEngineering = ("Applications/Science/Engineering",  
"Education") [static]
```

1.148.2.26 ApplicationsScienceGeoscience

```
tuple anise.features.packages.debian.Category.ApplicationsScienceGeoscience = ("Applications/Science/Geoscience",  
"Education") [static]
```

1.148.2.27 ApplicationsScienceMathematics

```
tuple anise.features.packages.debian.Category.ApplicationsScienceMathematics = ("Applications/Science/Mathematics",  
"Education") [static]
```

1.148.2.28 ApplicationsScienceMedicine

```
tuple anise.features.packages.debian.Category.ApplicationsScienceMedicine = ("Applications/Science/Medicine",  
"Education") [static]
```

1.148.2.29 ApplicationsSciencePhysics

```
tuple anise.features.packages.debian.Category.ApplicationsSciencePhysics = ("Applications/Science/Physics",  
"Education") [static]
```

1.148.2.30 ApplicationsScienceSocial

```
tuple anise.features.packages.debian.Category.ApplicationsScienceSocial = ("Applications/Science/Social",  
"Education") [static]
```

1.148.2.31 ApplicationsShells

```
tuple anise.features.packages.debian.Category.ApplicationsShells = ("Applications/Shells",  
"System") [static]
```

1.148.2.32 ApplicationsSounds

```
tuple anise.features.packages.debian.Category.ApplicationsSounds = ("Applications/Sound",  
"AudioVideo") [static]
```

1.148.2.33 ApplicationsSystem

```
tuple anise.features.packages.debian.Category.ApplicationsSystem = ("Applications/System",  
"System") [static]
```

1.148.2.34 ApplicationsSystemAdministration

```
tuple anise.features.packages.debian.Category.ApplicationsSystemAdministration = ("Applications/System/Adminis  
"System") [static]
```

1.148.2.35 ApplicationsSystemHardware

```
tuple anise.features.packages.debian.Category.ApplicationsSystemHardware = ("Applications/System/Hardware",  
"System") [static]
```

1.148.2.36 ApplicationsSystemLanguageenvironment

```
tuple anise.features.packages.debian.Category.ApplicationsSystemLanguageenvironment = ("Applications/System/La  
Environment", "System") [static]
```

1.148.2.37 ApplicationsSystemMonitoring

```
tuple anise.features.packages.debian.Category.ApplicationsSystemMonitoring = ("Applications/System/Monitoring"  
"System") [static]
```

1.148.2.38 ApplicationsSystemPackagemanagement

```
tuple anise.features.packages.debian.Category.ApplicationsSystemPackagemanagement = ("Applications/System/Pack  
Management", "System") [static]
```

1.148.2.39 ApplicationsSystemSecurity

```
tuple anise.features.packages.debian.Category.ApplicationsSystemSecurity = ("Applications/System/Security",  
"System") [static]
```

1.148.2.40 ApplicationsTerminalemulators

```
tuple anise.features.packages.debian.Category.ApplicationsTerminalemulators = ("Applications/Terminal  
Emulators", "System") [static]
```

1.148.2.41 ApplicationsText

```
tuple anise.features.packages.debian.Category.ApplicationsText = ("Applications/Text", "Utility")  
[static]
```

1.148.2.42 ApplicationsTvandrado

```
tuple anise.features.packages.debian.Category.ApplicationsTvandrado = ("Applications/TV and  
Radio", "AudioVideo") [static]
```

1.148.2.43 ApplicationsVideo

```
tuple anise.features.packages.debian.Category.ApplicationsVideo = ("Applications/Video", "Audio↵  
Video") [static]
```

1.148.2.44 ApplicationsViewers

```
tuple anise.features.packages.debian.Category.ApplicationsViewers = ("Applications/Viewers",  
"Utility") [static]
```

1.148.2.45 ApplicationsWebdevelopment

```
tuple anise.features.packages.debian.Category.ApplicationsWebdevelopment = ("Applications/Web  
Development", "Development") [static]
```

1.148.2.46 GamesAction

```
tuple anise.features.packages.debian.Category.GamesAction = ("Games/Action", "Game") [static]
```

1.148.2.47 GamesAdventure

```
tuple anise.features.packages.debian.Category.GamesAdventure = ("Games/Adventure", "Game")  
[static]
```

1.148.2.48 GamesBlocks

```
tuple anise.features.packages.debian.Category.GamesBlocks = ("Games/Blocks", "Game") [static]
```

1.148.2.49 GamesBoard

```
tuple anise.features.packages.debian.Category.GamesBoard = ("Games/Board", "Game") [static]
```

1.148.2.50 GamesCard

```
tuple anise.features.packages.debian.Category.GamesCard = ("Games/Card", "Game") [static]
```

1.148.2.51 GamesPuzzles

```
tuple anise.features.packages.debian.Category.GamesPuzzles = ("Games/Puzzles", "Game") [static]
```

1.148.2.52 GamesSimulation

```
tuple anise.features.packages.debian.Category.GamesSimulation = ("Games/Simulation", "Game")  
[static]
```

1.148.2.53 GamesStrategy

```
tuple anise.features.packages.debian.Category.GamesStrategy = ("Games/Strategy", "Game") [static]
```

1.148.2.54 GamesTools

```
tuple anise.features.packages.debian.Category.GamesTools = ("Games/Tools", "Game") [static]
```


1.148.2.55 GamesToys

```
tuple anise.features.packages.debian.Category.GamesToys = ("Games/Toys", "Game") [static]
```

1.148.2.56 Help

```
tuple anise.features.packages.debian.Category.Help = ("Help", "System") [static]
```

1.148.2.57 ScreenLocking

```
tuple anise.features.packages.debian.Category.ScreenLocking = ("Screen/Locking", "System") [static]
```

1.148.2.58 ScreenSaving

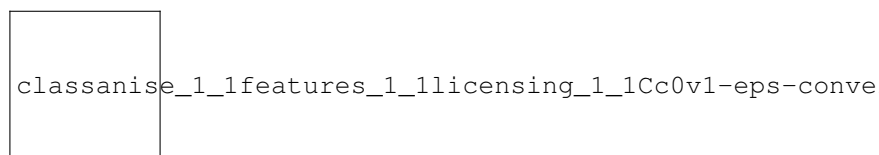
```
tuple anise.features.packages.debian.Category.ScreenSaving = ("Screen/Saving", "System") [static]
```

The documentation for this class was generated from the following file:

- [anise/features/packages/debian.py](#)

1.149 anise.features.licensing.Cc0v1 Class Reference

Inheritance diagram for anise.features.licensing.Cc0v1:



Public Member Functions

- def [__init__](#) (self)
- def [getfilename](#) (self)
Returns the full path of the license text file.
- def [getlicensetext](#) (self)
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.149.1 Constructor & Destructor Documentation

1.149.1.1 `__init__()`

```
def anise.features.licensing.Cc0v1.__init__ (
    self )
```

1.149.2 Member Function Documentation

1.149.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.149.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.149.3 Member Data Documentation

1.149.3.1 `filename`

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.149.3.2 name

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.149.3.3 namedebian

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.149.3.4 namepythonsetuputils

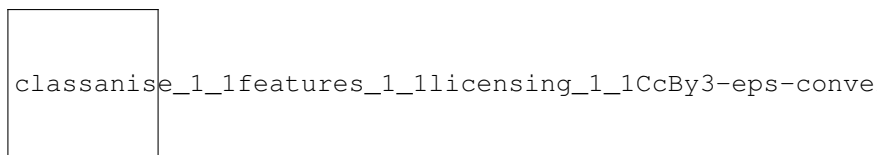
```
anise.features.licensing.BaseLicense.namepythonsetuputils [inherited]
```

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.150 anise.features.licensing.CcBy3 Class Reference

Inheritance diagram for anise.features.licensing.CcBy3:



Public Member Functions

- def [__init__](#) (self)
- def [getfilename](#) (self)
Returns the full path of the license text file.
- def [getlicensetext](#) (self)
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.150.1 Constructor & Destructor Documentation

1.150.1.1 `__init__()`

```
def anise.features.licensing.CcBy3.__init__ (
    self )
```

1.150.2 Member Function Documentation

1.150.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.150.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.150.3 Member Data Documentation

1.150.3.1 `filename`

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.150.3.2 `name`

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.150.3.3 namedebian

`anise.features.licensing.BaseLicense.namedebian` [inherited]

1.150.3.4 namepythonsetuputils

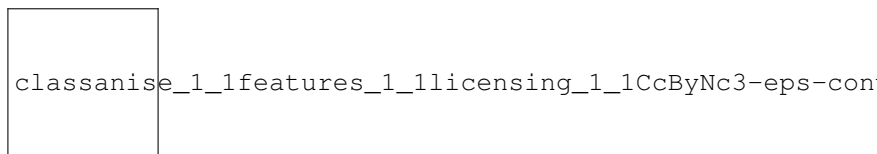
`anise.features.licensing.BaseLicense.namepythonsetuputils` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.151 anise.features.licensing.CcByNc3 Class Reference

Inheritance diagram for `anise.features.licensing.CcByNc3`:



Public Member Functions

- `def __init__ (self)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.151.1 Constructor & Destructor Documentation

1.151.1.1 `__init__()`

```
def anise.features.licensing.CcByNc3.__init__ (
    self )
```

1.151.2 Member Function Documentation

1.151.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.151.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.151.3 Member Data Documentation

1.151.3.1 `filename`

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.151.3.2 `name`

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.151.3.3 `namedebian`

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.151.3.4 namepythonsetuputils

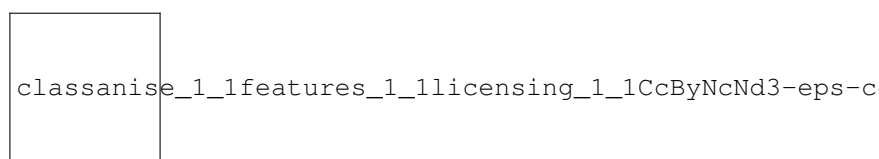
`anise.features.licensing.BaseLicense.namepythonsetuputils` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.152 anise.features.licensing.CcByNcNd3 Class Reference

Inheritance diagram for `anise.features.licensing.CcByNcNd3`:



Public Member Functions

- `def __init__ (self)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.152.1 Constructor & Destructor Documentation

1.152.1.1 `__init__()`

```
def anise.features.licensing.CcByNcNd3.__init__ (
    self )
```

1.152.2 Member Function Documentation

1.152.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.152.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.152.3 Member Data Documentation

1.152.3.1 `filename`

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.152.3.2 `name`

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.152.3.3 `namedebian`

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.152.3.4 `namepythonsetuputils`

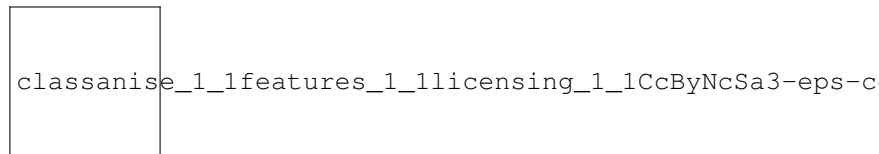
```
anise.features.licensing.BaseLicense.namepythonsetuputils [inherited]
```

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.153 anise.features.licensing.CcByNcSa3 Class Reference

Inheritance diagram for anise.features.licensing.CcByNcSa3:



Public Member Functions

- def `__init__` (self)
- def `getfilename` (self)
Returns the full path of the license text file.
- def `getlicensetext` (self)
Returns the full license text.

Public Attributes

- `name`
- `namedebian`
- `namepythonsetuputils`
- `filename`

1.153.1 Constructor & Destructor Documentation

1.153.1.1 `__init__()`

```
def anise.features.licensing.CcByNcSa3.__init__ (  
    self )
```

1.153.2 Member Function Documentation

1.153.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (  
    self ) [inherited]
```

Returns the full path of the license text file.

1.153.2.2 getlicensetext()

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.153.3 Member Data Documentation

1.153.3.1 filename

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.153.3.2 name

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.153.3.3 namedebian

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.153.3.4 namepythonsetuputils

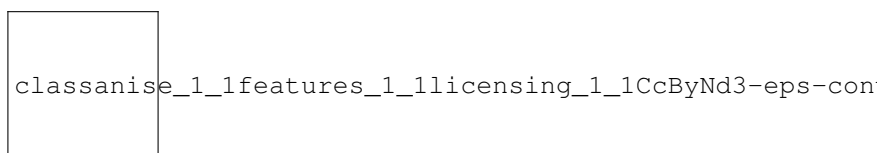
```
anise.features.licensing.BaseLicense.namepythonsetuputils [inherited]
```

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.154 anise.features.licensing.CcByNd3 Class Reference

Inheritance diagram for anise.features.licensing.CcByNd3:



Public Member Functions

- def `__init__` (self)
- def `getfilename` (self)
Returns the full path of the license text file.
- def `getlicensetext` (self)
Returns the full license text.

Public Attributes

- `name`
- `namedebian`
- `namepythonsetuputils`
- `filename`

1.154.1 Constructor & Destructor Documentation

1.154.1.1 `__init__()`

```
def anise.features.licensing.CcByNd3.__init__ (  
    self )
```

1.154.2 Member Function Documentation

1.154.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (  
    self ) [inherited]
```

Returns the full path of the license text file.

1.154.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (  
    self ) [inherited]
```

Returns the full license text.

1.154.3 Member Data Documentation

1.154.3.1 filename

`anise.features.licensing.BaseLicense.filename` [inherited]

1.154.3.2 name

`anise.features.licensing.BaseLicense.name` [inherited]

1.154.3.3 namedebian

`anise.features.licensing.BaseLicense.namedebian` [inherited]

1.154.3.4 namepythonsetuputils

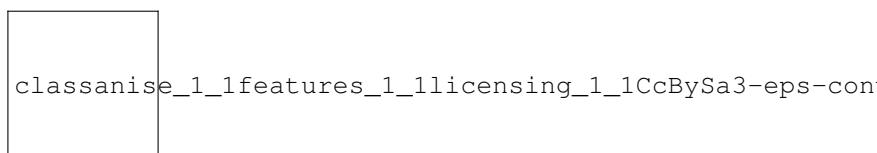
`anise.features.licensing.BaseLicense.namepythonsetuputils` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.155 `anise.features.licensing.CcBySa3` Class Reference

Inheritance diagram for `anise.features.licensing.CcBySa3`:



Public Member Functions

- `def __init__ (self)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.155.1 Constructor & Destructor Documentation

1.155.1.1 `__init__()`

```
def anise.features.licensing.CcBySa3.__init__ (
    self )
```

1.155.2 Member Function Documentation

1.155.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.155.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.155.3 Member Data Documentation

1.155.3.1 `filename`

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.155.3.2 name

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.155.3.3 namedebian

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.155.3.4 namepythonsetuputils

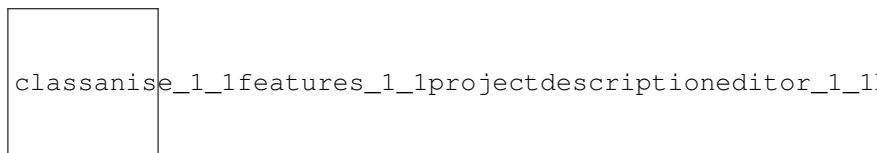
```
anise.features.licensing.BaseLicense.namepythonsetuputils [inherited]
```

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.156 anise.features.projectdescriptioneditor.base.ChangeHookCustomAction Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.base.ChangeHookCustomAction`:

**Public Member Functions**

- `def __init__ (self)`
- `def visible (self, ais, node)`
- `def execute (self, ais, node)`

Public Attributes

- [label](#)
- [function](#)
- [icon](#)

1.156.1 Constructor & Destructor Documentation

1.156.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.base.ChangeHookCustomAction.__init__ (
    self )
```

1.156.2 Member Function Documentation

1.156.2.1 `execute()`

```
def anise.features.projectdescriptioneditor.base.ChangeHookCustomAction.execute (
    self,
    ais,
    node )
```

1.156.2.2 `visible()`

```
def anise.features.projectdescriptioneditor.base.ChangeHookCustomAction.visible (
    self,
    ais,
    node )
```

1.156.3 Member Data Documentation

1.156.3.1 `function`

`anise.features.projectdescriptioneditor.CustomAction.function` [inherited]

1.156.3.2 `icon`

`anise.features.projectdescriptioneditor.CustomAction.icon` [inherited]

1.156.3.3 `label`

`anise.features.projectdescriptioneditor.CustomAction.label` [inherited]

The documentation for this class was generated from the following file:

- `anise/features/projectdescriptioneditor/base.py`

1.157 anise.features.versioncontrol.ChangeLogChange Class Reference

A change entry in a change log.

Public Member Functions

- `def __init__(self, changetext, commitmessage)`

Public Attributes

- [changetext](#)
- [commitmessage](#)

1.157.1 Detailed Description

A change entry in a change log.

1.157.2 Constructor & Destructor Documentation

1.157.2.1 __init__()

```
def anise.features.versioncontrol.ChangeLogChange.__init__ (
    self,
    changetext,
    commitmessage )
```

Parameters

<i>changetext</i>	The change text.
<i>commitmessage</i>	The versioncontrol.CommitMessage which this version label is defined by.

1.157.3 Member Data Documentation

1.157.3.1 changetext

`anise.features.versioncontrol.ChangeLogChange.changetext`

1.157.3.2 commitmessage

`anise.features.versioncontrol.ChangeLogChange.commitmessage`

The documentation for this class was generated from the following file:

- [anise/features/versioncontrol.py](#)

1.158 anise.features.versioncontrol.ChangeLogRevision Class Reference

A revision section in a change log.

Public Member Functions

- `def __init__(self, versionlabel, commitmessage)`

Public Attributes

- [versionlabel](#)
- [commitmessage](#)
- [changes](#)

1.158.1 Detailed Description

A revision section in a change log.

1.158.2 Constructor & Destructor Documentation

1.158.2.1 __init__()

```
def anise.features.versioncontrol.ChangeLogRevision.__init__(  
    self,  
    versionlabel,  
    commitmessage )
```

Parameters

<i>versionlabel</i>	The version label.
<i>commitmessage</i>	The versioncontrol.CommitMessage which this version label is defined by.

1.158.3 Member Data Documentation

1.158.3.1 changes

```
anise.features.versioncontrol.ChangeLogRevision.changes
```

1.158.3.2 commitmessage

```
anise.features.versioncontrol.ChangeLogRevision.commitmessage
```

1.158.3.3 versionlabel

```
anise.features.versioncontrol.ChangeLogRevision.versionlabel
```

The documentation for this class was generated from the following file:

- anise/features/[versioncontrol.py](#)

1.159 anise.utils.basic.ChDir Class Reference

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

Public Member Functions

- `def __init__(self, path)`
- `def __enter__(self)`
- `def __exit__(self, etype, evalue, etraceback)`

Public Attributes

- `newpath`
- `oldpath`

1.159.1 Detailed Description

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

1.159.2 Constructor & Destructor Documentation

1.159.2.1 __init__()

```
def anise.utils.basic.ChDir.__init__(  
    self,  
    path )
```

Parameters

<i>path</i>	The path to the new working directory.
-------------	--

1.159.3 Member Function Documentation

1.159.3.1 `__enter__()`

```
def anise.utils.basic.ChDir.__enter__ (
    self )
```

1.159.3.2 `__exit__()`

```
def anise.utils.basic.ChDir.__exit__ (
    self,
    etype,
    evalue,
    etraceback )
```

1.159.4 Member Data Documentation

1.159.4.1 `newpath`

```
anise.utils.basic.ChDir.newpath
```

1.159.4.2 `oldpath`

```
anise.utils.basic.ChDir.oldpath
```

The documentation for this class was generated from the following file:

- [anise/utils/basic.py](#)

1.160 anise.features.diagnostics.internals.ChildrenInformation Class Reference

Classes

- class [Arguments](#)

Public Member Functions

- def [__init__](#) (self, p, web=False)

Public Attributes

- [comment](#)
- [typeinfo](#)
- [value](#)
- [arguments](#)
- [argumentstruct](#)
- [returns](#)
- [rawdocstring](#)
- [docstring](#)
- [doesexist](#)
- [isclass](#)
- [isbound](#)
- [iscallable](#)
- [issimplevalue](#)
- [lastname part](#)
- [isfeature](#)
- [actions](#)
- [asdict](#)

1.160.1 Constructor & Destructor Documentation

1.160.1.1 [__init__\(\)](#)

```
def anise.features.diagnostics.internals.ChildrenInformation.__init__ (
    self,
    p,
    web = False )
```

1.160.2 Member Data Documentation

1.160.2.1 actions

`anise.features.diagnostics.internals.ChildrenInformation.actions`

1.160.2.2 arguments

`anise.features.diagnostics.internals.ChildrenInformation.arguments`

1.160.2.3 argumentstruct

`anise.features.diagnostics.internals.ChildrenInformation.argumentstruct`

1.160.2.4 asdict

`anise.features.diagnostics.internals.ChildrenInformation.asdict`

1.160.2.5 comment

`anise.features.diagnostics.internals.ChildrenInformation.comment`

1.160.2.6 docstring

`anise.features.diagnostics.internals.ChildrenInformation.docstring`

1.160.2.7 doesexist

`anise.features.diagnostics.internals.ChildrenInformation.doesexist`

1.160.2.8 isbound

`anise.features.diagnostics.internals.ChildrenInformation.isbound`

1.160.2.9 iscallable

```
anise.features.diagnostics.internals.ChildrenInformation.iscallable
```

1.160.2.10 isclass

```
anise.features.diagnostics.internals.ChildrenInformation.isclass
```

1.160.2.11 isfeature

```
anise.features.diagnostics.internals.ChildrenInformation.isfeature
```

1.160.2.12 issimplevalue

```
anise.features.diagnostics.internals.ChildrenInformation.issimplevalue
```

1.160.2.13 lastnamepart

```
anise.features.diagnostics.internals.ChildrenInformation.lastnamepart
```

1.160.2.14 rawdocstring

```
anise.features.diagnostics.internals.ChildrenInformation.rawdocstring
```

1.160.2.15 returns

```
anise.features.diagnostics.internals.ChildrenInformation.returns
```

1.160.2.16 typeinfo

```
anise.features.diagnostics.internals.ChildrenInformation.typeinfo
```

1.160.2.17 value

```
anise.features.diagnostics.internals.ChildrenInformation.value
```

The documentation for this class was generated from the following file:

- [anise/features/diagnostics.py](#)

1.161 anise.features.ui.ChoiceTree Class Reference

A tree of possible choices.

Classes

- class [Node](#)
One node in a tree of possible choices.

Public Member Functions

- def [__init__](#) (self)

Public Attributes

- [root](#)

1.161.1 Detailed Description

A tree of possible choices.

Typically used in `UserFeedback.treechoice`.

For adding nodes to the tree, use the [ChoiceTree.Node.createchild](#) method, which is available on the root node of this tree as well as on all deeper nodes.

1.161.2 Constructor & Destructor Documentation

1.161.2.1 [__init__](#)()

```
def anise.features.ui.ChoiceTree.__init__ (
    self )
```

1.161.3 Member Data Documentation

1.161.3.1 root

```
anise.features.ui.ChoiceTree.root
```

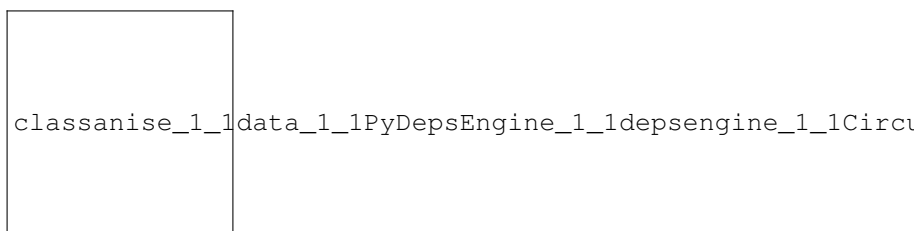
The documentation for this class was generated from the following file:

- [anise/features/ui.py](#)

1.162 anise.data.PyDepsEngine.depsengine.CircularDependenciesError Class Reference

Exception for circular dependencies in dependency resolution.

Inheritance diagram for `anise.data.PyDepsEngine.depsengine.CircularDependenciesError`:



Public Member Functions

- `def __init__(self, remainings)`
- `def __call__(self, args)`

Public Attributes

- [remainings](#)

1.162.1 Detailed Description

Exception for circular dependencies in dependency resolution.

1.162.2 Constructor & Destructor Documentation

1.162.2.1 `__init__()`

```
def anise.data.PyDepsEngine.depsengine.CircularDependenciesError.__init__ (
    self,
    remainings )
```

1.162.3 Member Function Documentation

1.162.3.1 `__call__()`

```
def anise.data.PyDepsEngine.depsengine.EngineError.__call__ (
    self,
    args ) [inherited]
```

1.162.4 Member Data Documentation

1.162.4.1 `remainings`

```
anise.data.PyDepsEngine.depsengine.CircularDependenciesError.remainings
```

The documentation for this class was generated from the following file:

- [anise/data/PyDepsEngine/depsengine.py](#)

1.163 anise.features.versioncontrol.CommitMessage Class Reference

A commit message.

Public Member Functions

- `def __init__` (self, [message](#), [revision](#), [committime](#)=None, [username](#)=None, kwargs)

Public Attributes

- [message](#)
- [revision](#)
- [committime](#)
- [username](#)
- [customdata](#)

1.163.1 Detailed Description

A commit message.

1.163.2 Constructor & Destructor Documentation

1.163.2.1 `__init__()`

```
def anise.features.versioncontrol.CommitMessage.__init__ (
    self,
    message,
    revision,
    committime = None,
    username = None,
    kwargs )
```

Parameters

<i>message</i>	The message text.
<i>revision</i>	The revision.
<i>committime</i>	The time of committing.
<i>username</i>	The user who committed.
<i>kwargs</i>	Additional commit data.

1.163.3 Member Data Documentation

1.163.3.1 `committime`

```
anise.features.versioncontrol.CommitMessage.committime
```

1.163.3.2 `customdata`

```
anise.features.versioncontrol.CommitMessage.customdata
```

1.163.3.3 `message`

```
anise.features.versioncontrol.CommitMessage.message
```

1.163.3.4 revision

```
anise.features.versioncontrol.CommitMessage.revision
```

1.163.3.5 username

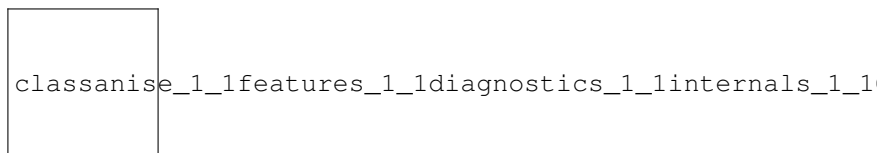
```
anise.features.versioncontrol.CommitMessage.username
```

The documentation for this class was generated from the following file:

- [anise/features/versioncontrol.py](#)

1.164 anise.features.diagnostics.internals.ConsoleWebModule Class Reference

Inheritance diagram for anise.features.diagnostics.internals.ConsoleWebModule:



Public Member Functions

- `def __init__ (self)`
- `def oninitialize (self)`
- `def do_listmembers (self, request, p)`
- `def do_listfeatures (self, request, p)`
- `def do_listkeys (self, request, p)`
- `def do_info (self, request, p)`
- `def do_exec (self, request, p)`
- `def do_execaction (self, request, p, idx)`

Public Attributes

- [webconsolelocals](#)

1.164.1 Constructor & Destructor Documentation

1.164.1.1 __init__()

```
def anise.features.diagnostics.internals.ConsoleWebModule.__init__ (
    self )
```

1.164.2 Member Function Documentation

1.164.2.1 do_exec()

```
def anise.features.diagnostics.internals.ConsoleWebModule.do_exec (
    self,
    request,
    p )
```

1.164.2.2 do_execaction()

```
def anise.features.diagnostics.internals.ConsoleWebModule.do_execaction (
    self,
    request,
    p,
    idx )
```

1.164.2.3 do_info()

```
def anise.features.diagnostics.internals.ConsoleWebModule.do_info (
    self,
    request,
    p )
```

1.164.2.4 do_listfeatures()

```
def anise.features.diagnostics.internals.ConsoleWebModule.do_listfeatures (
    self,
    request,
    p )
```

1.164.2.5 do_listkeys()

```
def anise.features.diagnostics.internals.ConsoleWebModule.do_listkeys (
    self,
    request,
    p )
```

1.164.2.6 do_listmembers()

```
def anise.features.diagnostics.internals.ConsoleWebModule.do_listmembers (
    self,
    request,
    p )
```

1.164.2.7 oninitialize()

```
def anise.features.diagnostics.internals.ConsoleWebModule.oninitialize (
    self )
```

1.164.3 Member Data Documentation

1.164.3.1 webconsolelocals

```
anise.features.diagnostics.internals.ConsoleWebModule.webconsolelocals
```

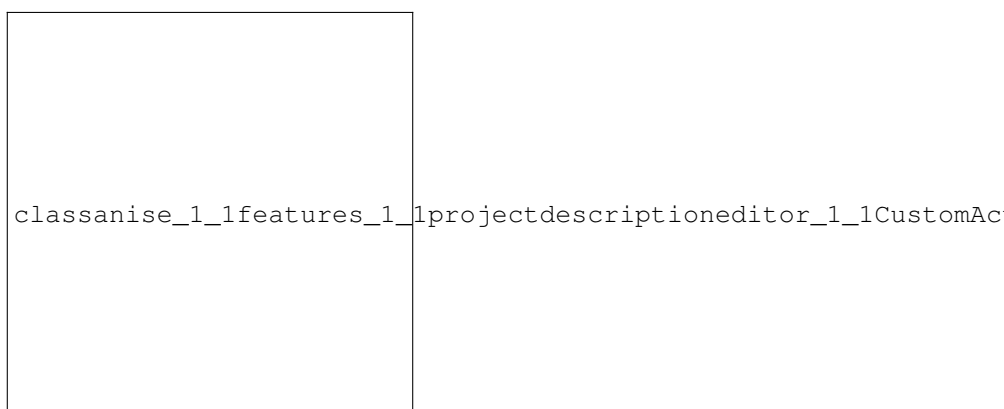
The documentation for this class was generated from the following file:

- [anise/features/diagnostics.py](#)

1.165 anise.features.projectdescriptioneditor.CustomAction Class Reference

A custom action in a [ProjectDescriptionEditorAssistant](#).

Inheritance diagram for anise.features.projectdescriptioneditor.CustomAction:



Public Member Functions

- `def __init__ (self, label, function=None, icon=None)`
- `def visible (self, ais, node)`
Checks if this custom action shall be visible.
- `def execute (self, ais, node)`
Executes this custom action.

Public Attributes

- [label](#)
- [function](#)
- [icon](#)

1.165.1 Detailed Description

A custom action in a [ProjectDescriptionEditorAssistant](#).

Can either be a global one or working on a particular node.

1.165.2 Constructor & Destructor Documentation

1.165.2.1 `__init__()`

```
def anise.features.projectdescriptioneditor.CustomAction.__init__ (
    self,
    label,
    function = None,
    icon = None )
```

Parameters

<i>label</i>	The label text.
<i>function</i>	The action function (if None: 'self.execute' is used).
<i>icon</i>	Icon name.

1.165.3 Member Function Documentation

1.165.3.1 `execute()`

```
def anise.features.projectdescriptioneditor.CustomAction.execute (
    self,
```

```
        ais,  
        node )
```

Executes this custom action.

Override this method in custom subclasses.

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.165.3.2 visible()

```
def anise.features.projectdescriptioneditor.CustomAction.visible (  
    self,  
    ais,  
    node )
```

Checks if this custom action shall be visible.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.165.4 Member Data Documentation

1.165.4.1 function

```
anise.features.projectdescriptioneditor.CustomAction.function
```

1.165.4.2 icon

```
anise.features.projectdescriptioneditor.CustomAction.icon
```

1.165.4.3 label

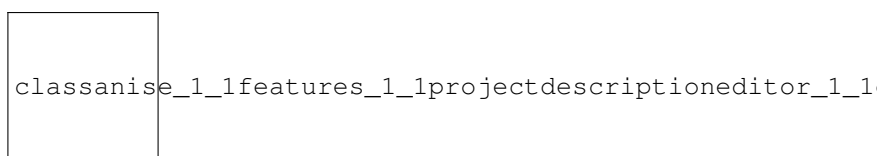
`anise.features.projectdescriptioneditor.CustomAction.label`

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor.py](#)

1.166 `anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorC` Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorC`:



Public Member Functions

- `def __init__ (self)`
- `def getfctref (self)`

Public Attributes

- [label](#)

1.166.1 Constructor & Destructor Documentation

1.166.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorC.__init__ (
    self )
```

1.166.2 Member Function Documentation

1.166.2.1 getfctref()

```
def anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorC.getfctref (
    self )
```

1.166.3 Member Data Documentation

1.166.3.1 label

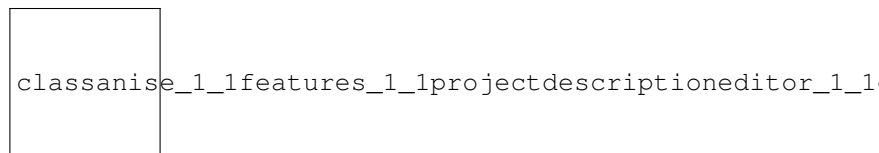
```
anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorC.label
```

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/datainjections.py](#)

1.167 anise.features.projectdescriptioneditor.datainjections.DataInjectionGenerator↵ Custom Class Reference

Inheritance diagram for anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorCustom:



Public Member Functions

- [def __init__](#) (self)
- [def getfctref](#) (self)

Public Attributes

- [label](#)

1.167.1 Constructor & Destructor Documentation

1.167.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorCustom.↔
init__ (
    self )
```

1.167.2 Member Function Documentation

1.167.2.1 `getfctref()`

```
def anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorCustom.↔
getfctref (
    self )
```

1.167.3 Member Data Documentation

1.167.3.1 `label`

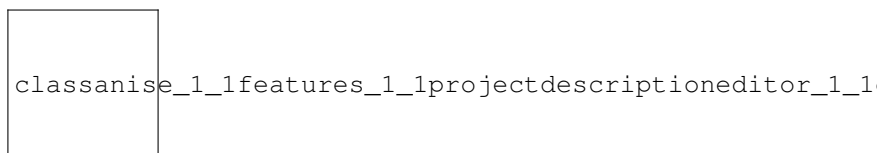
```
anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorCustom.label
```

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/datainjections.py](#)

1.168 `anise.features.projectdescriptioneditor.datainjections.DataInjectionGenerator`↔ Python Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorPython`:



Public Member Functions

- `def __init__ (self)`
- `def getfctref (self)`

Public Attributes

- [label](#)

1.168.1 Constructor & Destructor Documentation

1.168.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorPython.__  
init__ (   
    self )
```

1.168.2 Member Function Documentation

1.168.2.1 `getfctref()`

```
def anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorPython.  
getfctref (   
    self )
```

1.168.3 Member Data Documentation

1.168.3.1 `label`

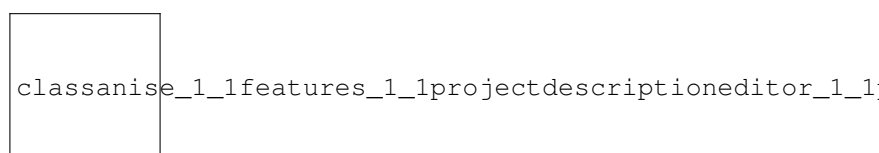
```
anise.features.projectdescriptioneditor.datainjections.DataInjectionGeneratorPython.label
```

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/datainjections.py](#)

1.169 anise.features.projectdescriptioneditor.packages.DebPackage Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.packages.DebPackage`:



Public Member Functions

- def `__init__` (self)
- def `ask` (self)
- def `gettaskref` (self)

Public Attributes

- `label`
- `taskref`

1.169.1 Constructor & Destructor Documentation

1.169.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.packages.DebPackage.__init__ (  
    self )
```

1.169.2 Member Function Documentation

1.169.2.1 `ask()`

```
def anise.features.projectdescriptioneditor.packages.DebPackage.ask (  
    self )
```

1.169.2.2 `gettaskref()`

```
def anise.features.projectdescriptioneditor.packages.AbstractPackage.gettaskref (  
    self ) [inherited]
```

1.169.3 Member Data Documentation

1.169.3.1 `label`

```
anise.features.projectdescriptioneditor.packages.DebPackage.label
```

1.169.3.2 taskref

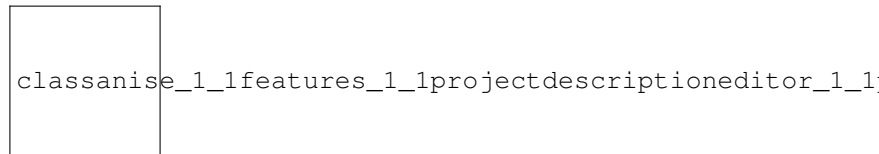
```
anise.features.projectdescriptioneditor.packages.DebPackage.taskref
```

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/packages.py](#)

1.170 anise.features.projectdescriptioneditor.packages.DebPackageAddExecLinkCustomAction Class Reference

Inheritance diagram for anise.features.projectdescriptioneditor.packages.DebPackageAddExecLinkCustomAction:



Public Member Functions

- `def __init__ (self)`
- `def visible (self, ais, node)`
- `def execute (self, ais, node)`

Public Attributes

- [label](#)
- [function](#)
- [icon](#)

1.170.1 Constructor & Destructor Documentation

1.170.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.packages.DebPackageAddExecLinkCustomAction.__  
init__ (  
    self )
```

1.170.2 Member Function Documentation

1.170.2.1 `execute()`

```
def anise.features.projectdescriptioneditor.packages.DebPackageAddExecLinkCustomAction.execute  
(  
    self,  
    ais,  
    node )
```

1.170.2.2 `visible()`

```
def anise.features.projectdescriptioneditor.packages.DebPackageAddExecLinkCustomAction.visible  
(  
    self,  
    ais,  
    node )
```

1.170.3 Member Data Documentation

1.170.3.1 `function`

`anise.features.projectdescriptioneditor.CustomAction.function` [inherited]

1.170.3.2 `icon`

`anise.features.projectdescriptioneditor.CustomAction.icon` [inherited]

1.170.3.3 `label`

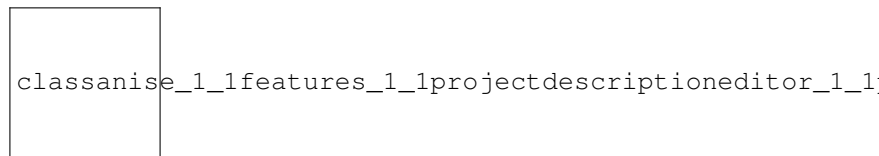
`anise.features.projectdescriptioneditor.CustomAction.label` [inherited]

The documentation for this class was generated from the following file:

- `anise/features/projectdescriptioneditor/packages.py`

1.171 anise.features.projectdescriptioneditor.packages.DebPackageAddStartMenuEntry Class Reference

Inheritance diagram for anise.features.projectdescriptioneditor.packages.DebPackageAddStartMenuEntry:



Public Member Functions

- def [__init__](#) (self)
- def [visible](#) (self, ais, node)
- def [execute](#) (self, ais, node)

Public Attributes

- [label](#)
- [function](#)
- [icon](#)

1.171.1 Constructor & Destructor Documentation

1.171.1.1 [__init__\(\)](#)

```
def anise.features.projectdescriptioneditor.packages.DebPackageAddStartMenuEntry.__init__ (
    self )
```

1.171.2 Member Function Documentation

1.171.2.1 [execute\(\)](#)

```
def anise.features.projectdescriptioneditor.packages.DebPackageAddStartMenuEntry.execute (
    self,
    ais,
    node )
```

1.171.2.2 visible()

```
def anise.features.projectdescriptioneditor.packages.DebPackageAddStartMenuEntry.visible (
    self,
    ais,
    node )
```

1.171.3 Member Data Documentation

1.171.3.1 function

```
anise.features.projectdescriptioneditor.CustomAction.function [inherited]
```

1.171.3.2 icon

```
anise.features.projectdescriptioneditor.CustomAction.icon [inherited]
```

1.171.3.3 label

```
anise.features.projectdescriptioneditor.CustomAction.label [inherited]
```

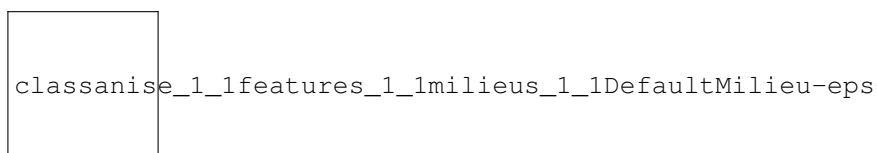
The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/packages.py](#)

1.172 anise.features.milieus.DefaultMilieu Class Reference

The default milieu.

Inheritance diagram for `anise.features.milieus.DefaultMilieu`:



Public Member Functions

- def [initialize](#) (self)
Initializes the milieu (composite).
- def [call](#) (self, cmdline, shell=False, decode=True, raise_on_errors=None, split_output_channels=False)
Executes a command line.
- def [createtempdir](#) (self, dirname=None)
Returns a fresh [anise.framework.files.TempDir](#).

1.172.1 Detailed Description

The default milieu.

1.172.2 Member Function Documentation

1.172.2.1 [call\(\)](#)

```
def anise.features.milieus.Milieu.call (
    self,
    cmdline,
    shell = False,
    decode = True,
    raise_on_errors = None,
    split_output_channels = False ) [inherited]
```

Executes a command line.

Parameters

<i>cmdline</i>	The command line as list of strings.
<i>shell</i>	If the command shall be interpreted by a shell (only the first part of cmdline will apply!).
<i>decode</i>	If the result is to be decoded to a string.
<i>raise_on_errors</i>	If errors shall raise Exceptions (also specifies the exception text in this case).
<i>split_output_channels</i>	If to separately record output and error stream.

Returns

: Tuple of returncode, program output.

1.172.2.2 [createtempdir\(\)](#)

```
def anise.features.milieus.Milieu.createtempdir (
    self,
    dirname = None ) [inherited]
```

Returns a fresh [anise.framework.files.TempDir](#).

Parameters

<i>dirname</i>	An optional name which the root directory should get; if not given, it gets a random name.
----------------	--

1.172.2.3 initialize()

```
def anise.features.milieus.Milieu.initialize (
    self ) [inherited]
```

Initializes the milieu (composite).

Override this method in custom subclasses or leave the default implementation.

The documentation for this class was generated from the following file:

- anise/features/[milieus.py](#)

1.173 anise.features.dependencies.Dependency Class Reference

A dependency.

Inheritance diagram for anise.features.dependencies.Dependency:

```
classanise_1_1features_1_1dependencies_1_1Dependency-
```

Public Member Functions

- def `__init__` (self, [type](#), [objectname](#)=None, [comment](#)="", [icon](#)=None, [visible](#)=True, [displayname](#)=None, [kwargs](#))

Public Attributes

- [deptype](#)
- [type](#)
- [objectname](#)
- [comment](#)
- [icon](#)
- [visible](#)
- [displayname](#)

1.173.1 Detailed Description

A dependency.

1.173.2 Constructor & Destructor Documentation

1.173.2.1 `__init__()`

```
def anise.features.dependencies.Dependency.__init__ (
    self,
    type,
    objectname = None,
    comment = "",
    icon = None,
    visible = True,
    displayname = None,
    kwargs )
```

Parameters

<i>type</i>	The dependency type (see Type).
<i>objectname</i>	A descriptive name for this dependency (e.g. a library name).
<i>comment</i>	A comment.
<i>icon</i>	The name of a dependency icon.
<i>visible</i>	If it is visible in the dependency list.
<i>displayname</i>	A display variant of the object name.
<i>kwargs</i>	Additional arguments.

1.173.3 Member Data Documentation

1.173.3.1 `comment`

`anise.features.dependencies.Dependency.comment`

1.173.3.2 `deptype`

`anise.features.dependencies.Dependency.deptype`

1.173.3.3 displayname

```
anise.features.dependencies.Dependency.displayname
```

1.173.3.4 icon

```
anise.features.dependencies.Dependency.icon
```

1.173.3.5 objectname

```
anise.features.dependencies.Dependency.objectname
```

1.173.3.6 type

```
anise.features.dependencies.Dependency.type
```

1.173.3.7 visible

```
anise.features.dependencies.Dependency.visible
```

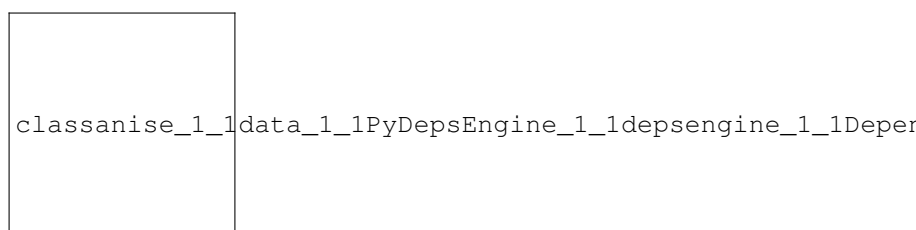
The documentation for this class was generated from the following file:

- [anise/features/dependencies.py](#)

1.174 anise.data.PyDepsEngine.depsengine.DependencyUnresolvedError Class Reference

Exception for unresolvable dependencies in dependency resolution.

Inheritance diagram for `anise.data.PyDepsEngine.depsengine.DependencyUnresolvedError`:



Public Member Functions

- `def __init__(self, o, req)`
- `def __call__(self, args)`

Public Attributes

- `o`
- `req`

1.174.1 Detailed Description

Exception for unresolvable dependencies in dependency resolution.

1.174.2 Constructor & Destructor Documentation

1.174.2.1 __init__()

```
def anise.data.PyDepsEngine.depsengine.DependencyUnresolvedError.__init__ (
    self,
    o,
    req )
```

1.174.3 Member Function Documentation

1.174.3.1 __call__()

```
def anise.data.PyDepsEngine.depsengine.EngineError.__call__ (
    self,
    args ) [inherited]
```

1.174.4 Member Data Documentation

1.174.4.1 o

```
anise.data.PyDepsEngine.depsengine.DependencyUnresolvedError.o
```

1.174.4.2 req

`anise.data.PyDepsEngine.depsengine.DependencyUnresolvedError.req`

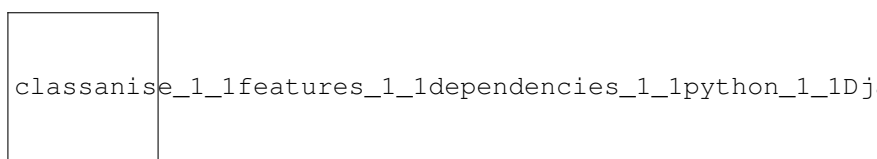
The documentation for this class was generated from the following file:

- `anise/data/PyDepsEngine/depsengine.py`

1.175 anise.features.dependencies.python.DjangoDependency Class Reference

A dependency describing the required Django framework installation.

Inheritance diagram for `anise.features.dependencies.python.DjangoDependency`:



Public Member Functions

- `def __init__(self, type, version, kwargs)`

Public Attributes

- `deptype`
- `type`
- `objectname`
- `comment`
- `icon`
- `visible`
- `displayname`

1.175.1 Detailed Description

A dependency describing the required Django framework installation.

1.175.2 Constructor & Destructor Documentation

1.175.2.1 __init__()

```
def anise.features.dependencies.python.DjangoDependency.__init__(  
    self,  
    type,  
    version,  
    kwargs )
```

Parameters

<i>type</i>	The dependency type (see dependencies.Type).
<i>version</i>	The version of the required Django framework (as string).
<i>kwargs</i>	Additional dependency infos.

1.175.3 Member Data Documentation

1.175.3.1 comment

`anise.features.dependencies.Dependency.comment` [inherited]

1.175.3.2 deptype

`anise.features.dependencies.Dependency.deptype` [inherited]

1.175.3.3 displayname

`anise.features.dependencies.Dependency.displayname` [inherited]

1.175.3.4 icon

`anise.features.dependencies.Dependency.icon` [inherited]

1.175.3.5 objectname

`anise.features.dependencies.Dependency.objectname` [inherited]

1.175.3.6 type

`anise.features.dependencies.Dependency.type` [inherited]

1.175.3.7 visible

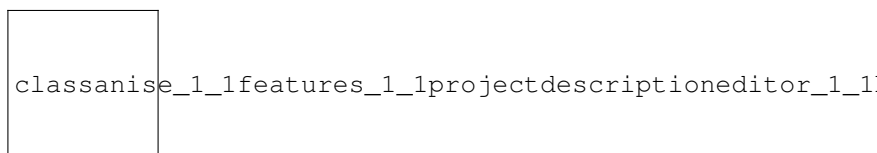
`anise.features.dependencies.Dependency.visible` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/dependencies/python.py](#)

1.176 `anise.features.projectdescriptioneditor.base.EditCodeExternalEditorCustom`↔ Action Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.base.EditCodeExternalEditorCustomAction`:



Public Member Functions

- `def __init__ (self)`
- `def visible (self, ais, node)`
- `def execute (self, ais, node)`

Public Attributes

- [label](#)
- [function](#)
- [icon](#)

1.176.1 Constructor & Destructor Documentation

1.176.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.base.EditCodeExternalEditorCustomAction.__init__ (
    self )
```

1.176.2 Member Function Documentation

1.176.2.1 execute()

```
def anise.features.projectdescriptioneditor.base.EditCodeExternalEditorCustomAction.execute (
    self,
    ais,
    node )
```

1.176.2.2 visible()

```
def anise.features.projectdescriptioneditor.base.EditCodeExternalEditorCustomAction.visible (
    self,
    ais,
    node )
```

1.176.3 Member Data Documentation

1.176.3.1 function

anise.features.projectdescriptioneditor.CustomAction.function [inherited]

1.176.3.2 icon

anise.features.projectdescriptioneditor.CustomAction.icon [inherited]

1.176.3.3 label

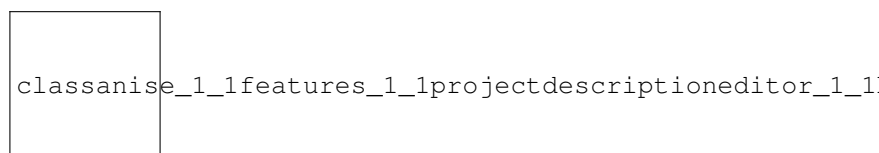
anise.features.projectdescriptioneditor.CustomAction.label [inherited]

The documentation for this class was generated from the following file:

- anise/features/projectdescriptioneditor/[base.py](#)

1.177 anise.features.projectdescriptioneditor.base.EditCodeInlineCustomAction Class Reference

Inheritance diagram for anise.features.projectdescriptioneditor.base.EditCodeInlineCustomAction:



Public Member Functions

- def `__init__` (self)
- def `visible` (self, ais, node)
- def `execute` (self, ais, node)

Public Attributes

- `label`
- `function`
- `icon`

1.177.1 Constructor & Destructor Documentation

1.177.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.base.EditCodeInlineCustomAction.__init__ (  
    self )
```

1.177.2 Member Function Documentation

1.177.2.1 `execute()`

```
def anise.features.projectdescriptioneditor.base.EditCodeInlineCustomAction.execute (  
    self,  
    ais,  
    node )
```

1.177.2.2 `visible()`

```
def anise.features.projectdescriptioneditor.base.EditCodeInlineCustomAction.visible (  
    self,  
    ais,  
    node )
```

1.177.3 Member Data Documentation

1.177.3.1 function

`anise.features.projectdescriptioneditor.CustomAction.function` [inherited]

1.177.3.2 icon

`anise.features.projectdescriptioneditor.CustomAction.icon` [inherited]

1.177.3.3 label

`anise.features.projectdescriptioneditor.CustomAction.label` [inherited]

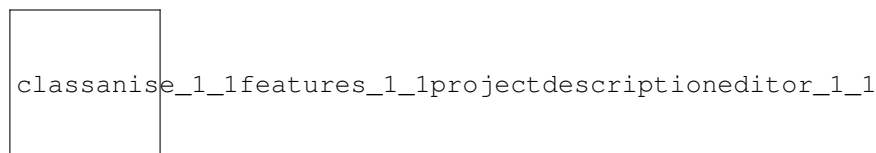
The documentation for this class was generated from the following file:

- `anise/features/projectdescriptioneditor/base.py`

1.178 anise.features.projectdescriptioneditor.internals.EditProjectdescApplication Class Reference

The raw mode editor.

Inheritance diagram for `anise.features.projectdescriptioneditor.internals.EditProjectdescApplication`:



Public Member Functions

- `def __init__ (self)`
- `def summarizeentry (self, e)`
- `def init (self, ais=None)`
Initializes the editor.
- `def execute (self)`
Executes the editor.
- `def addletentry (self)`
- `def changevalue (self, entry)`
- `def addvalentry (self, key=None, getkeyfromargs=False)`
- `def movelistchildentry (self, entryargs, fct_move)`
- `def changeentry (self, entry, remove_is_unsafe=True)`
- `def mainmenu (self)`
- `def visible (self)`
Checks if this feature action shall be visible.

Static Public Member Functions

- def `editnode` (`ais`, `node`)
Runs an editor for a particular node.

Public Attributes

- `ais`
- `label`
- `description`
- `forfeature`

Private Member Functions

- def `_store` (`self`)

1.178.1 Detailed Description

The raw mode editor.

It offers a configuration interface, which corresponds to the file format, but also custom actions for some nodes, which may be helpful for configuration.

1.178.2 Constructor & Destructor Documentation

1.178.2.1 `__init__()`

```
def anise.features.projectdescriptioneditor.internals.EditProjectdescApplication.__init__ (  
    self )
```

1.178.3 Member Function Documentation

1.178.3.1 `_store()`

```
def anise.features.projectdescriptioneditor.internals.EditProjectdescApplication._store (  
    self ) [private]
```

1.178.3.2 addletentry()

```
def anise.features.projectdescriptioneditor.internals.EditProjectdescApplication.addletentry (
    self )
```

1.178.3.3 addvalentry()

```
def anise.features.projectdescriptioneditor.internals.EditProjectdescApplication.addvalentry (
    self,
    key = None,
    getkeyfromargs = False )
```

1.178.3.4 changeentry()

```
def anise.features.projectdescriptioneditor.internals.EditProjectdescApplication.changeentry (
    self,
    entry,
    remove_is_unsafe = True )
```

1.178.3.5 changevalue()

```
def anise.features.projectdescriptioneditor.internals.EditProjectdescApplication.changevalue
(
    self,
    entry )
```

1.178.3.6 editnode()

```
def anise.features.projectdescriptioneditor.internals.EditProjectdescApplication.editnode (
    ais,
    node ) [static]
```

Runs an editor for a particular node.

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.178.3.7 execute()

```
def anise.features.projectdescriptioneditor.internals.EditProjectdescApplication.execute (
    self )
```

Executes the editor.

1.178.3.8 init()

```
def anise.features.projectdescriptioneditor.internals.EditProjectdescApplication.init (
    self,
    ais = None )
```

Initializes the editor.

This is required if you don't want to execute the editor but only use internal parts of it.

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
------------	---

1.178.3.9 mainmenu()

```
def anise.features.projectdescriptioneditor.internals.EditProjectdescApplication.mainmenu (
    self )
```

1.178.3.10 movelistchildentry()

```
def anise.features.projectdescriptioneditor.internals.EditProjectdescApplication.movelistchildentry
(
    self,
    entryargs,
    fct_move )
```

1.178.3.11 summarizeentry()

```
def anise.features.projectdescriptioneditor.internals.EditProjectdescApplication.summarizeentry
(
    self,
    e )
```

1.178.3.12 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.178.4 Member Data Documentation

1.178.4.1 ais

```
anise.features.projectdescriptioneditor.internals.EditProjectdescApplication.ais
```

1.178.4.2 description

```
anise.features.diagnostics.FeatureAction.description [inherited]
```

1.178.4.3 forfeature

```
anise.features.diagnostics.FeatureAction.forfeature [inherited]
```

1.178.4.4 label

```
anise.features.diagnostics.FeatureAction.label [inherited]
```

The documentation for this class was generated from the following file:

- anise/features/[projectdescriptioneditor.py](#)

1.179 anise.data.PyDepsEngine.depsengine.Engine Class Reference

A dependency engine manages a list of entity objects.

Public Member Functions

- `def __init__` (self, defaultprocessorclass=[Processor](#), defaultprocessorcfg=None, [stagesymbol_PRE](#)="PRE", [stagesymbol_DEFAULT](#)="DEFAULT", [stagesymbol_POST](#)="POST")
- `def add_object` (self, o, own=None, afterrequired=None, beforerequired=None, afteroptional=None, beforeoptional=None)
Adds an entity object to the engine.
- `def remove_object` (self, o)
Removes an entity object from the engine (all occurrences).
- `def add_processor` (self, processorclass=None, cfg)
Creates and adds a [Processor](#) to the engine, which in turn can return a list of entity objects processed regarding their dependencies.
- `def get_objects` (self, processorclass=None, cfg)
Returns the list of entity objects processed by the default dependencies processor.

Public Attributes

- [stagesymbol_PRE](#)
- [stagesymbol_DEFAULT](#)
- [stagesymbol_POST](#)

Private Attributes

- [_objects](#)
- [_processors](#)
- [_defaultprocessorclass](#)
- [_defaultprocessorcfg](#)

1.179.1 Detailed Description

A dependency engine manages a list of entity objects.

Depending on how it's configured (or subclassed), it holds a list of entity objects populated by [Engine.add_object](#).

On top of that, methods like [Engine.get_objects](#), [Engine.add_processor](#) and [Engine.create_compound_object](#) help getting a list sorted according to their dependencies.

1.179.2 Constructor & Destructor Documentation

1.179.2.1 __init__()

```
def anise.data.PyDepsEngine.depsengine.Engine.__init__ (
    self,
    defaultprocessorclass = Processor,
    defaultprocessorcfg = None,
    stagesymbol_PRE = "PRE",
    stagesymbol_DEFAULT = "DEFAULT",
    stagesymbol_POST = "POST" )
```


Parameters

<i>defaultprocessorclass</i>	The processor class to be used per default.
<i>defaultprocessorcfg</i>	The configuration applied per default to new processors (in their constructors).
<i>stagesymbol_PRE</i>	The name of the special symbol for earlier-than-the-rest objects. This is one of some special symbols for usage in the <code>own</code> info (see Engine.add_object). Each represent one stage of execution. You should only need them in exotic situations.
<i>stagesymbol_DEFAULT</i>	The name of the special symbol for the default staged objects (see above).
<i>stagesymbol_POST</i>	The name of the special symbol for later-than-the-rest objects (see above).

1.179.3 Member Function Documentation

1.179.3.1 add_object()

```
def anise.data.PyDepsEngine.depsengine.Engine.add_object (
    self,
    o,
    own = None,
    afterrequired = None,
    beforerequired = None,
    afteroptional = None,
    beforeoptional = None )
```

Adds an entity object to the engine.

All parameters marked with "Dependency info" are part of the dependency details of this entity object. Each is a list of symbols; either as an actual list of strings, or a comma-separated string (some other delimiters exist), or a function returning such objects. Note: Dependency processors are free to add and remove any parts in their internal view.

Parameters

<i>o</i>	The entity object.
<i>own</i>	Dependency info. List of symbols this entity object provides. Other entity objects can depend on those symbols somehow (see the other parameters). The list may contain a 'stage symbol' for some exotic placements (see Engine.__init__).
<i>afterrequired</i>	Dependency info. List of symbols this entity object depends on. Each object providing one of those symbols must be ordered before <code>o</code> . For each symbol there must exist at least one object providing it.
<i>beforerequired</i>	Dependency info. List of symbols this entity object reversely depends on. Each object providing one of those symbols must be ordered after <code>o</code> . For each symbol there must exist at least one object providing it.
<i>afteroptional</i>	Dependency info. List of symbols this entity object optionally depends on. Each object providing one of those symbols must be ordered before <code>o</code> . However, it is not an error if some of those symbol aren't provided by any object.
<i>beforeoptional</i>	Dependency info. List of symbols this entity object optionally reversely depends on. Each object providing one of those symbols must be ordered after <code>o</code> . However, it is not an error if some of those symbol aren't provided by any object.

1.179.3.2 add_processor()

```
def anise.data.PyDepsEngine.depsengine.Engine.add_processor (
    self,
    processorclass = None,
    cfg )
```

Creates and adds a [Processor](#) to the engine, which in turn can return a list of entity objects processed regarding their dependencies.

Parameters

<i>processorclass</i>	The Processor subclass to use (otherwise a default is used).
<i>cfg</i>	Additional configuration (passed to the constructor).

1.179.3.3 get_objects()

```
def anise.data.PyDepsEngine.depsengine.Engine.get_objects (
    self,
    processorclass = None,
    cfg )
```

Returns the list of entity objects processed by the default dependencies processor.

Parameters

<i>processorclass</i>	The Processor subclass to use (otherwise a default is used).
<i>cfg</i>	Additional configuration (passed to the constructor).

1.179.3.4 remove_object()

```
def anise.data.PyDepsEngine.depsengine.Engine.remove_object (
    self,
    o )
```

Removes an entity object from the engine (all occurrences).

Parameters

<i>o</i>	The entity object.
----------	--------------------

1.179.4 Member Data Documentation

1.179.4.1 `_defaultprocessorcfg`

`anise.data.PyDepsEngine.depsengine.Engine._defaultprocessorcfg` [private]

1.179.4.2 `_defaultprocessorclass`

`anise.data.PyDepsEngine.depsengine.Engine._defaultprocessorclass` [private]

1.179.4.3 `_objects`

`anise.data.PyDepsEngine.depsengine.Engine._objects` [private]

1.179.4.4 `_processors`

`anise.data.PyDepsEngine.depsengine.Engine._processors` [private]

1.179.4.5 `stagesymbol_DEFAULT`

`anise.data.PyDepsEngine.depsengine.Engine.stagesymbol_DEFAULT`

1.179.4.6 `stagesymbol_POST`

`anise.data.PyDepsEngine.depsengine.Engine.stagesymbol_POST`

1.179.4.7 `stagesymbol_PRE`

`anise.data.PyDepsEngine.depsengine.Engine.stagesymbol_PRE`

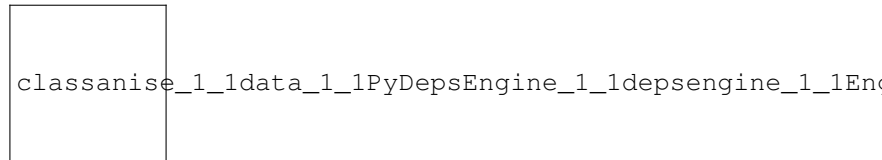
The documentation for this class was generated from the following file:

- `anise/data/PyDepsEngine/depsengine.py`

1.180 `anise.data.PyDepsEngine.depsengine.EngineError` Class Reference

Exceptions for all kinds of errors happening inside [PyDepsEngine](#).

Inheritance diagram for `anise.data.PyDepsEngine.depsengine.EngineError`:



Public Member Functions

- `def __call__ (self, args)`

1.180.1 Detailed Description

Exceptions for all kinds of errors happening inside [PyDepsEngine](#).

See subclasses.

1.180.2 Member Function Documentation

1.180.2.1 `__call__()`

```
def anise.data.PyDepsEngine.depsengine.EngineError.__call__ (
    self,
    args )
```

The documentation for this class was generated from the following file:

- `anise/data/PyDepsEngine/depsengine.py`

1.181 `anise.features.packages.appc.EnvironmentVariableDescription` Class Reference

Description for Appc environment variables to be added to the container image.

Public Member Functions

- `def __init__ (self, name, value)`

Public Attributes

- [name](#)
- [value](#)

1.181.1 Detailed Description

Description for Appc environment variables to be added to the container image.

1.181.2 Constructor & Destructor Documentation

1.181.2.1 `__init__()`

```
def anise.features.packages.appc.EnvironmentVariableDescription.__init__ (
    self,
    name,
    value )
```

Parameters

<i>name</i>	The variable name.
<i>value</i>	The value.

1.181.3 Member Data Documentation

1.181.3.1 `name`

`anise.features.packages.appc.EnvironmentVariableDescription.name`

1.181.3.2 `value`

`anise.features.packages.appc.EnvironmentVariableDescription.value`

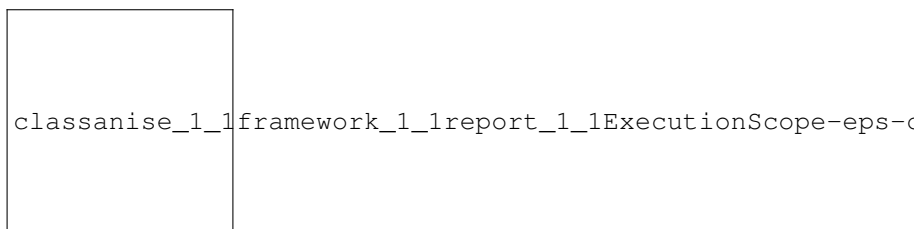
The documentation for this class was generated from the following file:

- [anise/features/packages/appc.py](#)

1.182 anise.framework.report.ExecutionScope Class Reference

An execution scope is one piece of work to do in a complete task executions.

Inheritance diagram for anise.framework.report.ExecutionScope:



Classes

- class [Progress](#)
Progress info for an execution scope.
- class [ProgressDeterminate](#)
Progress info for a determinate progress.
- class [ProgressFailed](#)
Progress info for a failed execution.
- class [ProgressFinished](#)
Progress info for a finished execution.
- class [ProgressIndeterminate](#)
Progress info for an indeterminate progress.

Public Member Functions

- def [__init__](#) (self, [title](#))
- def [appendchild](#) (self, [child](#))
Appends a child scope to this scope.
- def [progress](#) (self)
Returns the current progress state of this scope.
- def [setprogress](#) (self, [value](#))
Sets the progress of this scope.
- def [title](#) (self)
Returns the title string of this scope.
- def [rawsubparts](#) (self)
Returns a list of all direct subparts of this scope.
- def [newsubscope](#) (self, [title](#))
Adds a new sub scope to the current top scope.
- def [topscope](#) (self, [i=0](#))
Returns a scope from the top of the stack (without removing it).
- def [__enter__](#) (self)
- def [__exit__](#) (self, [etype](#), [evalue](#), [etraceback](#))
- def [id](#) (self)
Returns the ID of this part.
- def [value](#) (self)

- Returns the content value of this execution scope.*
- def [hints](#) (self)
Returns the content hint.
- def [address](#) (self)
Returns the address of this part.
- def [getscopepartsbyidrange](#) (self, minid=None, maxid=None)
Gets a list of execution scope parts by a range of IDs.

Private Attributes

- [_subparts](#)
- [_progress](#)
- [_title](#)

1.182.1 Detailed Description

An execution scope is one piece of work to do in a complete task executions.

It is the parent of another execution scope (just the root scope doesn't have a parent) and can have child scopes. This leads to a tree structure of scope nodes. Each node can have metadata infos like a title or a progress. Those are displayed during the execution of a task (together with other [ExecutionScopePart](#) subclasses' instances)

1.182.2 Constructor & Destructor Documentation

1.182.2.1 `__init__()`

```
def anise.framework.report.ExecutionScope.__init__ (
    self,
    title )
```

Parameters

<i>title</i>	The scope title text.
--------------	-----------------------

1.182.3 Member Function Documentation

1.182.3.1 `__enter__()`

```
def anise.framework.report.ExecutionScope.__enter__ (
    self )
```

1.182.3.2 `__exit__()`

```
def anise.framework.report.ExecutionScope.__exit__ (
    self,
    etype,
    evalue,
    etraceback )
```

1.182.3.3 `address()`

```
def anise.framework.report.ExecutionScopePart.address (
    self ) [inherited]
```

Returns the address of this part.

1.182.3.4 `appendchild()`

```
def anise.framework.report.ExecutionScope.appendchild (
    self,
    child )
```

Appends a child scope to this scope.

1.182.3.5 `getscopepartsbyidrange()`

```
def anise.framework.report.ExecutionScopePart.getscopepartsbyidrange (
    self,
    minid = None,
    maxid = None ) [inherited]
```

Gets a list of execution scope parts by a range of IDs.

1.182.3.6 `hints()`

```
def anise.framework.report.ExecutionScopePart.hints (
    self ) [inherited]
```

Returns the content hint.

This is typically used for storing the severity of log messages, so the presentation can differ.

1.182.3.7 id()

```
def anise.framework.report.ExecutionScopePart.id (
    self ) [inherited]
```

Returns the ID of this part.

1.182.3.8 newsubscope()

```
def anise.framework.report.ExecutionScope.newsubscope (
    self,
    title )
```

Adds a new sub scope to the current top scope.

Parameters

<i>title</i>	The scope title text.
--------------	-----------------------

1.182.3.9 progress()

```
def anise.framework.report.ExecutionScope.progress (
    self )
```

Returns the current progress state of this scope.

Returns

: An instance of [ExecutionScope.Progress](#).

1.182.3.10 rawsubparts()

```
def anise.framework.report.ExecutionScope.rawsubparts (
    self )
```

Returns a list of all direct subparts of this scope.

1.182.3.11 setprogress()

```
def anise.framework.report.ExecutionScope.setprogress (
    self,
    value )
```

Sets the progress of this scope.

Parameters

<i>value</i>	An instance of ExecutionScope.Progress .
--------------	--

1.182.3.12 title()

```
def anise.framework.report.ExecutionScope.title (
    self )
```

Returns the title string of this scope.

1.182.3.13 topscope()

```
def anise.framework.report.ExecutionScope.topscope (
    self,
    i = 0 )
```

Returns a scope from the top of the stack (without removing it).

Parameters

<i>i</i>	Which scope to return (counting from top).
----------	--

1.182.3.14 value()

```
def anise.framework.report.ExecutionScopePart.value (
    self ) [inherited]
```

Returns the content value of this execution scope.

1.182.4 Member Data Documentation**1.182.4.1 _progress**

```
anise.framework.report.ExecutionScope._progress [private]
```

1.182.4.2 `_subparts`

`anise.framework.report.ExecutionScope._subparts` [private]

1.182.4.3 `_title`

`anise.framework.report.ExecutionScope._title` [private]

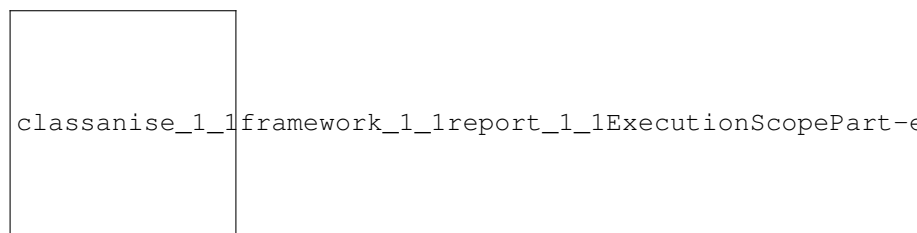
The documentation for this class was generated from the following file:

- `anise/framework/report.py`

1.183 `anise.framework.report.ExecutionScopePart` Class Reference

Base class of `ExecutionScope` (read there for more).

Inheritance diagram for `anise.framework.report.ExecutionScopePart`:



Public Member Functions

- `def __init__(self, value, hints=None)`
- `def id(self)`
Returns the ID of this part.
- `def value(self)`
Returns the content value of this execution scope.
- `def hints(self)`
Returns the content hint.
- `def address(self)`
Returns the address of this part.
- `def getscopepartsbyidrange(self, minid=None, maxid=None)`
Gets a list of execution scope parts by a range of IDs.

Private Attributes

- `_rootscope`
- `_address`
- `_id`
- `_id_counter`
- `_scopepartsbyid`
- `_topscope`
- `_value`
- `_hints`

1.183.1 Detailed Description

Base class of [ExecutionScope](#) (read there for more).

It also includes logging output or output from external programs, which are displayed together with real execution scopes during task execution.

1.183.2 Constructor & Destructor Documentation

1.183.2.1 `__init__()`

```
def anise.framework.report.ExecutionScopePart.__init__ (
    self,
    value,
    hints = None )
```

Parameters

<i>value</i>	The content value.
<i>hints</i>	A list of hint symbols (typically used for log message severity).

1.183.3 Member Function Documentation

1.183.3.1 `address()`

```
def anise.framework.report.ExecutionScopePart.address (
    self )
```

Returns the address of this part.

1.183.3.2 `getscopepartsbyidrange()`

```
def anise.framework.report.ExecutionScopePart.getscopepartsbyidrange (
    self,
    minid = None,
    maxid = None )
```

Gets a list of execution scope parts by a range of IDs.

1.183.3.3 hints()

```
def anise.framework.report.ExecutionScopePart.hints (  
    self )
```

Returns the content hint.

This is typically used for storing the severity of log messages, so the presentation can differ.

1.183.3.4 id()

```
def anise.framework.report.ExecutionScopePart.id (  
    self )
```

Returns the ID of this part.

1.183.3.5 value()

```
def anise.framework.report.ExecutionScopePart.value (  
    self )
```

Returns the content value of this execution scope.

1.183.4 Member Data Documentation

1.183.4.1 _address

```
anise.framework.report.ExecutionScopePart._address [private]
```

1.183.4.2 _hints

```
anise.framework.report.ExecutionScopePart._hints [private]
```

1.183.4.3 _id

```
anise.framework.report.ExecutionScopePart._id [private]
```

1.183.4.4 `_id_counter_`

```
anise.framework.report.ExecutionScopePart._id_counter_ [private]
```

1.183.4.5 `_rootscope`

```
anise.framework.report.ExecutionScopePart._rootscope [private]
```

1.183.4.6 `_scopepartsbyid`

```
anise.framework.report.ExecutionScopePart._scopepartsbyid [private]
```

1.183.4.7 `_topscope`

```
anise.framework.report.ExecutionScopePart._topscope [private]
```

1.183.4.8 `_value`

```
anise.framework.report.ExecutionScopePart._value [private]
```

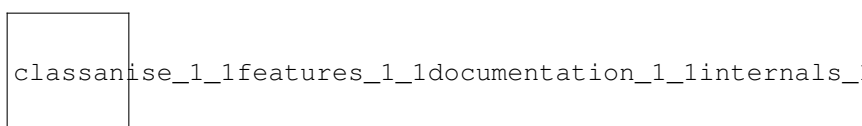
The documentation for this class was generated from the following file:

- [anise/framework/report.py](#)

1.184 `anise.features.documentation.internals.Export` Class Reference

Defines a documentation export.

Inheritance diagram for `anise.features.documentation.internals.Export`:



Public Member Functions

- def `__init__` (self, `key`, `format`, `name`=None, `heading`=None, `includedeveloperdoc`=False, `extractall`=True, `extractprivate`=True, `hideundocumented`=False, `imagepaths`=None, `additionaltext`="", `shownavbar`=False, `excludepatterns`=None, `predefineddocmacros`=None, `head3`=None, `head4`=None, `source`=None, `sources`=None)
- def `generate` (self, targetpath)
Generates the content.

Public Attributes

- `key`
- `format`
- `name`
- `heading`
- `includedeveloperdoc`
- `extractall`
- `extractprivate`
- `hideundocumented`
- `imagepaths`
- `additionaltext`
- `shownavbar`
- `excludepatterns`
- `predefineddocmacros`
- `head3`
- `head4`
- `source`
- `sources`

Private Member Functions

- def `_head3` (self)
- def `_head4` (self)

1.184.1 Detailed Description

Defines a documentation export.

1.184.2 Constructor & Destructor Documentation

1.184.2.1 `__init__()`

```
def anise.features.documentation.internals.Export.__init__ (
    self,
    key,
    format,
    name = None,
    heading = None,
    includedeveloperdoc = False,
    extractall = True,
    extractprivate = True,
    hideundocumented = False,
    imagepaths = None,
    additionaltext = "",
    shownavbar = False,
    excludepatterns = None,
    predefineddocmacros = None,
    head3 = None,
    head4 = None,
    source = None,
    sources = None )
```

Parameters

<i>key</i>	One of the keys stored in pool. Defines which documentation to export.
<i>format</i>	The output format (one of documentation.ExportFormat).
<i>name</i>	The export definition name. Finding definitions by name is possible with <code>exports.getbyname</code> .
<i>heading</i>	The title.
<i>includedeveloperdoc</i>	If the output shall include the developer documentation as well (does not work for all output formats).
<i>extractall</i>	If a developer documentation should include all elements (not just the documented ones).
<i>extractprivate</i>	If a developer documentation should include also private members.
<i>hideundocumented</i>	If the develo documentation should entirely leave out undocumented members.
<i>imagepaths</i>	Directories from where images are used in the documentation.
<i>additionaltext</i>	Additional text to append to the selected documentation.
<i>shownavbar</i>	If to show navigation bar. Does not work in all output formats.
<i>excludepatterns</i>	File patterns to exclude from reading.
<i>predefineddocmacros</i>	List of additional macros to define for reading.
<i>head3</i>	3rd heading.
<i>head4</i>	4th heading.
<i>source</i>	The documentation source root.
<i>sources</i>	Optional list of source paths (default: ["/"]).

1.184.3 Member Function Documentation

1.184.3.1 `_head3()`

```
def anise.features.documentation.internals.Export._head3 (
    self ) [private]
```

1.184.3.2 `_head4()`

```
def anise.features.documentation.internals.Export._head4 (
    self ) [private]
```

1.184.3.3 `generate()`

```
def anise.features.documentation.internals.Export.generate (
    self,
    targetpath )
```

Generates the content.

Override this method in custom subclasses or leave the default implementation.

1.184.4 Member Data Documentation

1.184.4.1 `additionaltext`

```
anise.features.documentation.internals.Export.additionaltext
```

1.184.4.2 `excludepatterns`

```
anise.features.documentation.internals.Export.excludepatterns
```

1.184.4.3 `extractall`

```
anise.features.documentation.internals.Export.extractall
```

1.184.4.4 `extractprivate`

`anise.features.documentation.internals.Export.extractprivate`

1.184.4.5 `format`

`anise.features.documentation.internals.Export.format`

1.184.4.6 `head3`

`anise.features.documentation.internals.Export.head3`

1.184.4.7 `head4`

`anise.features.documentation.internals.Export.head4`

1.184.4.8 `heading`

`anise.features.documentation.internals.Export.heading`

1.184.4.9 `hideundocumented`

`anise.features.documentation.internals.Export.hideundocumented`

1.184.4.10 `imagepaths`

`anise.features.documentation.internals.Export.imagepaths`

1.184.4.11 `includedeveloperdoc`

`anise.features.documentation.internals.Export.includedeveloperdoc`

1.184.4.12 key

`anise.features.documentation.internals.Export.key`

1.184.4.13 name

`anise.features.documentation.internals.Export.name`

1.184.4.14 predefineddocmacros

`anise.features.documentation.internals.Export.predefineddocmacros`

1.184.4.15 shownavbar

`anise.features.documentation.internals.Export.shownavbar`

1.184.4.16 source

`anise.features.documentation.internals.Export.source`

1.184.4.17 sources

`anise.features.documentation.internals.Export.sources`

The documentation for this class was generated from the following file:

- [anise/features/documentation.py](#)

1.185 anise.features.documentation.ExportFormat Class Reference

Enumeration of output formats.

Static Public Attributes

- string `Plaintext` = "Plaintext"
- string `PDF` = "PDF"
- string `HTML` = "HTML"

1.185.1 Detailed Description

Enumeration of output formats.

1.185.2 Member Data Documentation

1.185.2.1 HTML

```
string anise.features.documentation.ExportFormat.HTML = "HTML" [static]
```

1.185.2.2 PDF

```
string anise.features.documentation.ExportFormat.PDF = "PDF" [static]
```

1.185.2.3 Plaintext

```
string anise.features.documentation.ExportFormat.Plaintext = "Plaintext" [static]
```

The documentation for this class was generated from the following file:

- [anise/features/documentation.py](#)

1.186 `anise.features.documentation.internals.Exports` Class Reference

Storage of all documentation export definitions (i.e.

Public Member Functions

- def `__init__` (self)
- def `getbyname` (self, name)
Returns all export definitions with a given name.
- def `add` (self, export)
Exports a documentation to some targets.

Public Attributes

- [list](#)

1.186.1 Detailed Description

Storage of all documentation export definitions (i.e. where and how documentation products are to be created).

1.186.2 Constructor & Destructor Documentation

1.186.2.1 `__init__()`

```
def anise.features.documentation.internals.Exports.__init__ (
    self )
```

1.186.3 Member Function Documentation

1.186.3.1 `add()`

```
def anise.features.documentation.internals.Exports.add (
    self,
    export )
```

[Exports](#) a documentation to some targets.

Parameters

<i>export</i>	A export definition which specify an documentation export target (see Export).
---------------	---

1.186.3.2 `getbyname()`

```
def anise.features.documentation.internals.Exports.getbyname (
    self,
    name )
```

Returns all export definitions with a given name.

Parameters

<i>name</i>	A export definition name.
-------------	---------------------------

Returns

: List of export definitions.

1.186.4 Member Data Documentation**1.186.4.1 list**

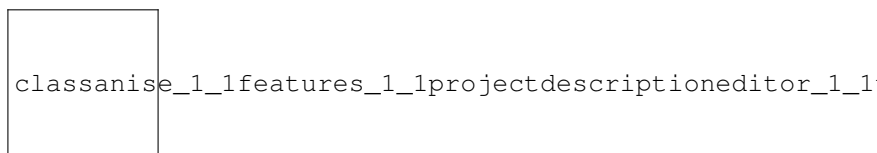
```
anise.features.documentation.internals.Exports.list
```

The documentation for this class was generated from the following file:

- [anise/features/documentation.py](#)

1.187 anise.features.projectdescriptioneditor.testing.ExternalProgramTestingImplementation Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.testing.ExternalProgramTestingImplementation`:

**Public Member Functions**

- `def __init__ (self)`
- `def execute (self, params)`

Public Attributes

- [label](#)

1.187.1 Constructor & Destructor Documentation

1.187.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.testing.ExternalProgramTestingImplementation.__init__ (
    self )
```

1.187.2 Member Function Documentation

1.187.2.1 `execute()`

```
def anise.features.projectdescriptioneditor.testing.ExternalProgramTestingImplementation.execute (
    self,
    params )
```

1.187.3 Member Data Documentation

1.187.3.1 `label`

```
anise.features.projectdescriptioneditor.testing.TestingImplementation.label [inherited]
```

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/testing.py](#)

1.188 anise.framework.features.Feature Class Reference

A feature is some encapsuled part of anise high-level functionality.

Public Member Functions

- def `__init__` (self, [name](#), hookhandlers, [featuremodule](#), [hooks](#))

Public Attributes

- [name](#)
- [featuremodule](#)
- [hooks](#)

Private Attributes

- [_hookhandlers](#)

1.188.1 Detailed Description

A feature is some encapsuled part of anise high-level functionality.

This class represents a loaded feature at runtime. The framework as well as other features access this object for interacting with that feature. Direct usage of high-level features from within `_projectdesc` is intended to not use this [Feature](#) objects but to use the embedded properties of the universe object.

1.188.2 Constructor & Destructor Documentation

1.188.2.1 `__init__()`

```
def anise.framework.features.Feature.__init__ (
    self,
    name,
    hookhandlers,
    featuremodule,
    hooks )
```

Parameters

<i>name</i>	The optional name of the feature.
<i>hookhandlers</i>	All hook handlers this feature registers globally (it may do that later on in a function as well).
<i>featuremodule</i>	The feature module.
<i>hooks</i>	List of Hook instances.

1.188.3 Member Data Documentation

1.188.3.1 `_hookhandlers`

```
anise.framework.features.Feature._hookhandlers [private]
```

1.188.3.2 `featuremodule`

```
anise.framework.features.Feature.featuremodule
```


1.188.3.3 hooks

```
anise.framework.features.Feature.hooks
```

1.188.3.4 name

```
anise.framework.features.Feature.name
```

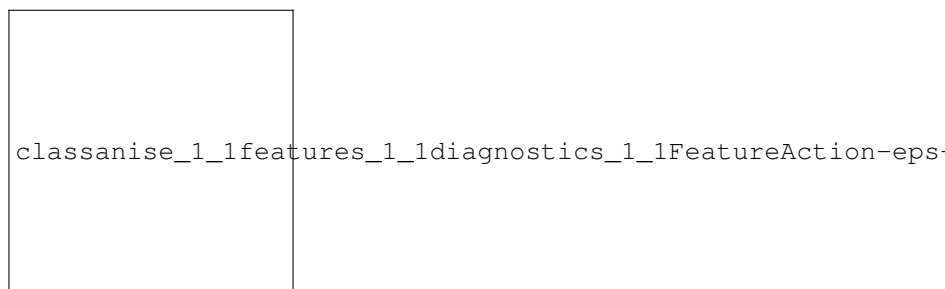
The documentation for this class was generated from the following file:

- [anise/framework/features.py](#)

1.189 anise.features.diagnostics.FeatureAction Class Reference

Abstract base class for an action, which can be triggered from the feature overview in the console.

Inheritance diagram for anise.features.diagnostics.FeatureAction:



Public Member Functions

- `def __init__(self, label, description, forfeature)`
- `def execute(self)`
Executes this feature action.
- `def visible(self)`
Checks if this feature action shall be visible.

Public Attributes

- [label](#)
- [description](#)
- [forfeature](#)

1.189.1 Detailed Description

Abstract base class for an action, which can be triggered from the feature overview in the console.

1.189.2 Constructor & Destructor Documentation

1.189.2.1 `__init__()`

```
def anise.features.diagnostics.FeatureAction.__init__ (
    self,
    label,
    description,
    forfeature )
```

Parameters

<i>label</i>	The label text.
<i>description</i>	The description text (may be a bit longer than label).
<i>forfeature</i>	Full name of the feature to associate this action with (like 'foo.bar').

1.189.3 Member Function Documentation

1.189.3.1 `execute()`

```
def anise.features.diagnostics.FeatureAction.execute (
    self )
```

Executes this feature action.

Override this method in custom subclasses.

1.189.3.2 `visible()`

```
def anise.features.diagnostics.FeatureAction.visible (
    self )
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.189.4 Member Data Documentation

1.189.4.1 description

```
anise.features.diagnostics.FeatureAction.description
```

1.189.4.2 forfeature

```
anise.features.diagnostics.FeatureAction.forfeature
```

1.189.4.3 label

```
anise.features.diagnostics.FeatureAction.label
```

The documentation for this class was generated from the following file:

- [anise/features/diagnostics.py](#)

1.190 anise.features.ui.web.internals.UserFeedback.FeedbackRequest Class Reference

Public Member Functions

- `def __init__(self, index, request, data, requestor)`
- `def isactive(self)`
- `def tojson(self)`

Public Attributes

- [index](#)
- [request](#)
- [data](#)
- [requestor](#)

1.190.1 Constructor & Destructor Documentation

1.190.1.1 __init__()

```
def anise.features.ui.web.internals.UserFeedback.FeedbackRequest.__init__(  
    self,  
    index,  
    request,  
    data,  
    requestor )
```

1.190.2 Member Function Documentation

1.190.2.1 isactive()

```
def anise.features.ui.web.internals.UserFeedback.FeedbackRequest.isactive (
    self )
```

1.190.2.2 tojson()

```
def anise.features.ui.web.internals.UserFeedback.FeedbackRequest.tojson (
    self )
```

1.190.3 Member Data Documentation

1.190.3.1 data

```
anise.features.ui.web.internals.UserFeedback.FeedbackRequest.data
```

1.190.3.2 index

```
anise.features.ui.web.internals.UserFeedback.FeedbackRequest.index
```

1.190.3.3 request

```
anise.features.ui.web.internals.UserFeedback.FeedbackRequest.request
```

1.190.3.4 requestor

```
anise.features.ui.web.internals.UserFeedback.FeedbackRequest.requestor
```

The documentation for this class was generated from the following file:

- [anise/features/ui/web.py](#)

1.191 anise.features.distributables.FileGroup Class Reference

A file group.

Public Member Functions

- `def __init__` (self, [sources](#), [name](#), [description](#)=None, [linkto](#)=None)

Public Attributes

- [sources](#)
- [name](#)
- [description](#)
- [linkto](#)

1.191.1 Detailed Description

A file group.

1.191.2 Constructor & Destructor Documentation

1.191.2.1 __init__()

```
def anise.features.distributables.FileGroup.__init__ (
    self,
    sources,
    name,
    description = None,
    linkto = None )
```

Parameters

<i>sources</i>	List of task implementations which generates the download group content. For a single entry, you may also pass it directly without a list around it.
<i>name</i>	The file group name.
<i>description</i>	The file group description text.
<i>linkto</i>	An optional link path, which is always kept up to date (pointing to the latest version). This is only applied to the first file in the group.

1.191.3 Member Data Documentation

1.191.3.1 description

```
anise.features.distributables.FileGroup.description
```

1.191.3.2 linkto

```
anise.features.distributables.FileGroup.linkto
```

1.191.3.3 name

```
anise.features.distributables.FileGroup.name
```

1.191.3.4 sources

```
anise.features.distributables.FileGroup.sources
```

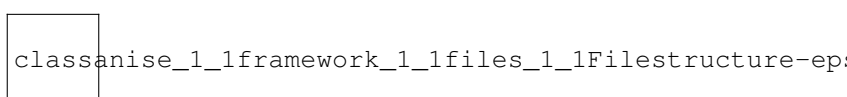
The documentation for this class was generated from the following file:

- [anise/features/distributables.py](#)

1.192 anise.framework.files.Filestructure Class Reference

This class represents a file or a directory structure with many subdirectories and files.

Inheritance diagram for `anise.framework.files.Filestructure`:



Public Member Functions

- def `__init__` (self, `path`, relative=False)
- def `datakeys` (self)
Returns a list of keys for all stored metadata properties.
- def `setdata` (self, k, v)
Sets a metadata property.
- def `getdata` (self, k, deflt)
Gets a metadata property.
- def `initialize` (self)
Initializes this [Filestructure](#).
- def `path` (self)
The path to this structure in the filesystem.
- def `dl` (self, subpath="", to=None, progresscallback=None)
Copies the complete filestructure or a subdirectory to a new destination.
- def `with_modified_rootname` (self, newname)
Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.
- def `mv` (self, newname)
Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.
- def `setautoopen` (self, `path`, interterminal=False)
Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Private Member Functions

- def `_initialize` (self)
The internal initialization routine, which fetches dynamic content (if any).

Private Attributes

- `_path`
- `_data`
- `_initialized`

1.192.1 Detailed Description

This class represents a file or a directory structure with many subdirectories and files.

There are methods for writing those items to disk and/or to modify the original structure in the filesystem.

It is more powerful for file exchange than working just with file paths, because a [Filestructure](#) can also carry arbitrary metadata information and allows dynamic content creation (see e.g. [TaskExecution](#) subclass).

Instances of this class (and its subclasses) are used by many features for file exchange.

1.192.2 Constructor & Destructor Documentation

1.192.2.1 `__init__()`

```
def anise.framework.files.Filestructure.__init__ (
    self,
    path,
    relative = False )
```

Parameters

<i>path</i>	Path to the file or directory.
<i>relative</i>	If path is relative to the project directory.

1.192.3 Member Function Documentation

1.192.3.1 `_initialize()`

```
def anise.framework.files.Filestructure._initialize (
    self ) [private]
```

The internal initialization routine, which fetches dynamic content (if any).

Called from outside before actual accesses occur.

Override this method in custom subclasses or leave the default implementation.

1.192.3.2 `datakeys()`

```
def anise.framework.files.Filestructure.datakeys (
    self )
```

Returns a list of keys for all stored metadata properties.

1.192.3.3 `dl()`

```
def anise.framework.files.Filestructure.dl (
    self,
    subpath = "",
    to = None,
    progresscallback = None )
```

Copies the complete filestructure or a subdirectory to a new destination.

Parameters

<i>subpath</i>	If given, only this subdirectory will be copied.
<i>to</i>	If given, it specifies the destination path; if not given, a new directory in the current working directory will be used.
<i>progresscallback</i>	Callback function for retrieving progress information.

1.192.3.4 getdata()

```
def anise.framework.files.Filestructure.getdata (
    self,
    k,
    deflt )
```

Gets a metadata property.

Parameters

<i>k</i>	The key name.
<i>deflt</i>	The default value (returned if the key doesn't exist).

1.192.3.5 initialize()

```
def anise.framework.files.Filestructure.initialize (
    self )
```

Initializes this [Filestructure](#).

When this is called, the dynamic content (if any) is created and written to disc. It is automatically called for typical usage and is not required to be called manually.

1.192.3.6 mv()

```
def anise.framework.files.Filestructure.mv (
    self,
    newname )
```

Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.

Parameters

<i>newname</i>	The new name (absolute or relative).
----------------	--------------------------------------

1.192.3.7 path()

```
def anise.framework.files.Filestructure.path (
    self )
```

The path to this structure in the filesystem.

1.192.3.8 setautoopen()

```
def anise.framework.files.Filestructure.setautoopen (
    self,
    path,
    interterminal = False )
```

Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Parameters

<i>path</i>	The relative path to the file which should automatically be opened.
<i>interterminal</i>	If this hint is for terminal (non-graphical) usage.

1.192.3.9 setdata()

```
def anise.framework.files.Filestructure.setdata (
    self,
    k,
    v )
```

Sets a metadata property.

Parameters

<i>k</i>	The key name.
<i>v</i>	The value.

1.192.3.10 with_modified_rootname()

```
def anise.framework.files.Filestructure.with_modified_rootname (
    self,
    newname )
```

Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.

Parameters

<i>newname</i>	The name which the new root directory should get.
----------------	---

1.192.4 Member Data Documentation

1.192.4.1 `_data`

```
anise.framework.files.Filestructure._data [private]
```

1.192.4.2 `_initialized`

```
anise.framework.files.Filestructure._initialized [private]
```

1.192.4.3 `_path`

```
anise.framework.files.Filestructure._path [private]
```

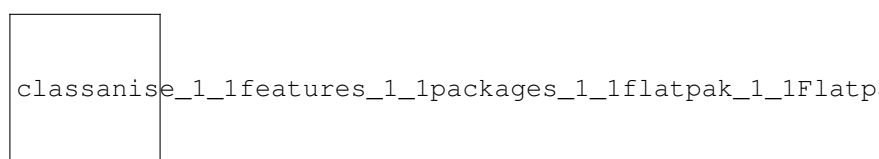
The documentation for this class was generated from the following file:

- [anise/framework/files.py](#)

1.193 anise.features.packages.flatpak.FlatpakMilieu Class Reference

Build environment for a new Flatpak image.

Inheritance diagram for `anise.features.packages.flatpak.FlatpakMilieu`:



Public Member Functions

- `def __init__` (self, [name](#), [sdk](#)="org.freedesktop.Sdk", [platform](#)="org.freedesktop.Platform", [kitversion](#)=None, [fullname](#)=None, [sockets](#)=["x11"], [filesystems](#)=["home"], [shares](#)=["network"], [command](#)=None, [menuentries](#)=None, [environment](#)=None, [flatpakpath](#)="flatpak", [gpg_signkeyid](#)="", [gpg_homedir](#)="", [gpg_exportto](#)="")
- `def initialize` (self)
- `def finish_build` (self)
- `def call` (self, args, [inside_sandbox](#)=True, [kwargs](#))
- `def createtempdir` (self, [dirname](#)=None)
- `def call_flatpak` (self, args, [raise_on_errors](#)="unable to run flatpak", [kwargs](#))
- `def deploy_to_repository` (self, [repository](#), [src](#)=None)
- `def call` (self, [cmdline](#), [shell](#)=False, [decode](#)=True, [raise_on_errors](#)=None, [split_output_channels](#)=False)

Executes a command line.

Public Attributes

- [name](#)
- [sdk](#)
- [platform](#)
- [kitversion](#)
- [fullname](#)
- [sockets](#)
- [filesystems](#)
- [shares](#)
- [command](#)
- [menuentries](#)
- [environment](#)
- [gpg_signkeyid](#)
- [gpg_homedir](#)
- [gpg_exportto](#)
- [pgpg](#)
- [flatpakpath](#)
- [flatpakcmd](#)

Private Attributes

- [_envrootpath](#)

1.193.1 Detailed Description

Build environment for a new Flatpak image.

Typically this is used automatically within [flatpak.flatpakpackage](#).

See [milieus.MilieuComposite](#) for general infos.

1.193.2 Constructor & Destructor Documentation

1.193.2.1 `__init__()`

```
def anise.features.packages.flatpak.FlatpakMilieu.__init__ (
    self,
    name,
    sdk = "org.freedesktop.Sdk",
    platform = "org.freedesktop.Platform",
    kitversion = None,
    fullname = None,
    sockets = ["x11"],
    filesystems = ["home"],
    shares = ["network"],
    command = None,
    menuentries = None,
    environment = None,
    flatpakpath = "flatpak",
    gpg_signkeyid = "",
    gpg_homedir = "",
    gpg_exportto = "" )
```

Parameters

<i>name</i>	The Flatpak image short name.
<i>sdk</i>	The sdk name, like "org.kde.Sdk".
<i>platform</i>	The platform name, like "org.kde.Platform".
<i>kitversion</i>	The version of the sdk and platform as string.
<i>fullname</i>	The full image name (dotted string).
<i>sockets</i>	The flatpak sockets to allow the application to access (as list of strings).
<i>filesystems</i>	The flatpak filesystems to allow the application to access (as list of strings).
<i>shares</i>	The flatpak shares to allow the application to access (as list of strings).
<i>command</i>	The optional command string. Must be a file in /app/bin inside the sandbox.
<i>menuentries</i>	The list of MenuEntry menu entries this image should add.
<i>environment</i>	Environment variables for within the sandbox (as dictionary).
<i>flatpakpath</i>	The path to the flatpak tool.
<i>gpg_signkeyid</i>	The gpg key id used for signing products.
<i>gpg_homedir</i>	For gpg_sign, this optionally specifies the directory which contains that key.
<i>gpg_exportto</i>	The optional target file name for making the gpg key available for download.

1.193.3 Member Function Documentation

1.193.3.1 `call()` [1/2]

```
def anise.features.milieus.Milieu.call (
    self,
    cmdline,
    shell = False,
    decode = True,
    raise_on_errors = None,
    split_output_channels = False ) [inherited]
```

Executes a command line.

Parameters

<i>cmdline</i>	The command line as list of strings.
<i>shell</i>	If the command shall be interpreted by a shell (only the first part of cmdline will apply!).
<i>decode</i>	If the result is to be decoded to a string.
<i>raise_on_errors</i>	If errors shall raise Exceptions (also specifies the exception text in this case).
<i>split_output_channels</i>	If to separately record output and error stream.

Returns

: Tuple of returncode, program output.

1.193.3.2 call() [2/2]

```
def anise.features.packages.flatpak.FlatpakMilieu.call (
    self,
    args,
    inside_sandbox = True,
    kwargs )
```

1.193.3.3 call_flatpak()

```
def anise.features.packages.flatpak.FlatpakMilieu.call_flatpak (
    self,
    args,
    raise_on_errors = "unable to run flatpak",
    kwargs )
```

1.193.3.4 createtempdir()

```
def anise.features.packages.flatpak.FlatpakMilieu.createtempdir (
    self,
    dirname = None )
```

1.193.3.5 deploy_to_repository()

```
def anise.features.packages.flatpak.FlatpakMilieu.deploy_to_repository (
    self,
    repository,
    src = None )
```

1.193.3.6 finish_build()

```
def anise.features.packages.flatpak.FlatpakMilieu.finish_build (
    self )
```

1.193.3.7 initialize()

```
def anise.features.packages.flatpak.FlatpakMilieu.initialize (
    self )
```

1.193.4 Member Data Documentation

1.193.4.1 `_envrootpath`

`anise.features.packages.flatpak.FlatpakMilieu._envrootpath` [private]

1.193.4.2 `command`

`anise.features.packages.flatpak.FlatpakMilieu.command`

1.193.4.3 `environment`

`anise.features.packages.flatpak.FlatpakMilieu.environment`

1.193.4.4 `filesystems`

`anise.features.packages.flatpak.FlatpakMilieu.filesystems`

1.193.4.5 `flatpakcmd`

`anise.features.packages.flatpak.FlatpakMilieu.flatpakcmd`

1.193.4.6 `flatpakpath`

`anise.features.packages.flatpak.FlatpakMilieu.flatpakpath`

1.193.4.7 `fullname`

`anise.features.packages.flatpak.FlatpakMilieu.fullname`

1.193.4.8 gpg_exportto

`anise.features.packages.flatpak.FlatpakMilieu.gpg_exportto`

1.193.4.9 gpg_homedir

`anise.features.packages.flatpak.FlatpakMilieu.gpg_homedir`

1.193.4.10 gpg_signkeyid

`anise.features.packages.flatpak.FlatpakMilieu.gpg_signkeyid`

1.193.4.11 kitversion

`anise.features.packages.flatpak.FlatpakMilieu.kitversion`

1.193.4.12 menuentries

`anise.features.packages.flatpak.FlatpakMilieu.menuentries`

1.193.4.13 name

`anise.features.packages.flatpak.FlatpakMilieu.name`

1.193.4.14 pgpg

`anise.features.packages.flatpak.FlatpakMilieu.pgpg`

1.193.4.15 platform

`anise.features.packages.flatpak.FlatpakMilieu.platform`

1.193.4.16 sdk

```
anise.features.packages.flatpak.FlatpakMilieu.sdk
```

1.193.4.17 shares

```
anise.features.packages.flatpak.FlatpakMilieu.shares
```

1.193.4.18 sockets

```
anise.features.packages.flatpak.FlatpakMilieu.sockets
```

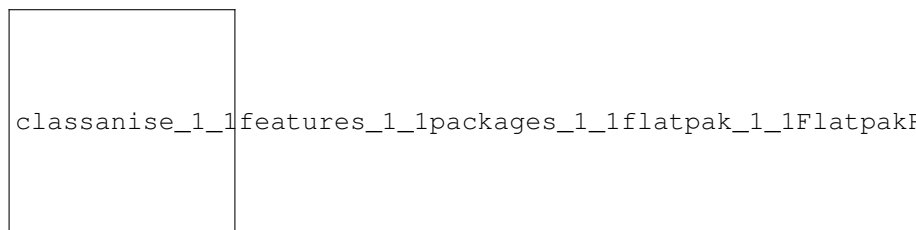
The documentation for this class was generated from the following file:

- [anise/features/packages/flatpak.py](#)

1.194 anise.features.packages.flatpak.FlatpakRef Class Reference

Description for creation of a Flatpak application reference (.flatpakref) file.

Inheritance diagram for anise.features.packages.flatpak.FlatpakRef:



Public Member Functions

- def `__init__` (self, `filename`=None, `title`=None, `branch`="master", `runtime``repourl`=None)
- def `generate` (self)
- def `set` (self, `fullname`, `repobaseurl`, `repourl`, `gpgkey`)
Called by the infrastructure for setting some data.
- def `datakeys` (self)
Returns a list of keys for all stored metadata properties.
- def `setdata` (self, k, v)
Sets a metadata property.
- def `getdata` (self, k, `deflt`)
Gets a metadata property.
- def `initialize` (self)

- Initializes this [Filestructure](#).*
- def [path](#) (self)
 - The path to this structure in the filesystem.*
- def [dl](#) (self, subpath="", to=None, progresscallback=None)
 - Copies the complete filestructure or a subdirectory to a new destination.*
- def [with_modified_rootname](#) (self, newname)
 - Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.*
- def [mv](#) (self, newname)
 - Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.*
- def [setautoopen](#) (self, [path](#), interterminal=False)
 - Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.*

Public Attributes

- [title](#)
- [branch](#)
- [runtimerepourl](#)
- [filename](#)
- [fullname](#)
- [repobaseurl](#)
- [repourl](#)
- [gpgkey](#)

1.194.1 Detailed Description

Description for creation of a Flatpak application reference (.flatpakref) file.

1.194.2 Constructor & Destructor Documentation

1.194.2.1 `__init__()`

```
def anise.features.packages.flatpak.FlatpakRef.__init__ (
    self,
    filename = None,
    title = None,
    branch = "master",
    runtimerepourl = None )
```

1.194.3 Member Function Documentation

1.194.3.1 datakeys()

```
def anise.framework.files.Filestructure.datakeys (
    self ) [inherited]
```

Returns a list of keys for all stored metadata properties.

1.194.3.2 dl()

```
def anise.framework.files.Filestructure.dl (
    self,
    subpath = "",
    to = None,
    progresscallback = None ) [inherited]
```

Copies the complete filestructure or a subdirectory to a new destination.

Parameters

<i>subpath</i>	If given, only this subdirectory will be copied.
<i>to</i>	If given, it specifies the destination path; if not given, a new directory in the current working directory will be used.
<i>progresscallback</i>	Callback function for retrieving progress information.

1.194.3.3 generate()

```
def anise.features.packages.flatpak.FlatpakRef.generate (
    self )
```

1.194.3.4 getdata()

```
def anise.framework.files.Filestructure.getdata (
    self,
    k,
    deflt ) [inherited]
```

Gets a metadata property.

Parameters

<i>k</i>	The key name.
<i>deflt</i>	The default value (returned if the key doesn't exist).

1.194.3.5 initialize()

```
def anise.framework.files.Filestructure.initialize (
    self ) [inherited]
```

Initializes this Filestructure.

When this is called, the dynamic content (if any) is created and written to disc. It is automatically called for typical usage and is not required to be called manually.

1.194.3.6 mv()

```
def anise.framework.files.Filestructure.mv (
    self,
    newname ) [inherited]
```

Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.

Parameters

<i>newname</i>	The new name (absolute or relative).
----------------	--------------------------------------

1.194.3.7 path()

```
def anise.framework.files.Filestructure.path (
    self ) [inherited]
```

The path to this structure in the filesystem.

1.194.3.8 set()

```
def anise.features.packages.flatpak.Metafile.set (
    self,
    fullname,
    repobaseurl,
    repourl,
    gpgkey ) [inherited]
```

Called by the infrastructure for setting some data.

This is part of a particular piece of internal infrastructure and is typically not required to be used.

1.194.3.9 setautoopen()

```
def anise.framework.files.Filestructure.setautoopen (
    self,
    path,
    interterminal = False ) [inherited]
```

Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Parameters

<i>path</i>	The relative path to the file which should automatically be opened.
<i>interterminal</i>	If this hint is for terminal (non-graphical) usage.

1.194.3.10 setdata()

```
def anise.framework.files.Filestructure.setdata (
    self,
    k,
    v ) [inherited]
```

Sets a metadata property.

Parameters

<i>k</i>	The key name.
<i>v</i>	The value.

1.194.3.11 with_modified_rootname()

```
def anise.framework.files.Filestructure.with_modified_rootname (
    self,
    newname ) [inherited]
```

Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.

Parameters

<i>newname</i>	The name which the new root directory should get.
----------------	---

1.194.4 Member Data Documentation

1.194.4.1 branch

`anise.features.packages.flatpak.FlatpakRef.branch`

1.194.4.2 filename

`anise.features.packages.flatpak.Metafile.filename` [inherited]

1.194.4.3 fullname

`anise.features.packages.flatpak.Metafile.fullname` [inherited]

1.194.4.4 gpgkey

`anise.features.packages.flatpak.Metafile.gpgkey` [inherited]

1.194.4.5 repobaseurl

`anise.features.packages.flatpak.Metafile.repobaseurl` [inherited]

1.194.4.6 repourl

`anise.features.packages.flatpak.Metafile.repourl` [inherited]

1.194.4.7 runtimerepourl

`anise.features.packages.flatpak.FlatpakRef.runtimerepourl`

1.194.4.8 title

```
anise.features.packages.flatpak.FlatpakRef.title
```

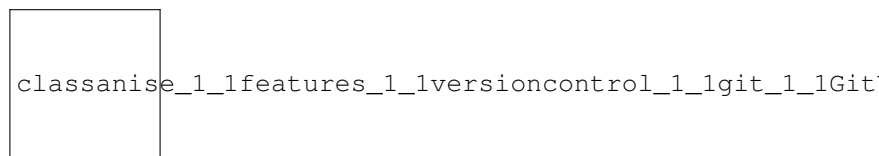
The documentation for this class was generated from the following file:

- [anise/features/packages/flatpak.py](#)

1.195 anise.features.versioncontrol.git.GitVersionControlSystem Class Reference

Git version control system adapter.

Inheritance diagram for anise.features.versioncontrol.git.GitVersionControlSystem:



Public Member Functions

- `def __init__ (self)`
- `def getfullversion (self)`
- `def createrawpackage (self)`
- `def syncversioncontrolsystem (self, commitmessage="", forceskipped=True, forceunchanged=False)`
- `def findvcs (self)`
- `def bindvcs (self, l)`
- `def getheadrevision (self)`
- `def fetchrawcommitmessage (self, revision)`
- `def getcommitmessagefromrawcommitmessage (self, s)`
- `def branchfromhere (self, branchname=None)`
- `def getpreviousrevision (self, revision)`
Gets the predecessor revision for a given revision.
- `def getcommitmessage (self, revision)`
Returns the raw commit message (as formatted by the vcs tool) for a revision.
- `def getchangelog (self, asstring=True)`
Computes the project change log from information in specially formatted vcs commit messages.
- `def addchange (self, changetext=None)`
Adds a change to the changelog.
- `def addlabel (self, label=None)`
Adds a version label for the current state.
- `def storechange (self, changetext)`
Stores a new entry to the changelog.
- `def storelabel (self, label)`
Stores a new version label.

Public Attributes

- [label](#)

1.195.1 Detailed Description

Git version control system adapter.

1.195.2 Constructor & Destructor Documentation

1.195.2.1 `__init__()`

```
def anise.features.versioncontrol.git.GitVersionControlSystem.__init__ (
    self )
```

1.195.3 Member Function Documentation

1.195.3.1 `addchange()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.addchange (
    self,
    changetext = None ) [inherited]
```

Adds a change to the changelog.

Parameters

<i>changetext</i>	The changelog entry text.
-------------------	---------------------------

1.195.3.2 `addlabel()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.addlabel (
    self,
    label = None ) [inherited]
```

Adds a version label for the current state.

Parameters

<i>label</i>	The label string.
--------------	-------------------

1.195.3.3 bindvcs()

```
def anise.features.versioncontrol.git.GitVersionControlSystem.bindvcs (
    self,
    l )
```

1.195.3.4 branchfromhere()

```
def anise.features.versioncontrol.git.GitVersionControlSystem.branchfromhere (
    self,
    branchname = None )
```

1.195.3.5 createrawpackage()

```
def anise.features.versioncontrol.git.GitVersionControlSystem.createrawpackage (
    self )
```

1.195.3.6 fetchrawcommitmessage()

```
def anise.features.versioncontrol.git.GitVersionControlSystem.fetchrawcommitmessage (
    self,
    revision )
```

1.195.3.7 findvcs()

```
def anise.features.versioncontrol.git.GitVersionControlSystem.findvcs (
    self )
```

1.195.3.8 getchangelog()

```
def anise.features.versioncontrol.internals.VersionControlSystem.getchangelog (
    self,
    asstring = True ) [inherited]
```

Computes the project change log from information in specially formatted vcs commit messages.

Parameters

<i>asString</i>	If to output the changelog as string (instead of an internal data structure).
-----------------	---

1.195.3.9 getcommitmessage()

```
def anise.features.versioncontrol.internals.VersionControlSystem.getcommitmessage (
    self,
    revision ) [inherited]
```

Returns the raw commit message (as formatted by the vcs tool) for a revision.

Parameters

<i>revision</i>	The revision.
-----------------	---------------

1.195.3.10 getcommitmessagefromrawcommitmessage()

```
def anise.features.versioncontrol.git.GitVersionControlSystem.getcommitmessagefromrawcommitmessage
(
    self,
    s )
```

1.195.3.11 getfullversion()

```
def anise.features.versioncontrol.git.GitVersionControlSystem.getfullversion (
    self )
```

1.195.3.12 gettheadrevision()

```
def anise.features.versioncontrol.git.GitVersionControlSystem.gettheadrevision (
    self )
```

1.195.3.13 getpreviousrevision()

```
def anise.features.versioncontrol.internals.VersionControlSystem.getpreviousrevision (
    self,
    revision ) [inherited]
```

Gets the predecessor revision for a given revision.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>revision</i>	A revision.
-----------------	-------------

Returns

: The predecessing revision.

1.195.3.14 storechange()

```
def anise.features.versioncontrol.internals.VersionControlSystem.storechange (
    self,
    changetext ) [inherited]
```

Stores a new entry to the changelog.

Do not use this directly, but addchange instead.

Parameters

<i>changetext</i>	The changelog entry text.
-------------------	---------------------------

1.195.3.15 storelabel()

```
def anise.features.versioncontrol.internals.VersionControlSystem.storelabel (
    self,
    label ) [inherited]
```

Stores a new version label.

Do not use this directly, but addlabel instead.

Parameters

<i>label</i>	The label string.
--------------	-------------------

1.195.3.16 syncversioncontrolsystem()

```
def anise.features.versioncontrol.git.GitVersionControlSystem.syncversioncontrolsystem (
    self,
    commitmessage = "",
    forceskipped = True,
    forceunchanged = False )
```

1.195.4 Member Data Documentation

1.195.4.1 label

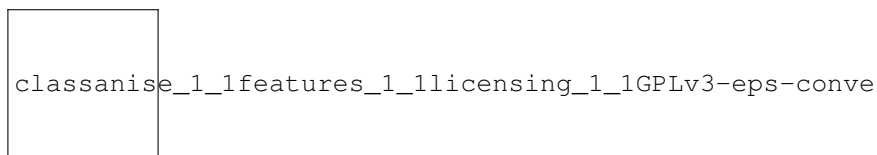
```
anise.features.versioncontrol.git.GitVersionControlSystem.label
```

The documentation for this class was generated from the following file:

- [anise/features/versioncontrol/git.py](#)

1.196 anise.features.licensing.GPLv3 Class Reference

Inheritance diagram for `anise.features.licensing.GPLv3`:



Public Member Functions

- `def __init__ (self)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.196.1 Constructor & Destructor Documentation

1.196.1.1 `__init__()`

```
def anise.features.licensing.GPLv3.__init__ (
    self )
```

1.196.2 Member Function Documentation

1.196.2.1 getfilename()

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.196.2.2 getlicensetext()

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.196.3 Member Data Documentation

1.196.3.1 filename

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.196.3.2 name

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.196.3.3 namedebian

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.196.3.4 namepythonsetuputils

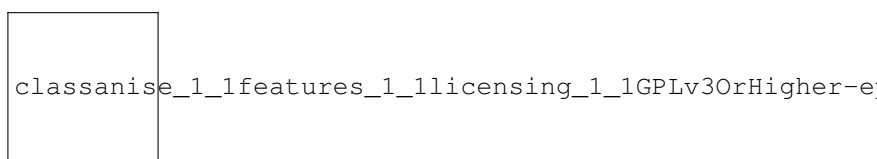
`anise.features.licensing.BaseLicense.namepythonsetuputils` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.197 anise.features.licensing.GPLv3OrHigher Class Reference

Inheritance diagram for `anise.features.licensing.GPLv3OrHigher`:



Public Member Functions

- `def __init__ (self)`
- `def getfilename (self)`
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.197.1 Constructor & Destructor Documentation

1.197.1.1 `__init__()`

```
def anise.features.licensing.GPLv3OrHigher.__init__ (
    self )
```

1.197.2 Member Function Documentation

1.197.2.1 getfilename()

```
def anise.features.licensing.GPLv3OrHigher.getfilename (
    self )
```

1.197.2.2 getlicensetext()

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.197.3 Member Data Documentation

1.197.3.1 filename

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.197.3.2 name

```
anise.features.licensing.GPLv3OrHigher.name
```

1.197.3.3 namedebian

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.197.3.4 namepythonsetuputils

```
anise.features.licensing.BaseLicense.namepythonsetuputils [inherited]
```

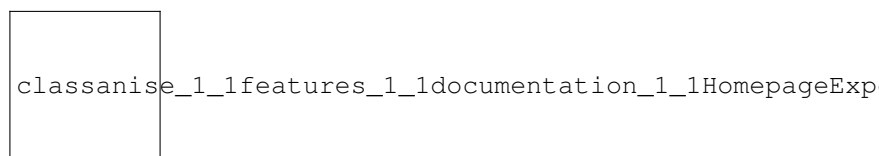
The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.198 anise.features.documentation.HomepageExport Class Reference

Describes an export of an anise documentation directly into the homepage.

Inheritance diagram for anise.features.documentation.HomepageExport:



Public Member Functions

- `def __init__ (self, key, name=None, heading=None, includedeveloperdoc=False, extractall=True, extractprivate=True, hideundocumented=False, imagepaths=None, additionaltext="", shownavbar=False, excludepatterns=None, predefineddocmacros=None, head3=None, head4=None, source=None, sources=None, linktarget=None, index=100000)`
- `def generate (self, targetpath)`
Generates the content.

Public Attributes

- [linktarget](#)
- [index](#)
- [key](#)
- [format](#)
- [name](#)
- [heading](#)
- [includedeveloperdoc](#)
- [extractall](#)
- [extractprivate](#)
- [hideundocumented](#)
- [imagepaths](#)
- [additionaltext](#)
- [shownavbar](#)
- [excludepatterns](#)
- [predefineddocmacros](#)
- [head3](#)
- [head4](#)
- [source](#)
- [sources](#)

1.198.1 Detailed Description

Describes an export of an anise documentation directly into the homepage.

1.198.2 Constructor & Destructor Documentation

1.198.2.1 `__init__()`

```
def anise.features.documentation.HomepageExport.__init__ (
    self,
    key,
    name = None,
    heading = None,
    includedeveloperdoc = False,
    extractall = True,
    extractprivate = True,
    hideundocumented = False,
    imagepaths = None,
    additionaltext = "",
    shownavbar = False,
    excludepatterns = None,
    predefineddocmacros = None,
    head3 = None,
    head4 = None,
    source = None,
    sources = None,
    linktarget = None,
    index = 100000 )
```

Parameters

<i>key</i>	One of the keys stored in pool. Defines which documentation to export.
<i>name</i>	The export definition name. Finding definitions by name is possible with <code>exports.getbyname</code> .
<i>heading</i>	The title.
<i>includedeveloperdoc</i>	If the output shall include the developer documentation as well (does not work for all output formats).
<i>extractall</i>	If a developer documentation should include all elements (not just the documented ones).
<i>extractprivate</i>	If a developer documentation should include also private members.
<i>hideundocumented</i>	If the develo documentation should entirely leave out undocumented members.
<i>imagepaths</i>	Directories from where images are used in the documentation.
<i>additionaltext</i>	Additional text to append to the selected documentation.
<i>shownavbar</i>	If to show navigation bar. Does not work in all output formats.
<i>excludepatterns</i>	File patterns to exclude from reading.
<i>predefineddocmacros</i>	List of additional macros to define for reading.
<i>head3</i>	3rd heading.
<i>head4</i>	4th heading.
<i>source</i>	The documentation source root.
<i>sources</i>	Optional list of source paths (default: ["/"]).
<i>linktarget</i>	Link target.
<i>index</i>	A position number. Used for ordering.

1.198.3 Member Function Documentation

1.198.3.1 generate()

```
def anise.features.documentation.internals.Export.generate (
    self,
    targetpath ) [inherited]
```

Generates the content.

Override this method in custom subclasses or leave the default implementation.

1.198.4 Member Data Documentation

1.198.4.1 additionaltext

```
anise.features.documentation.internals.Export.additionaltext [inherited]
```

1.198.4.2 excludepatterns

```
anise.features.documentation.internals.Export.excludepatterns [inherited]
```

1.198.4.3 extractall

```
anise.features.documentation.internals.Export.extractall [inherited]
```

1.198.4.4 extractprivate

```
anise.features.documentation.internals.Export.extractprivate [inherited]
```

1.198.4.5 format

```
anise.features.documentation.internals.Export.format [inherited]
```

1.198.4.6 head3

`anise.features.documentation.internals.Export.head3` [inherited]

1.198.4.7 head4

`anise.features.documentation.internals.Export.head4` [inherited]

1.198.4.8 heading

`anise.features.documentation.internals.Export.heading` [inherited]

1.198.4.9 hideundocumented

`anise.features.documentation.internals.Export.hideundocumented` [inherited]

1.198.4.10 imagepaths

`anise.features.documentation.internals.Export.imagepaths` [inherited]

1.198.4.11 includedeveloperdoc

`anise.features.documentation.internals.Export.includedeveloperdoc` [inherited]

1.198.4.12 index

`anise.features.documentation.HomepageExport.index`

1.198.4.13 key

`anise.features.documentation.internals.Export.key` [inherited]

1.198.4.14 linktarget

`anise.features.documentation.HomepageExport.linktarget`

1.198.4.15 name

`anise.features.documentation.internals.Export.name` [inherited]

1.198.4.16 predefineddocmacros

`anise.features.documentation.internals.Export.predefineddocmacros` [inherited]

1.198.4.17 shownavbar

`anise.features.documentation.internals.Export.shownavbar` [inherited]

1.198.4.18 source

`anise.features.documentation.internals.Export.source` [inherited]

1.198.4.19 sources

`anise.features.documentation.internals.Export.sources` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/documentation.py](#)

1.199 anise.framework.features.Hook Class Reference

A hook.

Public Member Functions

- `def __init__` (self, `hookkind`=`HookKind.FUNCTION`, `hide`=`False`, `interesting`=`5`, `doc`=`None`, `wizard_↵`
`_functionparams`=`None`, `wizard_functionadditionallog`=`None`, `wizard_functionbody`=`None`, `wizard_↵`
`classbase`=`None`, `wizard_classbody`=`None`, `wizard_loadfeatures`=`None`)
- `def gethandlers` (self)
Returns a list of all registered handlers for this hook.

Public Attributes

- `hookkind`
- `hide`
- `interesting`
- `wizard_functionparams`
- `wizard_functionadditionallog`
- `wizard_functionbody`
- `wizard_classbase`
- `wizard_classbody`
- `wizard_loadfeatures`

Private Attributes

- `__doc__`

1.199.1 Detailed Description

A hook.

See documentation for details.

1.199.2 Constructor & Destructor Documentation

1.199.2.1 __init__()

```
def anise.framework.features.Hook.__init__ (
    self,
    hookkind = HookKind.FUNCTION,
    hide = False,
    interesting = 5,
    doc = None,
    wizard_functionparams = None,
    wizard_functionadditionallog = None,
    wizard_functionbody = None,
    wizard_classbase = None,
    wizard_classbody = None,
    wizard_loadfeatures = None )
```

Parameters

<i>hookkind</i>	A hook kind (one of HookKind).
<i>hide</i>	If this hook is to be hidden in hook choosers.
<i>interesting</i>	A number which controls how interesting this hook is in a chooser.
<i>doc</i>	Hook documentation text.
<i>wizard_functionparams</i>	Info for hook handler wizards. List of function parameters.
<i>wizard_functionadditionallog</i>	Info for hook handler wizards. List of additional variable to log out in a function.
<i>wizard_functionbody</i>	Info for hook handler wizards. Additional function body part.
<i>wizard_classbase</i>	Info for hook handler wizards. Base class.
<i>wizard_classbody</i>	Info for hook handler wizards. Additional class body part.
<i>wizard_loadfeatures</i>	Info for hook handler wizards. Additional features to load.

1.199.3 Member Function Documentation**1.199.3.1 gethandlers()**

```
def anise.framework.features.Hook.gethandlers (
    self )
```

Returns a list of all registered handlers for this hook.

1.199.4 Member Data Documentation**1.199.4.1 __doc__**

```
anise.framework.features.Hook.__doc__ [private]
```

1.199.4.2 hide

```
anise.framework.features.Hook.hide
```

1.199.4.3 hookkind

```
anise.framework.features.Hook.hookkind
```

1.199.4.4 interesting

`anise.framework.features.Hook.interesting`

1.199.4.5 wizard_classbase

`anise.framework.features.Hook.wizard_classbase`

1.199.4.6 wizard_classbody

`anise.framework.features.Hook.wizard_classbody`

1.199.4.7 wizard_functionadditionallog

`anise.framework.features.Hook.wizard_functionadditionallog`

1.199.4.8 wizard_functionbody

`anise.framework.features.Hook.wizard_functionbody`

1.199.4.9 wizard_functionparams

`anise.framework.features.Hook.wizard_functionparams`

1.199.4.10 wizard_loadfeatures

`anise.framework.features.Hook.wizard_loadfeatures`

The documentation for this class was generated from the following file:

- [anise/framework/features.py](#)

1.200 anise.framework.features.HookHandler Class Reference

A hook handler bound to some hook.

Public Member Functions

- `def __init__` (self, [hookobj](#), [function](#), [provides](#), [requires](#), [prepares](#))

Public Attributes

- [hookobj](#)
- [function](#)
- [provides](#)
- [requires](#)
- [prepares](#)

1.200.1 Detailed Description

A hook handler bound to some hook.

See documentation for details.

1.200.2 Constructor & Destructor Documentation

1.200.2.1 __init__()

```
def anise.framework.features.HookHandler.__init__ (
    self,
    hookobj,
    function,
    provides,
    requires,
    prepares )
```

Parameters

<i>hookobj</i>	The hook.
<i>function</i>	The handler function.
<i>provides</i>	A list of symbol names this hook participates to provide (dependency provider).
<i>requires</i>	A list of symbol names this hook requires to be executed before.
<i>prepares</i>	A list of symbol names this hook must prepend.

1.200.3 Member Data Documentation

1.200.3.1 function

`anise.framework.features.HookHandler.function`

1.200.3.2 hookobj

`anise.framework.features.HookHandler.hookobj`

1.200.3.3 prepares

`anise.framework.features.HookHandler.prepares`

1.200.3.4 provides

`anise.framework.features.HookHandler.provides`

1.200.3.5 requires

`anise.framework.features.HookHandler.requires`

The documentation for this class was generated from the following file:

- `anise/framework/features.py`

1.201 anise.framework.features.HookKind Class Reference

Enumeration of hook kinds.

Static Public Attributes

- `FUNCTION`
- `CLASS`

1.201.1 Detailed Description

Enumeration of hook kinds.

1.201.2 Member Data Documentation

1.201.2.1 CLASS

```
anise.framework.features.HookKind.CLASS [static]
```

1.201.2.2 FUNCTION

```
anise.framework.features.HookKind.FUNCTION [static]
```

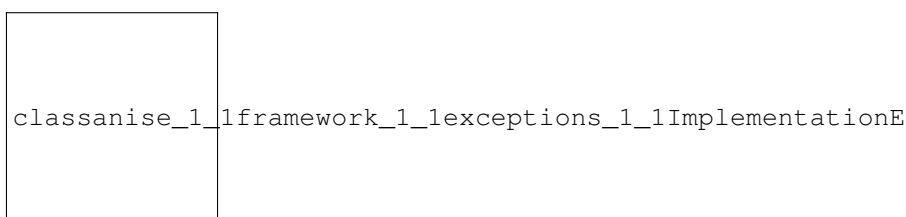
The documentation for this class was generated from the following file:

- [anise/framework/features.py](#)

1.202 anise.framework.exceptions.ImplementationError Class Reference

A software error in an implementation occurred.

Inheritance diagram for anise.framework.exceptions.ImplementationError:



Public Member Functions

- `def __call__ (self, args)`

1.202.1 Detailed Description

A software error in an implementation occurred.

1.202.2 Member Function Documentation

1.202.2.1 `__call__()`

```
def anise.framework.exceptions.AniseError.__call__ (
    self,
    args ) [inherited]
```

The documentation for this class was generated from the following file:

- [anise/framework/exceptions.py](#)

1.203 anise.framework.projects.IntermediateStructure Class Reference

A data structure which holds an in-memory representation of the project description.

Classes

- class [Let](#)
A [Let](#) entry in the project description structure.
- class [Node](#)
A node in the intermediate structure.
- class [ParseProjectDescriptionError](#)
Parsing error like bad input xml.
- class [ParseProjectDescriptionInternalError](#)
Parsing error which seems like a bug.
- class [Value](#)
A [Value](#) entry in the project description structure.

Public Member Functions

- def `__init__` (self, [entries](#)=None)
- def [entries](#) (self)
Returns the list of direct subitems ([IntermediateStructure.Node](#)).
- def [addentry](#) (self, node)
Adds a new entry to the list of direct subentries.
- def [removeentryat](#) (self, i)
Removes an entry from the list of direct subentries.
- def [removeentry](#) (self, node)
Removes an entry from the list of direct subentries.
- def [setentries](#) (self, [entries](#))
Sets the list of direct subentries.
- def `__str__` (self)
- def [tofile](#) (self, filename)
Saves the current state of this intermediate structure to a file.

- def [save](#) (self)
Saves the current state of this intermediate structure back to the disk.
- def [toxml](#) (self)
Generates an xml representation for this intermediate structure.
- def [find](#) (self, key, lets=False, vals=False)
Find all `lets` and/or `vals` with a certain key.
- def [injecttoprojectobject](#) (self, project, g, l)
Injects the (always static) information from this project description into a living universe object (typically for executing a task on top of it).

Static Public Member Functions

- def [fromxml](#) (xmlstring)
Generates an [anise.framework.projects.IntermediateStructure](#) for an xml document.
- def [fromfile](#) (filename)
Generates an [anise.framework.projects.IntermediateStructure](#) for an xml file.
- def [resolveobjectref](#) (oref)
Returns the resolved object for an objectref string.

Public Attributes

- [sourceformat](#)

Static Private Member Functions

- def [_injecttoprojectobject_let](#) (entry, g, l)
Executes the let action from an [IntermediateStructure.Let](#).
- def [_injecttoprojectobject_resolve](#) (s, g, l)
Returns the resolved object for an objectref string.
- def [_injecttoprojectobject_value](#) (entry, g, l)
Returns a plain Python value from an [IntermediateStructure.Value](#).
- def [_fromxml_val](#) (xentry)
Creates an [IntermediateStructure.Value](#) from an xml node.
- def [_fromxml_val_isstring](#) (xentry)
Checks if a particular xml node represents a string.
- def [_fromxml_val_isdict](#) (xentry, withoutkeys=False)
Checks if a particular xml node represents a value dict.
- def [_fromxml_val_islist](#) (xentry)
Checks if a particular xml node represents a value list.
- def [_fromxml_let](#) (xentry)
Creates an [IntermediateStructure.Let](#) from an xml node.
- def [_toxml_val](#) (entry, xcurr)
Converts an [IntermediateStructure.Value](#) to an xml node.
- def [_toxml_let](#) (entry)
Converts an [IntermediateStructure.Let](#) to an xml node.

Private Attributes

- [_entries](#)

1.203.1 Detailed Description

A data structure which 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 [IntermediateStructure](#).

1.203.2 Constructor & Destructor Documentation

1.203.2.1 `__init__()`

```
def anise.framework.projects.IntermediateStructure.__init__ (
    self,
    entries = None )
```

Parameters

<code>entries</code>	The new list of direct subitems (IntermediateStructure.Node).
----------------------	---

1.203.3 Member Function Documentation

1.203.3.1 `__str__()`

```
def anise.framework.projects.IntermediateStructure.__str__ (
    self )
```

1.203.3.2 `_fromxml_let()`

```
def anise.framework.projects.IntermediateStructure._fromxml_let (
    xentry ) [static], [private]
```

Creates an [IntermediateStructure.Let](#) from an xml node.

1.203.3.3 `_fromxml_val()`

```
def anise.framework.projects.IntermediateStructure._fromxml_val (
    xentry ) [static], [private]
```

Creates an [IntermediateStructure.Value](#) from an xml node.

1.203.3.4 `_fromxml_val_isdict()`

```
def anise.framework.projects.IntermediateStructure._fromxml_val_isdict (
    xentry,
    withoutkeys = False ) [static], [private]
```

Checks if a particular xml node represents a value dict.

1.203.3.5 `_fromxml_val_islist()`

```
def anise.framework.projects.IntermediateStructure._fromxml_val_islist (
    xentry ) [static], [private]
```

Checks if a particular xml node represents a value list.

1.203.3.6 `_fromxml_val_isstring()`

```
def anise.framework.projects.IntermediateStructure._fromxml_val_isstring (
    xentry ) [static], [private]
```

Checks if a particular xml node represents a string.

1.203.3.7 `_injecttoprojectobject_let()`

```
def anise.framework.projects.IntermediateStructure._injecttoprojectobject_let (
    entry,
    g,
    l ) [static], [private]
```

Executes the let action from an [IntermediateStructure.Let](#).

1.203.3.8 `_injecttoprojectobject_resolve()`

```
def anise.framework.projects.IntermediateStructure._injecttoprojectobject_resolve (
    s,
    g,
    l ) [static], [private]
```

Returns the resolved object for an objectref string.

1.203.3.9 `_injecttoprojectobject_value()`

```
def anise.framework.projects.IntermediateStructure._injecttoprojectobject_value (
    entry,
    g,
    l ) [static], [private]
```

Returns a plain Python value from an [IntermediateStructure.Value](#).

1.203.3.10 `_toxml_let()`

```
def anise.framework.projects.IntermediateStructure._toxml_let (
    entry ) [static], [private]
```

Converts an [IntermediateStructure.Let](#) to an xml node.

1.203.3.11 `_toxml_val()`

```
def anise.framework.projects.IntermediateStructure._toxml_val (
    entry,
    xcurr ) [static], [private]
```

Converts an [IntermediateStructure.Value](#) to an xml node.

1.203.3.12 `addentry()`

```
def anise.framework.projects.IntermediateStructure.addentry (
    self,
    node )
```

Adds a new entry to the list of direct subentries.

Parameters

<i>node</i>	The new IntermediateStructure.Node .
-------------	--

1.203.3.13 `entries()`

```
def anise.framework.projects.IntermediateStructure.entries (
    self )
```

Returns the list of direct subitems ([IntermediateStructure.Node](#)).

1.203.3.14 find()

```
def anise.framework.projects.IntermediateStructure.find (
    self,
    key,
    lets = False,
    vals = False )
```

Find all `lets` and/or `vals` with a certain key.

1.203.3.15 fromfile()

```
def anise.framework.projects.IntermediateStructure.fromfile (
    filename ) [static]
```

Generates an [anise.framework.projects.IntermediateStructure](#) for an xml file.

Parameters

<i>filename</i>	Path to the project description file.
-----------------	---------------------------------------

1.203.3.16 fromxml()

```
def anise.framework.projects.IntermediateStructure.fromxml (
    xmlstring ) [static]
```

Generates an [anise.framework.projects.IntermediateStructure](#) for an xml document.

1.203.3.17 injecttoprojectobject()

```
def anise.framework.projects.IntermediateStructure.injecttoprojectobject (
    self,
    project,
    g,
    l )
```

Injects the (always static) information from this project description into a living universe object (typically for executing a task on top of it).

Parameters

<i>project</i>	The universe object.
<i>g</i>	The globals dict.
<i>l</i>	The locals dict.

1.203.3.18 removeentry()

```
def anise.framework.projects.IntermediateStructure.removeentry (
    self,
    node )
```

Removes an entry from the list of direct subentries.

Parameters

<i>node</i>	The IntermediateStructure.Node to remove.
-------------	---

1.203.3.19 removeentryat()

```
def anise.framework.projects.IntermediateStructure.removeentryat (
    self,
    i )
```

Removes an entry from the list of direct subentries.

Parameters

<i>i</i>	The entry position to remove.
----------	-------------------------------

1.203.3.20 resolveobjectref()

```
def anise.framework.projects.IntermediateStructure.resolveobjectref (
    oref ) [static]
```

Returns the resolved object for an objectref string.

Parameters

<i>oref</i>	The objectref string (e.g. 'homepage.makehomepage' or 'some.foo.Class().meth()+1').
-------------	---

1.203.3.21 save()

```
def anise.framework.projects.IntermediateStructure.save (
    self )
```

Saves the current state of this intermediate structure back to the disk.

This works only for instances which were loaded from disk as well.

1.203.3.22 setentries()

```
def anise.framework.projects.IntermediateStructure.setentries (
    self,
    entries )
```

Sets the list of direct subentries.

Parameters

<i>entries</i>	The new list of IntermediateStructure.Node .
----------------	--

1.203.3.23 tofile()

```
def anise.framework.projects.IntermediateStructure.tofile (
    self,
    filename )
```

Saves the current state of this intermediate structure to a file.

Parameters

<i>filename</i>	The destination path.
-----------------	-----------------------

1.203.3.24 toxml()

```
def anise.framework.projects.IntermediateStructure.toxml (
    self )
```

Generates an xml representation for this intermediate structure.

1.203.4 Member Data Documentation

1.203.4.1 `_entries`

`anise.framework.projects.IntermediateStructure._entries` [private]

1.203.4.2 `sourceformat`

`anise.framework.projects.IntermediateStructure.sourceformat`

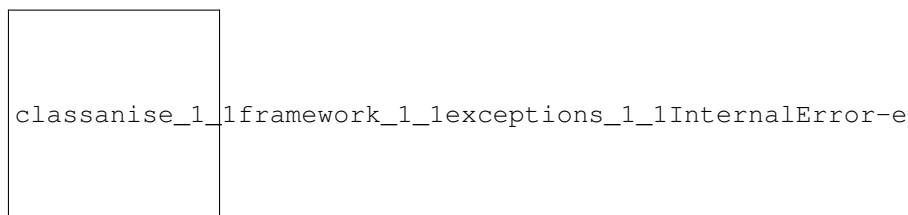
The documentation for this class was generated from the following file:

- [anise/framework/projects.py](#)

1.204 `anise.framework.exceptions.InternalError` Class Reference

An internal error in the anise code occurred.

Inheritance diagram for `anise.framework.exceptions.InternalError`:



Public Member Functions

- `def __call__ (self, args)`

1.204.1 Detailed Description

An internal error in the anise code occurred.

1.204.2 Member Function Documentation

1.204.2.1 `__call__()`

```
def anise.framework.exceptions.AniseError.__call__ (
    self,
    args ) [inherited]
```

The documentation for this class was generated from the following file:

- [anise/framework/exceptions.py](#)

1.205 anise.features.packages.python.internals Class Reference

Static Public Member Functions

- def [buildpythonwheelpackage](#) (name, version, homepage, description, long_description, licensename, maturity, authors, email, keywords, fsroot, applicationlinks, requirements, datafileacceptor=None)
Builds a Python wheel package.

Static Private Member Functions

- def [_isnotpythonfile](#) (fullpath)
Returns True iff the filename does not end with '.py'.
- def [_nsis_get_global](#) ()

1.205.1 Member Function Documentation

1.205.1.1 _isnotpythonfile()

```
def anise.features.packages.python.internals._isnotpythonfile (
    fullpath ) [static], [private]
```

Returns True iff the filename does not end with '.py'.

1.205.1.2 _nsis_get_global()

```
def anise.features.packages.python.internals._nsis_get_global ( ) [static], [private]
```

1.205.1.3 buildpythonwheelpackage()

```
def anise.features.packages.python.internals.buildpythonwheelpackage (
    name,
    version,
    homepage,
    description,
    long_description,
    licensename,
    maturity,
    authors,
    email,
    keywords,
    fsroot,
    applicationlinks,
    requirements,
    datafileacceptor = None ) [static]
```

Builds a Python wheel package.

Parameters

<i>name</i>	The package name.
<i>version</i>	The package version string.
<i>homepage</i>	The homepage url.
<i>description</i>	A short project description.
<i>long_description</i>	A longer project description.
<i>licensename</i>	The name of the project license.
<i>maturity</i>	The project maturity string (may be <code>None</code>).
<i>authors</i>	The project authors as list of string.
<i>email</i>	The contact email address.
<i>keywords</i>	Project keywords for search services.
<i>fsroot</i>	The root directory for the package content (may become damaged during the process).
<i>applicationlinks</i>	Application symlinks to register (list of <code>PythonWheelPackageApplicationLink</code>).
<i>requirements</i>	List of packages which are required for installation on the target system (in wheel notation).
<i>datafileacceptor</i>	An optional function which helps deciding if a file in the source folder is a data file.

The documentation for this class was generated from the following file:

- [anise/features/packages/python.py](#)

1.206 anise.features.dependencies.internals Class Reference

Classes

- class [Pool](#)
Storage of all dependencies.

Static Public Member Functions

- def [getdependencydocument](#) ()
Creates a Markdown documentation which describes the project dependencies in a human-readable form.

Static Private Member Functions

- def [_initproject](#) ()
- def [_add_homepage_section](#) ()

1.206.1 Member Function Documentation

1.206.1.1 `_add_homepage_section()`

```
def anise.features.dependencies.internals._add_homepage_section ( ) [static], [private]
```

1.206.1.2 `_initproject()`

```
def anise.features.dependencies.internals._initproject ( ) [static], [private]
```

1.206.1.3 `getdependencydocument()`

```
def anise.features.dependencies.internals.getdependencydocument ( ) [static]
```

Creates a Markdown documentation which describes the project dependencies in a human-readable form.

It is mostly used for including it into Doxygen documentation source.

Returns

: The Markdown documentation as string.

The documentation for this class was generated from the following file:

- [anise/features/dependencies.py](#)

1.207 `anise.features.projectdescriptioneditor.internals` Class Reference

Classes

- class [EditProjectdescApplication](#)
The raw mode editor.

Static Public Attributes

- [HOOK_GET_FEATURE_ACTIONS](#)
- [provides](#)

1.207.1 Member Data Documentation

1.207.1.1 HOOK_GET_FEATURE_ACTIONS

```
anise.features.projectdescriptioneditor.internals.HOOK_GET_FEATURE_ACTIONS [static]
```

1.207.1.2 provides

```
anise.features.projectdescriptioneditor.internals.provides [static]
```

The documentation for this class was generated from the following file:

- anise/features/[projectdescriptioneditor.py](#)

1.208 anise.features.python.internals Class Reference

Static Public Member Functions

- def [django_quickstart](#) (relpath)
Starts the application in a Django development server for fast testing.

1.208.1 Member Function Documentation

1.208.1.1 django_quickstart()

```
def anise.features.python.internals.django_quickstart (
    relpath ) [static]
```

Starts the application in a Django development server for fast testing.

It updates the database schema before, if required.

Parameters

<i>relpath</i>	Path to the Django application. Relative to the project root directory.
----------------	---

The documentation for this class was generated from the following file:

- anise/features/[python.py](#)

1.209 anise.features.diagnostics.internals Class Reference

Classes

- class [ChildrenInformation](#)
- class [ConsoleWebModule](#)
- class [ReadManualApplication](#)

Static Public Member Functions

- def [featureactions](#) ()
- def [getuniverseattr](#) (membername)
- def [getobjects](#) (obj)
- def [issimplevalue](#) (v)
- def [checkmembervisible](#) (parent, membername, hidemodules=True)
- def [isunexpandable](#) (obj)
- def [getpriority](#) (parent, membername)
- def [geticon](#) (parent, membername)
- def [getuniversedocumentation](#) (web)
- def [getdocumentation](#) (obj)
- def [getfunctionparamspec](#) (func)
- def [detach](#) ()
- def [correctname](#) (s)
- def [featureactionsforfeature](#) (s)
- def [action](#) (i)
- def [info](#) (v="universe")

Public Attributes

- [object](#)
- [originalobject](#)
- [associatedobjects](#)
- [mainpart](#)
- [arguments](#)
- [returns](#)

Static Public Attributes

- [HOOK_GET_FEATURE_ACTIONS](#)
- [provides](#)
- [infostring](#)
- string [listfeaturesstring](#)
- [HOOK_BEFORE_EXECUTION](#)
- [prepares](#)

Private Member Functions

- def [_initproject2](#) ()

Static Private Member Functions

- def [_initproject](#) ()

Static Private Attributes

- `_featureactions` = None

1.209.1 Member Function Documentation

1.209.1.1 `_initproject()`

```
def anise.features.diagnostics.internals._initproject ( ) [static], [private]
```

1.209.1.2 `_initproject2()`

```
def anise.features.diagnostics.internals._initproject2 ( ) [private]
```

1.209.1.3 `action()`

```
def anise.features.diagnostics.internals.action (
    i ) [static]
```

1.209.1.4 `checkmembervisible()`

```
def anise.features.diagnostics.internals.checkmembervisible (
    parent,
    membername,
    hidemodules = True ) [static]
```

1.209.1.5 `correctname()`

```
def anise.features.diagnostics.internals.correctname (
    s ) [static]
```

1.209.1.6 `detach()`

```
def anise.features.diagnostics.internals.detach ( ) [static]
```

1.209.1.7 featureactions()

```
def anise.features.diagnostics.internals.featureactions ( ) [static]
```

1.209.1.8 featureactionsforfeature()

```
def anise.features.diagnostics.internals.featureactionsforfeature (
    s ) [static]
```

1.209.1.9 getdocumentation()

```
def anise.features.diagnostics.internals.getdocumentation (
    obj ) [static]
```

1.209.1.10 getfunctionparamspec()

```
def anise.features.diagnostics.internals.getfunctionparamspec (
    func ) [static]
```

1.209.1.11 geticon()

```
def anise.features.diagnostics.internals.geticon (
    parent,
    membername ) [static]
```

1.209.1.12 getobjects()

```
def anise.features.diagnostics.internals.getobjects (
    obj ) [static]
```

1.209.1.13 getpriority()

```
def anise.features.diagnostics.internals.getpriority (
    parent,
    membername ) [static]
```

1.209.1.14 getuniverseattr()

```
def anise.features.diagnostics.internals.getuniverseattr (
    membername ) [static]
```

1.209.1.15 getuniversedocumentation()

```
def anise.features.diagnostics.internals.getuniversedocumentation (
    web ) [static]
```

1.209.1.16 info()

```
def anise.features.diagnostics.internals.info (
    v = "universe" ) [static]
```

1.209.1.17 issimplevalue()

```
def anise.features.diagnostics.internals.issimplevalue (
    v ) [static]
```

1.209.1.18 isunexpandable()

```
def anise.features.diagnostics.internals.isunexpandable (
    obj ) [static]
```

1.209.2 Member Data Documentation**1.209.2.1 _featureactions**

```
anise.features.diagnostics.internals._featureactions = None [static], [private]
```

1.209.2.2 arguments

```
anise.features.diagnostics.internals.arguments
```

1.209.2.3 associatedobjects

```
anise.features.diagnostics.internals.associatedobjects
```

1.209.2.4 HOOK_BEFORE_EXECUTION

```
anise.features.diagnostics.internals.HOOK_BEFORE_EXECUTION [static]
```

1.209.2.5 HOOK_GET_FEATURE_ACTIONS

```
anise.features.diagnostics.internals.HOOK_GET_FEATURE_ACTIONS [static]
```

1.209.2.6 infostring

```
anise.features.diagnostics.internals.infostring [static]
```

1.209.2.7 listfeaturesstring

```
string anise.features.diagnostics.internals.listfeaturesstring [static]
```

Initial value:

```
= """
Information about 'universe'

Loaded features:
{features}

Members:
{members}

{documentation}

Available actions: {actions}
"""
```

1.209.2.8 mainpart

`anise.features.diagnostics.internals.mainpart`

1.209.2.9 object

`anise.features.diagnostics.internals.object`

1.209.2.10 originalobject

`anise.features.diagnostics.internals.originalobject`

1.209.2.11 prepares

`anise.features.diagnostics.internals.prepares` [static]

1.209.2.12 provides

`anise.features.diagnostics.internals.provides` [static]

1.209.2.13 returns

`anise.features.diagnostics.internals.returns`

The documentation for this class was generated from the following file:

- [anise/features/diagnostics.py](#)

1.210 anise.features.releasing.internals Class Reference

Classes

- class [Releasetasks](#)
Storage of all actions to execute for releasing.

Static Public Member Functions

- def [release](#) ()
Releases the project.

Static Private Member Functions

- def [_initproject1](#) ()
- def [_initproject2](#) ()

1.210.1 Member Function Documentation

1.210.1.1 [_initproject1\(\)](#)

```
def anise.features.releasing.internals._initproject1 ( ) [static], [private]
```

1.210.1.2 [_initproject2\(\)](#)

```
def anise.features.releasing.internals._initproject2 ( ) [static], [private]
```

1.210.1.3 [release\(\)](#)

```
def anise.features.releasing.internals.release ( ) [static]
```

Releases the project.

Mostly this is executing all registered hookhandlers for HOOK_RELEASE.

The documentation for this class was generated from the following file:

- [anise/features/releasing.py](#)

1.211 [anise.features.signing.internals](#) Class Reference

Static Public Member Functions

- def [win32_osslsigncode](#) (binpath, pempath, outpath, projectname, projecthomepage, osslsigncodepath, password)
Signs a win32 binary with osslsigncode.
- def [p12cert2pem](#) (opensslpath, p12path, pemoutpath, password)
Converts a .p12 certificate to a .pem file.

1.211.1 Member Function Documentation

1.211.1.1 p12cert2pem()

```
def anise.features.signing.internals.p12cert2pem (
    opensslpath,
    p12path,
    pemoutpath,
    password ) [static]
```

Converts a .p12 certificate to a .pem file.

Parameters

<i>opensslpath</i>	Path to the OpenSSL binary.
<i>p12path</i>	Path to the p12 certificate file.
<i>pemoutpath</i>	Destination path for the pem file.
<i>password</i>	Certificate password of the p12 file.

1.211.1.2 win32_osslsigncode()

```
def anise.features.signing.internals.win32_osslsigncode (
    binpath,
    pempath,
    outpath,
    projectname,
    projecthomepage,
    osslsigncodepath,
    password ) [static]
```

Signs a win32 binary with osslsigncode.

Parameters

<i>binpath</i>	Path to the binary to sign.
<i>pempath</i>	Path to the pem certificate file to use.
<i>outpath</i>	Destination path for the signed binary.
<i>projectname</i>	The project name.
<i>projecthomepage</i>	The project homepage url.
<i>osslsigncodepath</i>	Path to the osslsigncode tool.
<i>password</i>	Certificate password.

The documentation for this class was generated from the following file:

- [anise/features/signing.py](#)

1.212 anise.features.testing.internals Class Reference

Classes

- class [TestReportWebModule](#)
Anise web module for generating the html test report.
- class [TestRun](#)
This instance keeps track of the test results.
- class [Tests](#)
Storage of all tests.

Static Public Member Functions

- def [storereporthtml](#) (testrun, title, targetpath)
Stores the test report as html to filesystem.
- def [storereporttext](#) (testrun, title, targetpath)
Stores the test report as plaintext to filesystem.

Static Private Member Functions

- def [_initproject](#) ()
- def [_initproject2](#) ()

1.212.1 Member Function Documentation

1.212.1.1 [_initproject\(\)](#)

```
def anise.features.testing.internals._initproject ( ) [static], [private]
```

1.212.1.2 [_initproject2\(\)](#)

```
def anise.features.testing.internals._initproject2 ( ) [static], [private]
```

1.212.1.3 [storereporthtml\(\)](#)

```
def anise.features.testing.internals.storereporthtml (
    testrun,
    title,
    targetpath ) [static]
```

Stores the test report as html to filesystem.

1.212.1.4 storereporttext()

```
def anise.features.testing.internals.storereporttext (
    testrun,
    title,
    targetpath ) [static]
```

Stores the test report as plaintext to filesystem.

The documentation for this class was generated from the following file:

- [anise/features/testing.py](#)

1.213 anise.features.mediagalleries.internals Class Reference

Classes

- class [Pool](#)
Storage of all image galleries.

Static Private Member Functions

- [def _initproject \(\)](#)
- [def _add_homepage_section \(\)](#)

1.213.1 Member Function Documentation

1.213.1.1 _add_homepage_section()

```
def anise.features.mediagalleries.internals._add_homepage_section ( ) [static], [private]
```

1.213.1.2 _initproject()

```
def anise.features.mediagalleries.internals._initproject ( ) [static], [private]
```

The documentation for this class was generated from the following file:

- [anise/features/mediagalleries.py](#)

1.214 anise.features.ui.terminal.internals Class Reference

Classes

- class [UserFeedback](#)
Uses the standard input and standard output channel of the terminal for user interaction.

Public Member Functions

- def [uimodeselected](#) ()

Static Public Attributes

- [HOOK_UIMODE_SELECTED](#)
- [provides](#)

1.214.1 Member Function Documentation

1.214.1.1 uimodeselected()

```
def anise.features.ui.terminal.internals.uimodeselected ( )
```

1.214.2 Member Data Documentation

1.214.2.1 HOOK_UIMODE_SELECTED

```
anise.features.ui.terminal.internals.HOOK_UIMODE_SELECTED [static]
```

1.214.2.2 provides

```
anise.features.ui.terminal.internals.provides [static]
```

The documentation for this class was generated from the following file:

- [anise/features/ui/terminal.py](#)

1.215 anise.features.ui.web.internals Class Reference

Classes

- class [UserFeedback](#)
Uses available web browser sessions for user interaction.

Static Public Member Functions

- def [uimodeselected](#) ()

1.215.1 Member Function Documentation

1.215.1.1 uimodeselected()

```
def anise.features.ui.web.internals.uimodeselected ( ) [static]
```

The documentation for this class was generated from the following file:

- anise/features/ui/[web.py](#)

1.216 anise.features.milieus.internals Class Reference

Static Public Member Functions

- def [get](#) (x=None, defaultmilieu=None)
Returns a [Milieu](#) (or [MilieuComposite](#)).
- def [registermilieu](#) (name, createcallable)
Registers a milieu by a name.

Static Private Member Functions

- def [_initproject](#) ()

1.216.1 Member Function Documentation

1.216.1.1 _initproject()

```
def anise.features.milieus.internals._initproject ( ) [static], [private]
```

1.216.1.2 `get()`

```
def anise.features.milieus.internals.get (
    x = None,
    defaultmilieu = None ) [static]
```

Returns a [Milieu](#) (or [MilieuComposite](#)).

Typically this milieu is created by the given specification. The specification is a list of tuples. Each tuple has at least a 'class' key pointing to a Milieu(Composite) subclass and optional parameters. The milieux will be created stacked together.

Parameters

<code>x</code>	The milieu specification (or a direct milieu instance).
<code>defaultmilieu</code>	The default milieu to create if no other one is specified.

1.216.1.3 `registermilieu()`

```
def anise.features.milieus.internals.registermilieu (
    name,
    createcallable ) [static]
```

Registers a milieu by a name.

This allows shorter names for large class names, and also to register easier factories.

Parameters

<code>name</code>	The name to register the new milieu with.
<code>createcallable</code>	The class or callable which creates the instances.

The documentation for this class was generated from the following file:

- [anise/features/milieus.py](#)

1.217 anise.features.ui.internals Class Reference

Classes

- class [SetImplicitStopAllowed](#)
Controls if implicit application stop is enabled (e.g.
- class [UserFeedback](#)
This interface provides mechanisms for asking the user for some kinds of interaction.
- class [UserFeedbackProxy](#)
This user feedback implementation forwards requests to another internal [UserFeedback](#), depending on which mode (web, terminal) is selected.

Static Public Member Functions

- def [initproject1](#) ()
- def [initproject2](#) ()

Static Private Member Functions

- def [_default](#) ()
Opens the graphical task chooser and executes the selected task afterwards.
- def [_default_baddir](#) ()
Opens the graphical guide for starts in non-Anise directories.

1.217.1 Member Function Documentation

1.217.1.1 `_default()`

```
def anise.features.ui.internals._default ( ) [static], [private]
```

Opens the graphical task chooser and executes the selected task afterwards.

1.217.1.2 `_default_baddir()`

```
def anise.features.ui.internals._default_baddir ( ) [static], [private]
```

Opens the graphical guide for starts in non-Anise directories.

1.217.1.3 `initproject1()`

```
def anise.features.ui.internals.initproject1 ( ) [static]
```

1.217.1.4 `initproject2()`

```
def anise.features.ui.internals.initproject2 ( ) [static]
```

The documentation for this class was generated from the following file:

- [anise/features/ui.py](#)

1.218 `anise.features.distributables.internals` Class Reference

Classes

- class [Pool](#)
Storage of all dependencies.

Static Private Member Functions

- def [_initproject](#) ()
- def [_add_homepage_section](#) ()

1.218.1 Member Function Documentation

1.218.1.1 `_add_homepage_section()`

```
def anise.features.distributables.internals._add_homepage_section ( ) [static], [private]
```

1.218.1.2 `_initproject()`

```
def anise.features.distributables.internals._initproject ( ) [static], [private]
```

The documentation for this class was generated from the following file:

- [anise/features/distributables.py](#)

1.219 anise.features.versioncontrol.git.internals Class Reference

Classes

- class [_Version](#)
Version number of an Anise Git project.

Static Public Member Functions

- def [callgit](#) (opts, raise_on_errors="errors in git", kwargs)
- def [getgitrevision](#) (dirpath)
Determines the revision number of the git working copy.
- def [getgitrevisionhashfromgitrevision](#) (dirpath, revision)
Determines the git revision hash (this is the value which is also used by Git for addressing revisions) for a git revision number (which is not really a part of Git versioning).
- def [listgitrepo](#) (directory)
Determines the file tree of an git repository.

Static Private Member Functions

- def [_register_vcs](#) ()

1.219.1 Member Function Documentation

1.219.1.1 `_register_vcs()`

```
def anise.features.versioncontrol.git.internals._register_vcs ( ) [static], [private]
```

1.219.1.2 `callgit()`

```
def anise.features.versioncontrol.git.internals.callgit (
    opts,
    raise_on_errors = "errors in git",
    kwargs ) [static]
```

1.219.1.3 `getgitrevision()`

```
def anise.features.versioncontrol.git.internals.getgitrevision (
    dirpath ) [static]
```

Determines the revision number of the git working copy.

Parameters

<i>dirpath</i>	The working copy directory.
----------------	-----------------------------

1.219.1.4 `getgitrevisionhashfromgitrevision()`

```
def anise.features.versioncontrol.git.internals.getgitrevisionhashfromgitrevision (
    dirpath,
    revision ) [static]
```

Determines the git revision hash (this is the value which is also used by Git for addressing revisions) for a git revision number (which is not really a part of Git versioning).

Parameters

<i>dirpath</i>	The working copy directory.
<i>revision</i>	The revision number (an Anise concept which does not really exist in Git).

1.219.1.5 `listgitrepo()`

```
def anise.features.versioncontrol.git.internals.listgitrepo (
    directory ) [static]
```


Determines the file tree of an git repository.

Parameters

<i>directory</i>	The directory you want to list.
------------------	---------------------------------

The documentation for this class was generated from the following file:

- [anise/features/versioncontrol/git.py](#)

1.220 anise.features.packages.android.internals Class Reference

Static Public Member Functions

- def [buildandroidpackage_with_ant](#) (projectroot, name, version, targetname, androidpath)
Builds an Android package with Ant.

1.220.1 Member Function Documentation

1.220.1.1 buildandroidpackage_with_ant()

```
def anise.features.packages.android.internals.buildandroidpackage_with_ant (
    projectroot,
    name,
    version,
    targetname,
    androidpath ) [static]
```

Builds an Android package with Ant.

Parameters

<i>projectroot</i>	The path to the Android package.
<i>name</i>	The project name.
<i>version</i>	The project version.
<i>targetname</i>	The Ant target name which shall be built.
<i>androidpath</i>	Path to the Android sdk's <code>android</code> tool.

The documentation for this class was generated from the following file:

- [anise/features/packages/android.py](#)

1.221 anise.features.versioncontrol.svn.internals Class Reference

Classes

- class [_Version](#)
Version number of an Anise Subversion project.

Static Public Member Functions

- def [getsvnrevision](#) (dirpath)
Determines the revision number of a Subversion working copy.
- def [listsvnrepo](#) (dirpath)
Determines the file tree of an Subversion repository.

Static Private Member Functions

- def [_register_vcs](#) ()

1.221.1 Member Function Documentation

1.221.1.1 [_register_vcs\(\)](#)

```
def anise.features.versioncontrol.svn.internals._register_vcs ( ) [static], [private]
```

1.221.1.2 [getsvnrevision\(\)](#)

```
def anise.features.versioncontrol.svn.internals.getsvnrevision (
    dirpath ) [static]
```

Determines the revision number of a Subversion working copy.

Parameters

<i>dirpath</i>	The directory path.
----------------	---------------------

1.221.1.3 [listsvnrepo\(\)](#)

```
def anise.features.versioncontrol.svn.internals.listsvnrepo (
    dirpath ) [static]
```

Determines the file tree of an Subversion repository.

Parameters

<code>dirpath</code>	The directory path.
----------------------	---------------------

The documentation for this class was generated from the following file:

- [anise/features/versioncontrol/svn.py](#)

1.222 anise.features.base.internals Class Reference

Static Public Member Functions

- def [extractfilesfromretval](#) (r, autoopen)
Can be used after executing a task in the anise engine for making a result file available in the target directory.
- def [getbannerimage](#) ()
Determines the location of the banner image.
- def [getprojectstatusdescription](#) (short=False)
Returns a short plaintext describing the specified project maturity (`universe.base.maturity`).
- def [getregisteredtasks](#) ()
Determines the list of all available tasks.

Static Private Member Functions

- def [_initproject](#) ()
- def [_beautify_longdescription](#) ()

1.222.1 Member Function Documentation

1.222.1.1 [_beautify_longdescription\(\)](#)

```
def anise.features.base.internals._beautify_longdescription ( ) [static], [private]
```

1.222.1.2 [_initproject\(\)](#)

```
def anise.features.base.internals._initproject ( ) [static], [private]
```

1.222.1.3 [extractfilesfromretval\(\)](#)

```
def anise.features.base.internals.extractfilesfromretval (
    r,
    autoopen ) [static]
```

Can be used after executing a task in the anise engine for making a result file available in the target directory.

Before you do so, all files created as result are in a temporary area, which will be flushed when the process terminates.

Parameters

<i>r</i>	The raw result (arbitrary object, list, whatever).
<i>autoopen</i>	If anise shall automatically open a file in the result if there is a hint set this way.

1.222.1.4 getbannerimage()

```
def anise.features.base.internals.getbannerimage ( ) [static]
```

Determines the location of the banner image.

Returns

: File path relative to the project root directory.

1.222.1.5 getprojectstatusdescription()

```
def anise.features.base.internals.getprojectstatusdescription (
    short = False ) [static]
```

Returns a short plaintext describing the specified project maturity (`universe.base.maturity`).

Parameters

<i>short</i>	If The output shall be a very reduced one.
--------------	--

1.222.1.6 getregisteredtasks()

```
def anise.features.base.internals.getregisteredtasks ( ) [static]
```

Determines the list of all available tasks.

The documentation for this class was generated from the following file:

- [anise/features/base.py](#)

1.223 anise.features.packages.debian.internals Class Reference**Static Public Member Functions**

- def [builddebpackage](#) (name, version, image, licensename, authors, email, summary, description, section, architecture, websiteurl, menuentries, executablelinks, requireddependencies, stronglyrecommendeddependencies, suggesteddependencies, services, rawpkgpath, projectfsroot, prerm="", postinst="")

Builds a Debian package (.deb).

1.223.1 Member Function Documentation

1.223.1.1 builddebpackage()

```
def anise.features.packages.debian.internals.builddebpackage (
    name,
    version,
    image,
    licensename,
    authors,
    email,
    summary,
    description,
    section,
    architecture,
    websiteurl,
    menuentries,
    executablelinks,
    requireddependencies,
    stronglyrecommendeddependencies,
    suggesteddependencies,
    services,
    rawpkgpath,
    projectfsroot,
    prerm = "",
    postinst = "" ) [static]
```

Builds a Debian package (.deb).

Parameters

<i>name</i>	The package name.
<i>version</i>	The package version string.
<i>image</i>	A project screenshot
<i>licensename</i>	The name of the project license.
<i>authors</i>	Project authors as list of strings.
<i>email</i>	Project email adress.
<i>summary</i>	Short project description.
<i>description</i>	Longer project description.
<i>section</i>	The Debian section name.
<i>architecture</i>	The Debian architecture name.
<i>websiteurl</i>	The project homepage url.
<i>menuentries</i>	Menu entries to register (list of MenuEntry).
<i>executablelinks</i>	Symlinks to executables to be created (string/string dictionary).
<i>requireddependencies</i>	List of required dependencies.
<i>stronglyrecommendeddependencies</i>	List of dependencies, which are not strongly required but very typical.
<i>suggesteddependencies</i>	List of optional dependencies.
<i>services</i>	List of services to be registered.
<i>rawpkgpath</i>	The raw source directory; can be damaged.
<i>projectfsroot</i>	The project source directory path.

Parameters

<i>prerm</i>	Bash code snippet executed before the package is removed on a target machine.
<i>postinst</i>	Bash code snippet executed after the package is installed on a target machine.

The documentation for this class was generated from the following file:

- [anise/features/packages/debian.py](#)

1.224 anise.features.versioncontrol.internals Class Reference

Classes

- class [PseudoVersionControlSystem](#)
A pseudo version control system implementation which does nothing.
- class [VersionControlSystem](#)
A version control system.

Static Private Member Functions

- [def _prepare_fallbacks \(\)](#)
- [def _prepare_vcstasks \(\)](#)
- [def _add_homepage_section \(\)](#)

1.224.1 Member Function Documentation

1.224.1.1 _add_homepage_section()

```
def anise.features.versioncontrol.internals._add_homepage_section ( ) [static], [private]
```

1.224.1.2 _prepare_fallbacks()

```
def anise.features.versioncontrol.internals._prepare_fallbacks ( ) [static], [private]
```

1.224.1.3 `_prepare_vcstasks()`

```
def anise.features.versioncontrol.internals._prepare_vcstasks ( ) [static], [private]
```

The documentation for this class was generated from the following file:

- [anise/features/versioncontrol.py](#)

1.225 `anise.features.documentation.internals` Class Reference

Classes

- class [AbstractFileExport](#)
Abstract base class for exports, which write documentation content to files.
- class [Export](#)
Defines a documentation export.
- class [Exports](#)
Storage of all documentation export definitions (i.e.
- class [Pool](#)
Storage of all documentation text generators.
- class [ShortSnippets](#)
Storage of text snippets for use in the documentation texts like `@a_foo`.

Static Public Member Functions

- def [writedocumentations](#) (destdir, throw_on_existing, fortype)
Immediately exports the documentation as the export definitions currently specify.
- def [loadfile](#) (name)
- def [scalecolor](#) (c, brightness, saturation=None)
Internally used for scaling a color tuple to a new one with the same hue but a given brightness.
- def [deriveccolors](#) (c, saturation=None)
Internally used for computing some css colors for the given basecolor (with different brightnesses).
- def [generate_doxygen_userdocumentation_ashtml](#) (head1, head2, head3="", head4="", additionaltext="", sourcedir, targetdir, basecolor, bannerimage=None, documentations, showdoxygennavbar=False, anise-datadir, doxygensettings="", excludepatterns=None, predefined=None, aliases=None, sources=None, mergehead1and2totitle=False)
Writes all user documentations as html (powered by Doxygen).
- def [generate_doxygen_userdocumentation_asmanpage](#) (projectname, entityname, section, additionaltext="", sourcedir, targetdir, documentations, targetdocumentation, doxygensettings="", excludepatterns=None, predefined=None, aliases=None, sources=None)
Writes one user documentation page as manpage (powered by Doxygen) and returns the path to the result file.
- def [generate_doxygen_userdocumentation_asplaintext](#) (projectname, head, additionaltext="", sourcedir, targetfile, documentations, targetdocumentation, doxygensettings="", excludepatterns=None, predefined=None, aliases=None, sources=None)
Writes one user documentation page as plain text (powered by Doxygen).
- def [generate_doxygen_userdocumentation_aspdf](#) (head1, head2, additionaltext="", sourcedir, targetfile, documentations, targetdocumentation, doxygensettings="", includedeveloperdoc=False, excludepatterns=None, extractprivate=True, extractall=True, hideundocumented=False, predefined=None, imagepaths=None, aliases=None, sources=None)

Writes one user documentation page as pdf (powered by Doxygen).

- def [generate_doxygen_homepage](#) (head1, head2, head3, head4, sourcedir, targetdir, basecolor, bannerimage, sections, anisedatadir, doxygensettings="", excludepatterns=None, predefined=None, aliases=None, sources=None)

Writes the project homepage (powered by Doxygen).

- def [generate_doxygen_documentation](#) (head1, head2, head3, head4, sourcedir, targetdir, doxygensettings, outputformat, additionalsources, basecolor, bannerimage, headlinks, showdoxygennavbar, css, documentations, anisedatadir=None, htmlheadertranslation=None, extractprivate=True, extractall=True, hideundocumented=False, excludepatterns=None, predefined=None, imagepaths=None, aliases=None, sources=None)

Writes some output with Doxygen.

Static Public Attributes

- [provides](#)
- [requires](#)
- [HOOK_BEFORE_EXECUTION](#)

Private Member Functions

- def [_compose_readmes](#) ()
- def [_add_homepage_section](#) ()

Static Private Member Functions

- def [_initproject](#) ()
- def [_add_release_task_for_export_to_source](#) ()

1.225.1 Member Function Documentation

1.225.1.1 [_add_homepage_section\(\)](#)

```
def anise.features.documentation.internals._add_homepage_section ( ) [private]
```

1.225.1.2 [_add_release_task_for_export_to_source\(\)](#)

```
def anise.features.documentation.internals._add_release_task_for_export_to_source ( ) [static], [private]
```


1.225.1.3 _compose_readmes()

```
def anise.features.documentation.internals._compose_readmes ( ) [private]
```

1.225.1.4 _initproject()

```
def anise.features.documentation.internals._initproject ( ) [static], [private]
```

1.225.1.5 derivecolors()

```
def anise.features.documentation.internals.derivecolors (
    c,
    saturation = None ) [static]
```

Internally used for computing some css colors for the given basecolor (with different brightnesses).

1.225.1.6 generate_doxygen_documentation()

```
def anise.features.documentation.internals.generate_doxygen_documentation (
    head1,
    head2,
    head3,
    head4,
    sourcedir,
    targetdir,
    doxygensettings,
    outputformat,
    additionalsources,
    basecolor,
    bannerimage,
    headlinks,
    showdoxygenavbar,
    css,
    documentations,
    anisedatadir = None,
    htmlheadertranslation = None,
    extractprivate = True,
    extractall = True,
    hideundocumented = False,
    excludepatterns = None,
    predefined = None,
    imagepaths = None,
    aliases = None,
    sources = None ) [static]
```

Writes some output with Doxygen.

Mostly used by some higher-level functions in this module.

Parameters

<i>head1</i>	The first heading.
<i>head2</i>	The second heading.
<i>head3</i>	The third heading.
<i>head4</i>	The fourth heading.
<i>sourcedir</i>	The directory where the sourcefiles are.
<i>targetdir</i>	The directory for the output.
<i>doxygensettings</i>	Additional Doxygen configuration for finetuning.
<i>outputformat</i>	The desired output format (as in doxygen config's GENERATE_xxx keys).
<i>additionalsources</i>	Auxiliary source code passed to Doxygen for injecting additional content.
<i>basecolor</i>	The basecolor for colouring the output.
<i>bannerimage</i>	Absolute path to the banner image (will be copied).
<i>headlinks</i>	Links in the heading panel (html output only).
<i>showdoxygennavbar</i>	If the doxygen navigation bar should be visible.
<i>css</i>	Additional css (html output only).
<i>documentations</i>	An anise documentations tuple for further pages, which do not exist in the sourcecode (e.g. the readme sources) as added by <code>anise.features.documentation.pool.add</code> .
<i>anisedatadir</i>	Absolute path to the anise data dir (needed for finding icons, ...; typically somewhere in the anise installation path).
<i>htmlheadertranslation</i>	A function which can replace stuff in the default html header (html output only).
<i>extractprivate</i>	If private members shall be extracted to the developer documentation.
<i>extractall</i>	If all members shall be extracted to the developer documentation.
<i>hideundocumented</i>	If undocumented member shall be left out in the developer documentation.
<i>excludepatterns</i>	File path patterns to exclude for composing the documentation.
<i>predefined</i>	A list of (C++) macros to define before the source files are parsed.
<i>imagepaths</i>	Directories from where images are used in the documentation.
<i>aliases</i>	Custom command aliases (as string/string-dict).
<i>sources</i>	Optional list of source paths (default: ["/"]).

1.225.1.7 generate_doxygen_homepage()

```
def anise.features.documentation.internals.generate_doxygen_homepage (
    head1,
    head2,
    head3,
    head4,
    sourcedir,
    targetdir,
    basecolor,
    bannerimage,
    sections,
    anisedatadir,
    doxygensettings = "",
    excludepatterns = None,
    predefined = None,
    aliases = None,
    sources = None ) [static]
```

Writes the project homepage (powered by Doxygen).

You find it afterwards in `index.html` (and tons of other files).

Parameters

<i>head1</i>	The first heading (typically the project name).
<i>head2</i>	The second heading (typically a one-liner description).
<i>head3</i>	The third heading (typically version information).
<i>head4</i>	The fourth heading.
<i>sourcedir</i>	The directory where the sourcefiles are.
<i>targetdir</i>	The directory for the html output.
<i>basecolor</i>	The basecolor for colouring the output.
<i>bannerimage</i>	Absolute path to the banner image (will be copied).
<i>sections</i>	A list of (sectiontitle,sectionsource,additionalparams) tuples for the actual content.
<i>anisedatadir</i>	Absolute path to the anise data dir (needed for finding icons, ...; typically somewhere in the anise installation path).
<i>doxygensettings</i>	Additional Doxygen configuration for finetuning.
<i>excludepatterns</i>	File path patterns to exclude for composing the documentation.
<i>predefined</i>	A list of (C++) macros to define before the source files are parsed.
<i>aliases</i>	Custom command aliases (as string/string-dict).
<i>sources</i>	Optional list of source paths (default: ["/"]).

1.225.1.8 generate_doxygen_userdocumentation_ashtml()

```
def anise.features.documentation.internals.generate_doxygen_userdocumentation_ashtml (
    head1,
    head2,
    head3 = "",
    head4 = "",
    additionaltext = "",
    sourcedir,
    targetdir,
    basecolor,
    bannerimage = None,
    documentations,
    showdoxygennavbar = False,
    anisedatadir,
    doxygensettings = "",
    excludepatterns = None,
    predefined = None,
    aliases = None,
    sources = None,
    mergehead1and2totitle = False ) [static]
```

Writes all user documentations as html (powered by Doxygen).

You find it afterwards in the `(documentkey).html` files (and tons of other files). This will contain the developer documentation as well, so references into it will work.

Parameters

<i>head1</i>	The first heading (typically the project name).
<i>head2</i>	The second heading.
<i>head3</i>	The third heading.
<i>head4</i>	The fourth heading.
<i>additionaltext</i>	Auxiliary source code passed to Doxygen for injecting additional content.
<i>sourcedir</i>	The directory where the sourcefiles are.
<i>targetdir</i>	The directory for the html output.
<i>basecolor</i>	The basecolor for colouring the output.
<i>bannerimage</i>	Absolute path to the banner image (will be copied).
<i>documentations</i>	An anise documentations tuple for further pages, which do not exist in the sourcecode (e.g. the readme sources) as added by <code>anise.features.documentation.pool.add</code> .
<i>showdoxygennavbar</i>	If the doxygen navigation bar should be visible.
<i>anisedatadir</i>	Path to the Anise data directory (which contains some data files which might be needed).
<i>doxygensettings</i>	Additional Doxygen configuration for finetuning.
<i>excludepatterns</i>	File path patterns to exclude for composing the documentation.
<i>predefined</i>	A list of (C++) macros to define before the source files are parsed.
<i>aliases</i>	Custom command aliases (as string/string-dict).
<i>sources</i>	Optional list of source paths (default: ["/"]).
<i>mergehead1and2totitle</i>	If head1 and head2 shall be merged together for the page title.

1.225.1.9 `generate_doxygen_userdocumentation_asmanpage()`

```
def anise.features.documentation.internals.generate_doxygen_userdocumentation_asmanpage (
    projectname,
    entityname,
    section,
    additionaltext = "",
    sourcedir,
    targetdir,
    documentations,
    targetdocumentation,
    doxygensettings = "",
    excludepatterns = None,
    predefined = None,
    aliases = None,
    sources = None ) [static]
```

Writes one user documentation page as manpage (powered by Doxygen) and returns the path to the result file.

Parameters

<i>projectname</i>	The project name.
<i>entityname</i>	The NAME for the manpage (typically the name of the command, function or whatever).
<i>section</i>	The section number (see man manpage).
<i>additionaltext</i>	Auxiliary source code passed to Doxygen for injecting additional content.

Parameters

<i>sourcedir</i>	The directory where the sourcefiles are.
<i>targetdir</i>	The directory for the manpage output.
<i>documentations</i>	An anise documentations tuple for further pages, which do not exist in the sourcecode (e.g. the readme sources) as added by anise.features.documentation.pool.add.
<i>targetdocumentation</i>	The key for the documentation which must be written.
<i>doxygensettings</i>	Additional Doxygen configuration for finetuning.
<i>excludepatterns</i>	File path patterns to exclude for composing the documentation.
<i>predefined</i>	A list of (C++) macros to define before the source files are parsed.
<i>aliases</i>	Custom command aliases (as string/string-dict).
<i>sources</i>	Optional list of source paths (default: [" / "]).

1.225.1.10 generate_doxygen_userdocumentation_aspdf()

```
def anise.features.documentation.internals.generate_doxygen_userdocumentation_aspdf (
    head1,
    head2,
    additionaltext = "",
    sourcedir,
    targetfile,
    documentations,
    targetdocumentation,
    doxygensettings = "",
    includedeveloperdoc = False,
    excludepatterns = None,
    extractprivate = True,
    extractall = True,
    hideundocumented = False,
    predefined = None,
    imagepaths = None,
    aliases = None,
    sources = None ) [static]
```

Writes one user documentation page as pdf (powered by Doxygen).

Parameters

<i>head1</i>	The first heading (typically the project name).
<i>head2</i>	The second heading.
<i>additionaltext</i>	Auxiliary source code passed to Doxygen for injecting additional content.
<i>sourcedir</i>	The directory where the sourcefiles are.
<i>targetfile</i>	The target file for the pdf output.
<i>documentations</i>	An anise documentations tuple for further pages, which do not exist in the sourcecode (e.g. the readme sources) as added by anise.features.documentation.pool.add.
<i>targetdocumentation</i>	The key for the documentation which must be written.
<i>doxygensettings</i>	Additional Doxygen configuration for finetuning.
<i>includedeveloperdoc</i>	If the pdf shall include the developer documentation.
<i>excludepatterns</i>	File path patterns to exclude for composing the documentation.

Parameters

<i>extractprivate</i>	If private members shall be extracted to the developer documentation.
<i>extractall</i>	If all members shall be extracted to the developer documentation.
<i>hideundocumented</i>	If undocumented member shall be left out in the developer documentation.
<i>predefined</i>	A list of (C++) macros to define before the source files are parsed.
<i>imagepaths</i>	Directories from where images are used in the documentation.
<i>aliases</i>	Custom command aliases (as string/string-dict).
<i>sources</i>	Optional list of source paths (default: ["/"]).

1.225.1.11 generate_doxygen_userdocumentation_asplaintext()

```
def anise.features.documentation.internals.generate_doxygen_userdocumentation_asplaintext (
    projectname,
    head,
    additionaltext = "",
    sourcedir,
    targetfile,
    documentations,
    targetdocumentation,
    doxygensettings = "",
    excludepatterns = None,
    predefined = None,
    aliases = None,
    sources = None ) [static]
```

Writes one user documentation page as plain text (powered by Doxygen).

Parameters

<i>projectname</i>	The project name.
<i>head</i>	The heading.
<i>additionaltext</i>	Auxiliary source code passed to Doxygen for injecting additional content.
<i>sourcedir</i>	The directory where the sourcefiles are.
<i>targetfile</i>	The target file for the plaintext output.
<i>documentations</i>	An anise documentations tuple for further pages, which do not exist in the sourcecode (e.g. the readme sources) as added by <code>anise.features.documentation.pool.add</code> .
<i>targetdocumentation</i>	The key for the documentation which must be written.
<i>doxygensettings</i>	Additional Doxygen configuration for finetuning.
<i>excludepatterns</i>	File path patterns to exclude for composing the documentation.
<i>predefined</i>	A list of (C++) macros to define before the source files are parsed.
<i>aliases</i>	Custom command aliases (as string/string-dict).
<i>sources</i>	Optional list of source paths (default: ["/"]).

1.225.1.12 loadfile()

```
def anise.features.documentation.internals.loadfile (
    name ) [static]
```

1.225.1.13 scalecolor()

```
def anise.features.documentation.internals.scalecolor (
    c,
    brightness,
    saturation = None ) [static]
```

Internally used for scaling a color tuple to a new one with the same hue but a given brightness.

1.225.1.14 writedocumentations()

```
def anise.features.documentation.internals.writedocumentations (
    destdir,
    throw_on_existing,
    fortype ) [static]
```

Immediately exports the documentation as the export definitions currently specify.

1.225.2 Member Data Documentation

1.225.2.1 HOOK_BEFORE_EXECUTION

```
anise.features.documentation.internals.HOOK_BEFORE_EXECUTION [static]
```

1.225.2.2 provides

```
anise.features.documentation.internals.provides [static]
```

1.225.2.3 requires

```
anise.features.documentation.internals.requires [static]
```

The documentation for this class was generated from the following file:

- [anise/features/documentation.py](#)

1.226 anise.features.packages.appc.internals Class Reference

Static Public Member Functions

- def [call](#) (cmd, raise_on_errors="unable to run acbuild")

Static Private Member Functions

- def [_initproject](#) ()

1.226.1 Member Function Documentation

1.226.1.1 [_initproject\(\)](#)

```
def anise.features.packages.appc.internals._initproject ( ) [static], [private]
```

1.226.1.2 [call\(\)](#)

```
def anise.features.packages.appc.internals.call (
    cmd,
    raise_on_errors = "unable to run acbuild" ) [static]
```

The documentation for this class was generated from the following file:

- anise/features/packages/[appc.py](#)

1.227 anise.features.packages.flatpak.internals Class Reference

Static Private Member Functions

- def [_initproject](#) ()

1.227.1 Member Function Documentation

1.227.1.1 `_initproject()`

```
def anise.features.packages.flatpak.internals._initproject ( ) [static], [private]
```

The documentation for this class was generated from the following file:

- [anise/features/packages/flatpak.py](#)

1.228 anise.features.build.autotools.internals Class Reference

Static Public Attributes

- [HOOK_BEFORE_DEFINITION](#)
- [requires](#)

Private Member Functions

- [def _initproject \(\)](#)

1.228.1 Member Function Documentation

1.228.1.1 `_initproject()`

```
def anise.features.build.autotools.internals._initproject ( ) [private]
```

1.228.2 Member Data Documentation

1.228.2.1 `HOOK_BEFORE_DEFINITION`

```
anise.features.build.autotools.internals.HOOK_BEFORE_DEFINITION [static]
```

1.228.2.2 `requires`

```
anise.features.build.autotools.internals.requires [static]
```

The documentation for this class was generated from the following file:

- [anise/features/build/autotools.py](#)

1.229 anise.features.packages.win32.internals Class Reference

Static Public Member Functions

- def [buildwindowsinstallerpackage](#) (name, version, licensefile, authors, email, summary, architecture, websiteurl, menuentries, rawpkgpath, projectfsroot, projecticon=None, globalscript="", oninitscript="", oninstallscrip="", usemodernui2=False)

Builds a Windows installer executable (.exe).

1.229.1 Member Function Documentation

1.229.1.1 buildwindowsinstallerpackage()

```
def anise.features.packages.win32.internals.buildwindowsinstallerpackage (
    name,
    version,
    licensefile,
    authors,
    email,
    summary,
    architecture,
    websiteurl,
    menuentries,
    rawpkgpath,
    projectfsroot,
    projecticon = None,
    globalscript = "",
    oninitscript = "",
    oninstallscrip = "",
    usemodernui2 = False ) [static]
```

Builds a Windows installer executable (.exe).

Parameters

<i>name</i>	The package name.
<i>version</i>	The package version string.
<i>licensefile</i>	The path to the full text of the project's license.
<i>authors</i>	Project authors as list of strings.
<i>email</i>	Project email adress.
<i>summary</i>	Short project description.
<i>architecture</i>	The Debian architecture name.
<i>websiteurl</i>	The project homepage url.
<i>menuentries</i>	Menu entries to register (list of MenuEntry).
<i>rawpkgpath</i>	The raw source directory; can be damaged.
<i>projectfsroot</i>	The project source directory path.
<i>projecticon</i>	Path to the project icon.
<i>globalscript</i>	NSIS script global part.
<i>oninitscript</i>	NSIS script which runs at initialization of the installer.
<i>oninstallscrip</i>	NSIS script which runs after file copying of the installer.
<i>usemodernui2</i>	If Modern UI 2 shall be used (brings a more fresh look).

The documentation for this class was generated from the following file:

- [anise/features/packages/win32.py](#)

1.230 anise.features.build.make.internals Class Reference

Static Public Attributes

- [HOOK_BEFORE_DEFINITION](#)
- [requires](#)

Private Member Functions

- [def _initproject\(\)](#)

1.230.1 Member Function Documentation

1.230.1.1 _initproject()

```
def anise.features.build.make.internals._initproject ( ) [private]
```

1.230.2 Member Data Documentation

1.230.2.1 HOOK_BEFORE_DEFINITION

```
anise.features.build.make.internals.HOOK_BEFORE_DEFINITION [static]
```

1.230.2.2 requires

```
anise.features.build.make.internals.requires [static]
```

The documentation for this class was generated from the following file:

- [anise/features/build/make.py](#)

1.231 anise.features.packages.internals Class Reference

Static Public Member Functions

- def `enrichrawpackage` (packagedir)
Adds files to a raw packages (as created by makerawpackage), like the license file.
- def `getversion` ()
Determines the version string of the project.
- def `makerawpackage` ()
Builds a raw package in some flavour.

Static Private Member Functions

- def `_initproject` ()

Static Private Attributes

- `_makerawpackage_cache` = None

1.231.1 Member Function Documentation

1.231.1.1 `_initproject()`

```
def anise.features.packages.internals._initproject ( ) [static], [private]
```

1.231.1.2 `enrichrawpackage()`

```
def anise.features.packages.internals.enrichrawpackage (
    packagedir ) [static]
```

Adds files to a raw packages (as created by makerawpackage), like the license file.

Parameters

<code>packagedir</code>	The raw package root directory path.
-------------------------	--------------------------------------

1.231.1.3 `getversion()`

```
def anise.features.packages.internals.getversion ( ) [static]
```

Determines the version string of the project.

1.231.1.4 makerawpackage()

```
def anise.features.packages.internals.makerawpackage ( ) [static]
```

Builds a raw package in some flavour.

This is used by other components (e.g. packaging) for creating distributable binary packages.

1.231.2 Member Data Documentation

1.231.2.1 _makerawpackage_cache

```
anise.features.packages.internals._makerawpackage_cache = None [static], [private]
```

The documentation for this class was generated from the following file:

- [anise/features/packages.py](#)

1.232 anise.features.packagestore.internals Class Reference

Static Public Member Functions

- def [sliceversion](#) (versionstring)
Slices a version string to a tuple.
- def [quotename](#) (s)
Quotes a name for using as filename.
- def [unquotename](#) (s)
Unquotes a name quoted by [internals::quotename\(\)](#) back.

Static Public Attributes

- [HOOK_BEFORE_DEFINITION](#)
- [provides](#)

Private Member Functions

- def [_initproject](#) ()

1.232.1 Member Function Documentation

1.232.1.1 `_initproject()`

```
def anise.features.packagestore.internals._initproject ( ) [private]
```

1.232.1.2 `quotename()`

```
def anise.features.packagestore.internals.quotename (
    s ) [static]
```

Quotes a name for using as filename.

Parameters

<code>s</code>	A string to quote.
----------------	--------------------

Returns

: A quoted string which can be used as a filename.

1.232.1.3 `sliceversion()`

```
def anise.features.packagestore.internals.sliceversion (
    versionstring ) [static]
```

Slices a version string to a tuple.

Parameters

<code>versionstring</code>	Version as string.
----------------------------	--------------------

Returns

: Version tuple.

1.232.1.4 `unquotename()`

```
def anise.features.packagestore.internals.unquotename (
    s ) [static]
```

Unquotes a name quoted by `internals::quotename()` back.

Parameters

s	The quoted string.
---	--------------------

Returns

: The original name.

1.232.2 Member Data Documentation

1.232.2.1 HOOK_BEFORE_DEFINITION

```
anise.features.packagestore.internals.HOOK_BEFORE_DEFINITION [static]
```

1.232.2.2 provides

```
anise.features.packagestore.internals.provides [static]
```

The documentation for this class was generated from the following file:

- [anise/features/packagestore.py](#)

1.233 anise.features.filetransfer.internals Class Reference

Static Public Member Functions

- def [generatenewindex](#) (appidx, indexpattern, projectname)
- def [updateindexfile](#) (appidxpath, indexpattern, projectname)
- def [createprogresscallback](#) (execscope)

1.233.1 Member Function Documentation

1.233.1.1 createprogresscallback()

```
def anise.features.filetransfer.internals.createprogresscallback (  
    execscope ) [static]
```

1.233.1.2 generatenewindex()

```
def anise.features.filetransfer.internals.generatenewindex (
    appidx,
    indexpattern,
    projectname ) [static]
```

1.233.1.3 updateindexfile()

```
def anise.features.filetransfer.internals.updateindexfile (
    appidxpath,
    indexpattern,
    projectname ) [static]
```

The documentation for this class was generated from the following file:

- [anise/features/filetransfer.py](#)

1.234 anise.features.homepage.internals Class Reference

Classes

- class [Sections](#)
Storage of all homepage sections.

Static Private Member Functions

- def [_initproject](#) ()
- def [_basic_homepage_sections](#) ()

1.234.1 Member Function Documentation

1.234.1.1 _basic_homepage_sections()

```
def anise.features.homepage.internals._basic_homepage_sections ( ) [static], [private]
```


1.234.1.2 `_initproject()`

```
def anise.features.homepage.internals._initproject ( ) [static], [private]
```

The documentation for this class was generated from the following file:

- [anise/features/homepage.py](#)

1.235 anise.framework.features.internals Class Reference

Internal stuff.

Static Public Member Functions

- def [injectdefaultstoproject](#) (project)
Adds feature data structures to the universe object.
- def [injecthookstoproject](#) (project)
Adds hooks to the universe object, which are loaded in feature loading.
- def [loadfeatures](#) (featurespath, name="")
Loads features from a path.
- def [resetfeatures](#) ()
Resets the features.

1.235.1 Detailed Description

Internal stuff.

1.235.2 Member Function Documentation

1.235.2.1 `injectdefaultstoproject()`

```
def anise.framework.features.internals.injectdefaultstoproject (
    project ) [static]
```

Adds feature data structures to the universe object.

1.235.2.2 `injecthookstoproject()`

```
def anise.framework.features.internals.injecthookstoproject (
    project ) [static]
```

Adds hooks to the universe object, which are loaded in feature loading.

1.235.2.3 loadfeatures()

```
def anise.framework.features.internals.loadfeatures (
    featurespath,
    name = "" ) [static]
```

Loads features from a path.

1.235.2.4 resetfeatures()

```
def anise.framework.features.internals.resetfeatures ( ) [static]
```

Resets the features.

The documentation for this class was generated from the following file:

- anise/framework/[features.py](#)

1.236 anise.features.datainjections.internals Class Reference

Classes

- class [Pool](#)

Static Private Member Functions

- def [_sync](#) ()
Sync if not marked as omitsyncvcs.
- def [_initproject](#) ()

1.236.1 Member Function Documentation

1.236.1.1 _initproject()

```
def anise.features.datainjections.internals._initproject ( ) [static], [private]
```

1.236.1.2 `_sync()`

```
def anise.features.datainjections.internals._sync ( ) [static], [private]
```

Sync if not marked as `omitsyncvcs`.

The documentation for this class was generated from the following file:

- [anise/features/datainjections.py](#)

1.237 anise.features.licensing.internals Class Reference

Static Public Attributes

- [HOOK_BEFORE_EXECUTION](#)
- [provides](#)

Private Member Functions

- [def _instantiatelicense \(\)](#)

Static Private Member Functions

- [def _writelicense \(destdir\)](#)
- [def _add_homepage_section \(\)](#)

1.237.1 Member Function Documentation

1.237.1.1 `_add_homepage_section()`

```
def anise.features.licensing.internals._add_homepage_section ( ) [static], [private]
```

1.237.1.2 `_instantiatelicense()`

```
def anise.features.licensing.internals._instantiatelicense ( ) [private]
```

1.237.1.3 `_writelicense()`

```
def anise.features.licensing.internals._writelicense (
    destdir ) [static], [private]
```

1.237.2 Member Data Documentation

1.237.2.1 HOOK_BEFORE_EXECUTION

`anise.features.licensing.internals.HOOK_BEFORE_EXECUTION` [static]

1.237.2.2 provides

`anise.features.licensing.internals.provides` [static]

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.238 `anise.features.build.qmake.internals` Class Reference

Static Public Attributes

- [HOOK_BEFORE_DEFINITION](#)
- [requires](#)

Private Member Functions

- `def _initproject ()`

1.238.1 Member Function Documentation

1.238.1.1 `_initproject()`

```
def anise.features.build.qmake.internals._initproject ( ) [private]
```

1.238.2 Member Data Documentation

1.238.2.1 HOOK_BEFORE_DEFINITION

```
anise.features.build.qmake.internals.HOOK_BEFORE_DEFINITION [static]
```

1.238.2.2 requires

```
anise.features.build.qmake.internals.requires [static]
```

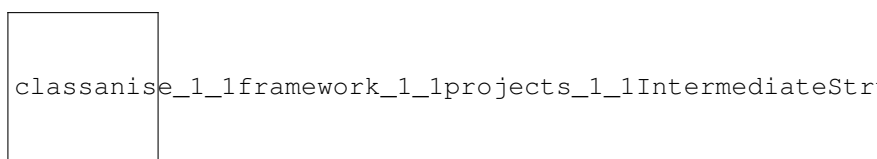
The documentation for this class was generated from the following file:

- [anise/features/build/qmake.py](#)

1.239 anise.framework.projects.IntermediateStructure.Let Class Reference

A [Let](#) entry in the project description structure.

Inheritance diagram for `anise.framework.projects.IntermediateStructure.Let`:



Public Member Functions

- `def __init__(self, k, args)`
- `def __str__(self)`
- `def key(self)`
The key name.
- `def parent(self)`
Returns the parent node (either [IntermediateStructure](#) or [IntermediateStructure.Node](#)).
- `def args(self)`
Returns the list of stored argument keys.
- `def getarg(self, key)`
Returns an argument (as [IntermediateStructure.Value](#)).
- `def setarg(self, key, node)`
Sets an argument.
- `def removearg(self, key)`
Removes an argument.
- `def clearargs(self)`
Clears the list of arguments.
- `def remove(self)`
Removes this node.

Public Attributes

- [islet](#)
- [isvalue](#)

1.239.1 Detailed Description

A [Let](#) entry in the project description structure.

This corresponds to a `let` node in the xml format. It represents the execution of a parameterized part of anise feature program logic.

1.239.2 Constructor & Destructor Documentation

1.239.2.1 `__init__()`

```
def anise.framework.projects.IntermediateStructure.Let.__init__ (
    self,
    k,
    args )
```

Parameters

<i>k</i>	The full qualified method name.
----------	---------------------------------

1.239.3 Member Function Documentation

1.239.3.1 `__str__()`

```
def anise.framework.projects.IntermediateStructure.Let.__str__ (
    self )
```

1.239.3.2 `args()`

```
def anise.framework.projects.IntermediateStructure.Node.args (
    self ) [inherited]
```

Returns the list of stored argument keys.

1.239.3.3 clearargs()

```
def anise.framework.projects.IntermediateStructure.Node.clearargs (
    self ) [inherited]
```

Clears the list of arguments.

1.239.3.4 getarg()

```
def anise.framework.projects.IntermediateStructure.Node.getarg (
    self,
    key ) [inherited]
```

Returns an argument (as [IntermediateStructure.Value](#)).

Parameters

<i>key</i>	The argument key.
------------	-------------------

1.239.3.5 key()

```
def anise.framework.projects.IntermediateStructure.Node.key (
    self ) [inherited]
```

The key name.

1.239.3.6 parent()

```
def anise.framework.projects.IntermediateStructure.Node.parent (
    self ) [inherited]
```

Returns the parent node (either [IntermediateStructure](#) or [IntermediateStructure.Node](#)).

1.239.3.7 remove()

```
def anise.framework.projects.IntermediateStructure.Node.remove (
    self ) [inherited]
```

Removes this node.

1.239.3.8 removearg()

```
def anise.framework.projects.IntermediateStructure.Node.removearg (
    self,
    key ) [inherited]
```

Removes an argument.

Parameters

<i>key</i>	The argument key.
------------	-------------------

1.239.3.9 setarg()

```
def anise.framework.projects.IntermediateStructure.Node.setarg (
    self,
    key,
    node ) [inherited]
```

Sets an argument.

Parameters

<i>key</i>	The argument key.
<i>node</i>	The argument value (as IntermediateStructure.Value).

1.239.4 Member Data Documentation

1.239.4.1 islet

```
anise.framework.projects.IntermediateStructure.Let.islet
```

1.239.4.2 isvalue

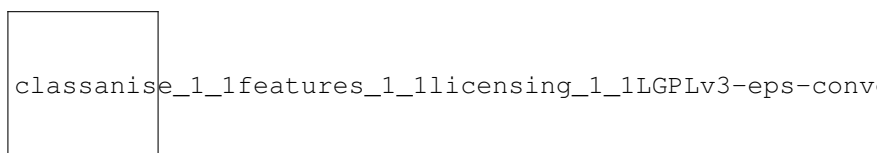
```
anise.framework.projects.IntermediateStructure.Node.isvalue [inherited]
```

The documentation for this class was generated from the following file:

- [anise/framework/projects.py](#)

1.240 anise.features.licensing.LGPLv3 Class Reference

Inheritance diagram for `anise.features.licensing.LGPLv3`:



Public Member Functions

- def `__init__` (self)
- def `getfilename` (self)
Returns the full path of the license text file.
- def `getlicensetext` (self)
Returns the full license text.

Public Attributes

- `name`
- `namedebian`
- `namepythonsetuputils`
- `filename`

1.240.1 Constructor & Destructor Documentation

1.240.1.1 `__init__()`

```
def anise.features.licensing.LGPLv3.__init__ (  
    self )
```

1.240.2 Member Function Documentation

1.240.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (  
    self ) [inherited]
```

Returns the full path of the license text file.

1.240.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (  
    self ) [inherited]
```

Returns the full license text.

1.240.3 Member Data Documentation

1.240.3.1 filename

`anise.features.licensing.BaseLicense.filename` [inherited]

1.240.3.2 name

`anise.features.licensing.BaseLicense.name` [inherited]

1.240.3.3 namedebian

`anise.features.licensing.BaseLicense.namedebian` [inherited]

1.240.3.4 namepythonsetuputils

`anise.features.licensing.BaseLicense.namepythonsetuputils` [inherited]

The documentation for this class was generated from the following file:

- `anise/features/licensing.py`

1.241 `anise.features.projectdescriptioneditor.filetransfer.LocalFiletransfer` Class Reference

Public Member Functions

- `def __init__(self)`
- `def ask(self)`

Public Attributes

- `label`

1.241.1 Constructor & Destructor Documentation

1.241.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.filetransfer.LocalFiletransfer.__init__ (
    self )
```

1.241.2 Member Function Documentation

1.241.2.1 `ask()`

```
def anise.features.projectdescriptioneditor.filetransfer.LocalFiletransfer.ask (
    self )
```

1.241.3 Member Data Documentation

1.241.3.1 `label`

```
anise.features.projectdescriptioneditor.filetransfer.LocalFiletransfer.label
```

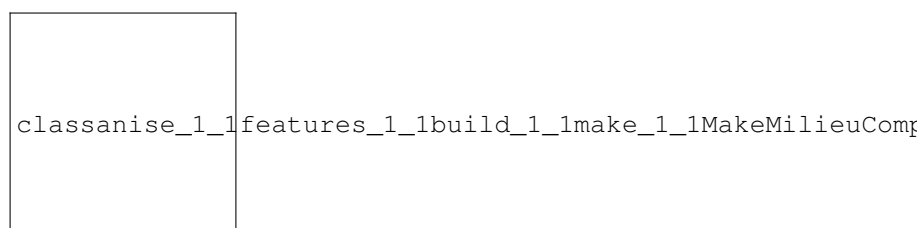
The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/filetransfer.py](#)

1.242 anise.features.build.make.MakeMilieuComposite Class Reference

Build environment composite for make builds.

Inheritance diagram for `anise.features.build.make.MakeMilieuComposite`:



Public Member Functions

- `def __init__` (self, `makepath`="make", `processcount`=None)
- `def call_make` (self, args, kwargs)
- `def make` (self, `targetname`=None, `makefile`=None)
- `def initialize` (self)
Initializes the milieu (composite).
- `def setinner` (self, `inner`)
Sets the inner milieu.
- `def __getattr__` (self, item)

Public Attributes

- `makepath`
- `processcount`
- `makecmd`
- `inner`

1.242.1 Detailed Description

Build environment composite for make builds.

See [milieus.MilieuComposite](#) for general infos.

1.242.2 Constructor & Destructor Documentation

1.242.2.1 __init__()

```
def anise.features.build.make.MakeMilieuComposite.__init__ (
    self,
    makepath = "make",
    processcount = None )
```

Parameters

<code>makepath</code>	The path to the <code>make</code> tool.
<code>processcount</code>	The maximum number of concurrent processes to execute.

1.242.3 Member Function Documentation

1.242.3.1 `__getattr__()`

```
def anise.features.milieus.MilieuComposite.__getattr__ (
    self,
    item ) [inherited]
```

1.242.3.2 `call_make()`

```
def anise.features.build.make.MakeMilieuComposite.call_make (
    self,
    args,
    kwargs )
```

1.242.3.3 `initialize()`

```
def anise.features.milieus.MilieuComposite.initialize (
    self ) [inherited]
```

Initializes the milieu (composite).

Override this method in custom subclasses or leave the default implementation.

1.242.3.4 `make()`

```
def anise.features.build.make.MakeMilieuComposite.make (
    self,
    targetname = None,
    makefile = None )
```

1.242.3.5 `setinner()`

```
def anise.features.milieus.MilieuComposite.setinner (
    self,
    inner ) [inherited]
```

Sets the inner milieu.

This is part of a particular piece of internal infrastructure and is typically not required to be used.

1.242.4 Member Data Documentation

1.242.4.1 inner

`anise.features.milieus.MilieuComposite.inner` [inherited]

1.242.4.2 makecmd

`anise.features.build.make.MakeMilieuComposite.makecmd`

1.242.4.3 makepath

`anise.features.build.make.MakeMilieuComposite.makepath`

1.242.4.4 processcount

`anise.features.build.make.MakeMilieuComposite.processcount`

The documentation for this class was generated from the following file:

- [anise/features/build/make.py](#)

1.243 anise.features.base.Maturity Class Reference

Enumeration describing possible project states.

Static Public Attributes

- string [Planning](#) = 'Development Status :: 1 - Planning'
- string [PreAlpha](#) = 'Development Status :: 2 - Pre-Alpha'
- string [Alpha](#) = 'Development Status :: 3 - Alpha'
- string [Beta](#) = 'Development Status :: 4 - Beta'
- string [ProductionStable](#) = 'Development Status :: 5 - Production/Stable'
- string [Mature](#) = 'Development Status :: 6 - Mature'
- string [Inactive](#) = 'Development Status :: 7 - Inactive'

1.243.1 Detailed Description

Enumeration describing possible project states.

1.243.2 Member Data Documentation

1.243.2.1 Alpha

```
string anise.features.base.Maturity.Alpha = 'Development Status :: 3 - Alpha' [static]
```

1.243.2.2 Beta

```
string anise.features.base.Maturity.Beta = 'Development Status :: 4 - Beta' [static]
```

1.243.2.3 Inactive

```
string anise.features.base.Maturity.Inactive = 'Development Status :: 7 - Inactive' [static]
```

1.243.2.4 Mature

```
string anise.features.base.Maturity.Mature = 'Development Status :: 6 - Mature' [static]
```

1.243.2.5 Planning

```
string anise.features.base.Maturity.Planning = 'Development Status :: 1 - Planning' [static]
```

1.243.2.6 PreAlpha

```
string anise.features.base.Maturity.PreAlpha = 'Development Status :: 2 - Pre-Alpha' [static]
```

1.243.2.7 ProductionStable

```
string anise.features.base.Maturity.ProductionStable = 'Development Status :: 5 - Production/Stable'
[static]
```

The documentation for this class was generated from the following file:

- [anise/features/base.py](#)

1.244 anise.features.mediagalleries.Media Class Reference

One image in a gallery.

Public Member Functions

- `def __init__(self, path, description, mediatype)`

Public Attributes

- [path](#)
- [description](#)
- [mediatype](#)

1.244.1 Detailed Description

One image in a gallery.

1.244.2 Constructor & Destructor Documentation

1.244.2.1 __init__()

```
def anise.features.mediagalleries.Media.__init__(
    self,
    path,
    description,
    mediatype )
```

Parameters

<i>path</i>	The image file path.
<i>description</i>	Description text.

1.244.3 Member Data Documentation

1.244.3.1 description

`anise.features.mediagalleries.Media.description`

1.244.3.2 mediatype

`anise.features.mediagalleries.Media.mediatype`

1.244.3.3 path

`anise.features.mediagalleries.Media.path`

The documentation for this class was generated from the following file:

- [anise/features/mediagalleries.py](#)

1.245 anise.features.mediagalleries.MediaGallery Class Reference

An image gallery.

Public Member Functions

- `def __init__ (self, path=None, paths=None)`
- `def getmedialist (self)`
Gets the list of media in the gallery.
- `def cloneto (self, destpath)`
Clones the media gallery directory to a different place.

Public Attributes

- [paths](#)

Private Member Functions

- `def _matchfile (self, name)`

1.245.1 Detailed Description

An image gallery.

1.245.2 Constructor & Destructor Documentation

1.245.2.1 `__init__()`

```
def anise.features.mediagalleries.MediaGallery.__init__ (
    self,
    path = None,
    paths = None )
```

Parameters

<i>path</i>	The directory path where the image gallery is stored.
<i>paths</i>	The directory paths where the image gallery is stored.

1.245.3 Member Function Documentation

1.245.3.1 `_matchfile()`

```
def anise.features.mediagalleries.MediaGallery._matchfile (
    self,
    name ) [private]
```

1.245.3.2 `cloneto()`

```
def anise.features.mediagalleries.MediaGallery.cloneto (
    self,
    destpath )
```

Clones the media gallery directory to a different place.

Parameters

<i>destpath</i>	The destination where to copy the image gallery data. Absolute path.
-----------------	--

Returns

: The new [MediaGallery](#) instance pointing to the cloned directory.

1.245.3.3 getmedialist()

```
def anise.features.mediagalleries.MediaGallery.getmedialist (
    self )
```

Gets the list of media in the gallery.

Returns

: List of [Media](#) instances.

1.245.4 Member Data Documentation**1.245.4.1 paths**

```
anise.features.mediagalleries.MediaGallery.paths
```

The documentation for this class was generated from the following file:

- [anise/features/mediagalleries.py](#)

1.246 anise.features.packages.win32.MenuEntry Class Reference

Specification for one start menu entry added by a Windows package installer.

Public Member Functions

- `def __init__ (self, title, command, icon=None, params="", workdir="$INSTDIR")`

Public Attributes

- [title](#)
- [command](#)
- [icon](#)
- [params](#)
- [workdir](#)

1.246.1 Detailed Description

Specification for one start menu entry added by a Windows package installer.

1.246.2 Constructor & Destructor Documentation

1.246.2.1 `__init__()`

```
def anise.features.packages.win32.MenuEntry.__init__ (
    self,
    title,
    command,
    icon = None,
    params = "",
    workdir = "$INSTDIR" )
```

Parameters

<i>title</i>	Menu entry title.
<i>command</i>	Menu entry command.
<i>icon</i>	The path to a menu entry icon.
<i>params</i>	Additional command parameters (as string).
<i>workdir</i>	The working directory for this menu entry.

1.246.3 Member Data Documentation

1.246.3.1 `command`

```
anise.features.packages.win32.MenuEntry.command
```

1.246.3.2 `icon`

```
anise.features.packages.win32.MenuEntry.icon
```

1.246.3.3 `params`

```
anise.features.packages.win32.MenuEntry.params
```

1.246.3.4 title

```
anise.features.packages.win32.MenuEntry.title
```

1.246.3.5 workdir

```
anise.features.packages.win32.MenuEntry.workdir
```

The documentation for this class was generated from the following file:

- [anise/features/packages/win32.py](#)

1.247 anise.features.packages.debian.MenuEntry Class Reference

Specification for one menu entry added by a Debian installation package.

Public Member Functions

- `def __init__(self, name, title, category, command, gui, icon)`

Public Attributes

- [name](#)
- [title](#)
- [category](#)
- [command](#)
- [gui](#)
- [icon](#)

1.247.1 Detailed Description

Specification for one menu entry added by a Debian installation package.

1.247.2 Constructor & Destructor Documentation

1.247.2.1 __init__()

```
def anise.features.packages.debian.MenuEntry.__init__(  
    self,  
    name,  
    title,  
    category,  
    command,  
    gui,  
    icon )
```

Parameters

<i>name</i>	Internal menu entry name.
<i>title</i>	Menu entry Title.
<i>category</i>	Menu entry category.
<i>command</i>	Menu entry command.
<i>gui</i>	If the command opens a gui (instead of a terminal application).
<i>icon</i>	The path to a menu entry icon.

1.247.3 Member Data Documentation

1.247.3.1 category

```
anise.features.packages.debian.MenuEntry.category
```

1.247.3.2 command

```
anise.features.packages.debian.MenuEntry.command
```

1.247.3.3 gui

```
anise.features.packages.debian.MenuEntry.gui
```

1.247.3.4 icon

```
anise.features.packages.debian.MenuEntry.icon
```

1.247.3.5 name

```
anise.features.packages.debian.MenuEntry.name
```

1.247.3.6 title

`anise.features.packages.debian.MenuEntry.title`

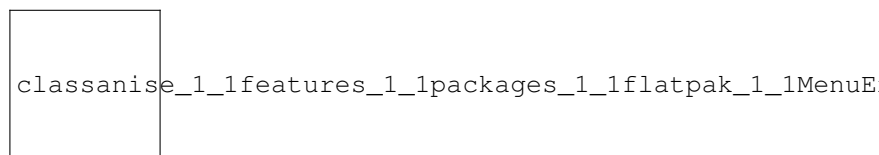
The documentation for this class was generated from the following file:

- [anise/features/packages/debian.py](#)

1.248 anise.features.packages.flatpak.MenuEntry Class Reference

Specification for one menu entry added by a Flatpak.

Inheritance diagram for `anise.features.packages.flatpak.MenuEntry`:



Public Member Functions

- `def __init__ (self, command="", a, b)`
See more about parameters in [packages.debian.MenuEntry](#).

1.248.1 Detailed Description

Specification for one menu entry added by a Flatpak.

1.248.2 Constructor & Destructor Documentation

1.248.2.1 __init__()

```
def anise.features.packages.flatpak.MenuEntry.__init__ (
    self,
    command = "",
    a,
    b )
```

See more about parameters in [packages.debian.MenuEntry](#).

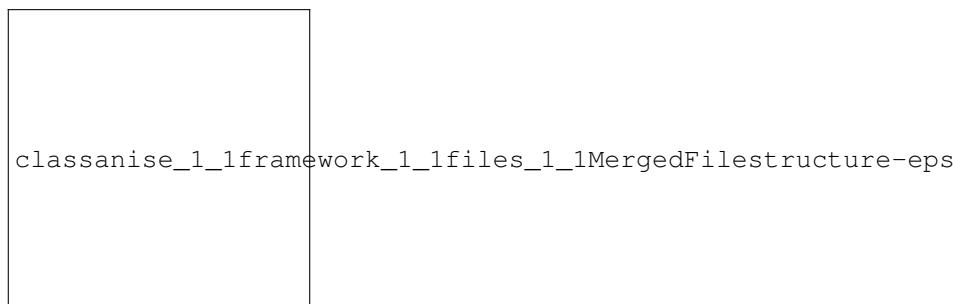
The documentation for this class was generated from the following file:

- [anise/features/packages/flatpak.py](#)

1.249 anise.framework.files.MergedFilestructure Class Reference

Merges many [Filestructure](#) instances together in one tree.

Inheritance diagram for `anise.framework.files.MergedFilestructure`:



Public Member Functions

- def `__init__` (self, `fromlist`=None, `frommap`=None)
- def `getdests` (self)
- def `getsource` (self, `desttuple`)
- def `getmerged` (self)
- def `datakeys` (self)
 - Returns a list of keys for all stored metadata properties.*
- def `setdata` (self, `k`, `v`)
 - Sets a metadata property.*
- def `getdata` (self, `k`, `deflt`)
 - Gets a metadata property.*
- def `initialize` (self)
 - Initializes this [Filestructure](#).*
- def `path` (self)
 - The path to this structure in the filesystem.*
- def `dl` (self, `subpath`="", `to`=None, `progresscallback`=None)
 - Copies the complete filestructure or a subdirectory to a new destination.*
- def `with_modified_rootname` (self, `newname`)
 - Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.*
- def `mv` (self, `newname`)
 - Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.*
- def `setautoopen` (self, `path`, `intermodal`=False)
 - Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.*

Public Attributes

- `fromlist`
- `frommap`
- `task`
- `params`
- `inner`

1.249.1 Detailed Description

Merges many [Filestructure](#) instances together in one tree.

1.249.2 Constructor & Destructor Documentation

1.249.2.1 `__init__()`

```
def anise.framework.files.MergedFilestructure.__init__ (
    self,
    fromlist = None,
    frommap = None )
```

Parameters

<i>fromlist</i>	A list of Filestructure . Become root directory entries with their original file names.
<i>frommap</i>	A dict of string/Filestructure. Values become part of the result tree with the dict key specifying a subpath.

1.249.3 Member Function Documentation

1.249.3.1 `datakeys()`

```
def anise.framework.files.Filestructure.datakeys (
    self ) [inherited]
```

Returns a list of keys for all stored metadata properties.

1.249.3.2 `dl()`

```
def anise.framework.files.Filestructure.dl (
    self,
    subpath = "",
    to = None,
    progresscallback = None ) [inherited]
```

Copies the complete filestructure or a subdirectory to a new destination.

Parameters

<i>subpath</i>	If given, only this subdirectory will be copied.
<i>to</i>	If given, it specifies the destination path; if not given, a new directory in the current working directory will be used.
<i>progresscallback</i>	Callback function for retrieving progress information.

1.249.3.3 `getdata()`

```
def anise.framework.files.Filestructure.getdata (
    self,
    k,
    deflt ) [inherited]
```

Gets a metadata property.

Parameters

<i>k</i>	The key name.
<i>deflt</i>	The default value (returned if the key doesn't exist).

1.249.3.4 `getdests()`

```
def anise.framework.files.MergedFilestructure.getdests (
    self )
```

1.249.3.5 `getmerged()`

```
def anise.framework.files.AbstractRebuildDirectoriesFilestructure.getmerged (
    self ) [inherited]
```

1.249.3.6 `getsource()`

```
def anise.framework.files.MergedFilestructure.getsource (
    self,
    desttuple )
```

1.249.3.7 initialize()

```
def anise.framework.files.Filestructure.initialize (
    self ) [inherited]
```

Initializes this [Filestructure](#).

When this is called, the dynamic content (if any) is created and written to disc. It is automatically called for typical usage and is not required to be called manually.

1.249.3.8 mv()

```
def anise.framework.files.Filestructure.mv (
    self,
    newname ) [inherited]
```

Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.

Parameters

<i>newname</i>	The new name (absolute or relative).
----------------	--------------------------------------

1.249.3.9 path()

```
def anise.framework.files.Filestructure.path (
    self ) [inherited]
```

The path to this structure in the filesystem.

1.249.3.10 setautoopen()

```
def anise.framework.files.Filestructure.setautoopen (
    self,
    path,
    interminal = False ) [inherited]
```

Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Parameters

<i>path</i>	The relative path to the file which should automatically be opened.
<i>interminal</i>	If this hint is for terminal (non-graphical) usage.

1.249.3.11 setdata()

```
def anise.framework.files.Filestructure.setdata (
    self,
    k,
    v ) [inherited]
```

Sets a metadata property.

Parameters

<i>k</i>	The key name.
<i>v</i>	The value.

1.249.3.12 with_modified_rootname()

```
def anise.framework.files.Filestructure.with_modified_rootname (
    self,
    newname ) [inherited]
```

Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.

Parameters

<i>newname</i>	The name which the new root directory should get.
----------------	---

1.249.4 Member Data Documentation

1.249.4.1 fromlist

```
anise.framework.files.MergedFilestructure.fromlist
```

1.249.4.2 frommap

```
anise.framework.files.MergedFilestructure.frommap
```

1.249.4.3 inner

```
anise.framework.files.TaskExecution.inner [inherited]
```

1.249.4.4 params

```
anise.framework.files.TaskExecution.params [inherited]
```

1.249.4.5 task

```
anise.framework.files.TaskExecution.task [inherited]
```

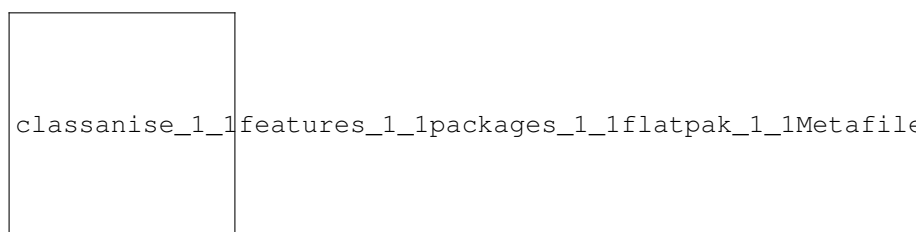
The documentation for this class was generated from the following file:

- anise/framework/[files.py](#)

1.250 anise.features.packages.flatpak.Metafile Class Reference

An abstract description for a Flatpak meta file for easier access (e.g.

Inheritance diagram for anise.features.packages.flatpak.Metafile:



Public Member Functions

- def [__init__](#) (self, [filename](#))
- def [set](#) (self, [fullname](#), [repobaseurl](#), [repourl](#), [gpgkey](#))
Called by the infrastructure for setting some data.
- def [generate](#) (self)
Generates and returns the metafile content as string.
- def [datakeys](#) (self)
Returns a list of keys for all stored metadata properties.
- def [setdata](#) (self, k, v)
Sets a metadata property.
- def [getdata](#) (self, k, deflt)
Gets a metadata property.

- def `initialize` (self)
Initializes this [Filestructure](#).
- def `path` (self)
The path to this structure in the filesystem.
- def `dl` (self, subpath="", to=None, progresscallback=None)
Copies the complete filestructure or a subdirectory to a new destination.
- def `with_modified_rootname` (self, newname)
Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.
- def `mv` (self, newname)
Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.
- def `setautoopen` (self, `path`, interterminal=False)
Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Public Attributes

- `filename`
- `fullname`
- `repobaseurl`
- `repourl`
- `gpgkey`

Private Member Functions

- def `_initialize` (self)

Private Attributes

- `_path`

1.250.1 Detailed Description

An abstract description for a Flatpak meta file for easier access (e.g.

a `.flatpakref`).

Use one of the non-abstract subclasses.

1.250.2 Constructor & Destructor Documentation

1.250.2.1 `__init__()`

```
def anise.features.packages.flatpak.Metafile.__init__ (
    self,
    filename )
```

1.250.3 Member Function Documentation

1.250.3.1 `_initialize()`

```
def anise.features.packages.flatpak.Metafile._initialize (
    self ) [private]
```

1.250.3.2 `datakeys()`

```
def anise.framework.files.Filestructure.datakeys (
    self ) [inherited]
```

Returns a list of keys for all stored metadata properties.

1.250.3.3 `dl()`

```
def anise.framework.files.Filestructure.dl (
    self,
    subpath = "",
    to = None,
    progresscallback = None ) [inherited]
```

Copies the complete filestructure or a subdirectory to a new destination.

Parameters

<i>subpath</i>	If given, only this subdirectory will be copied.
<i>to</i>	If given, it specifies the destination path; if not given, a new directory in the current working directory will be used.
<i>progresscallback</i>	Callback function for retrieving progress information.

1.250.3.4 `generate()`

```
def anise.features.packages.flatpak.Metafile.generate (
    self )
```

Generates and returns the metafile content as string.

Override this method in custom subclasses.

1.250.3.5 getdata()

```
def anise.framework.files.Filestructure.getdata (
    self,
    k,
    deflt ) [inherited]
```

Gets a metadata property.

Parameters

<i>k</i>	The key name.
<i>deflt</i>	The default value (returned if the key doesn't exist).

1.250.3.6 initialize()

```
def anise.framework.files.Filestructure.initialize (
    self ) [inherited]
```

Initializes this Filestructure.

When this is called, the dynamic content (if any) is created and written to disc. It is automatically called for typical usage and is not required to be called manually.

1.250.3.7 mv()

```
def anise.framework.files.Filestructure.mv (
    self,
    newname ) [inherited]
```

Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.

Parameters

<i>newname</i>	The new name (absolute or relative).
----------------	--------------------------------------

1.250.3.8 path()

```
def anise.framework.files.Filestructure.path (
    self ) [inherited]
```

The path to this structure in the filesystem.

1.250.3.9 set()

```
def anise.features.packages.flatpak.Metafile.set (
    self,
    fullname,
    repobaseurl,
    repourl,
    gpgkey )
```

Called by the infrastructure for setting some data.

This is part of a particular piece of internal infrastructure and is typically not required to be used.

1.250.3.10 setautoopen()

```
def anise.framework.files.Filestructure.setautoopen (
    self,
    path,
    interterminal = False ) [inherited]
```

Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Parameters

<i>path</i>	The relative path to the file which should automatically be opened.
<i>interterminal</i>	If this hint is for terminal (non-graphical) usage.

1.250.3.11 setdata()

```
def anise.framework.files.Filestructure.setdata (
    self,
    k,
    v ) [inherited]
```

Sets a metadata property.

Parameters

<i>k</i>	The key name.
<i>v</i>	The value.

1.250.3.12 with_modified_rootname()

```
def anise.framework.files.Filestructure.with_modified_rootname (
    self,
    newname ) [inherited]
```

Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.

Parameters

<i>newname</i>	The name which the new root directory should get.
----------------	---

1.250.4 Member Data Documentation

1.250.4.1 _path

```
anise.features.packages.flatpak.Metafile._path [private]
```

1.250.4.2 filename

```
anise.features.packages.flatpak.Metafile.filename
```

1.250.4.3 fullname

```
anise.features.packages.flatpak.Metafile.fullname
```

1.250.4.4 gpgkey

```
anise.features.packages.flatpak.Metafile.gpgkey
```

1.250.4.5 repobaseurl

```
anise.features.packages.flatpak.Metafile.repobaseurl
```

1.250.4.6 repourl

```
anise.features.packages.flatpak.Metafile.repourl
```

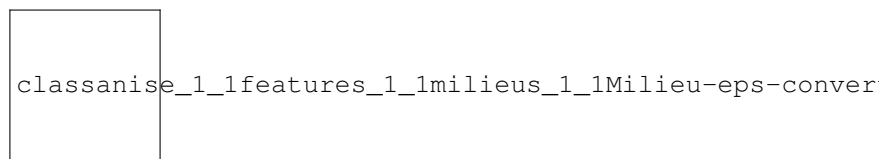
The documentation for this class was generated from the following file:

- [anise/features/packages/flatpak.py](#)

1.251 anise.features.milieus.Milieu Class Reference

Milieux are a functionality which allows a build task to work in different environments.

Inheritance diagram for anise.features.milieus.Milieu:



Public Member Functions

- `def __init__ (self)`
- `def initialize (self)`
Initializes the milieu (composite).
- `def call (self, cmdline, shell=False, decode=True, raise_on_errors=None, split_output_channels=False)`
Executes a command line.
- `def createtempdir (self, dirname=None)`
Returns a fresh [anise.framework.files.TempDir](#).

Private Member Functions

- `def _initialize (self)`
Called from outside for initialization.

Private Attributes

- `_initialized`

1.251.1 Detailed Description

Milieux are a functionality which allows a build task to work in different environments.

This is advanced functionality and requires the build task to be coded in a special way. Instead of directly executing actions, e.g. executing external commands, it must use the `call()` method of a `Milieu` instance for it.

Tasks can get a `Milieu` by calling `milieus.get()` ([internals.get](#)). Making milieux available on the other hand is possible with `defaultmilieu` and `milieus.register()` ([internals.registermilieu](#)).

See subclasses or implement an own one. See also [MilieuComposite](#).

Subclasses often not only implement methods from this class, but also add new tools to it.

1.251.2 Constructor & Destructor Documentation

1.251.2.1 `__init__()`

```
def anise.features.milieus.Milieu.__init__ (
    self )
```

1.251.3 Member Function Documentation

1.251.3.1 `_initialize()`

```
def anise.features.milieus.Milieu._initialize (
    self ) [private]
```

Called from outside for initialization.

This is part of a particular piece of internal infrastructure and is typically not required to be used.

1.251.3.2 `call()`

```
def anise.features.milieus.Milieu.call (
    self,
    cmdline,
    shell = False,
    decode = True,
    raise_on_errors = None,
    split_output_channels = False )
```

Executes a command line.

Parameters

<i>cmdline</i>	The command line as list of strings.
<i>shell</i>	If the command shall be interpreted by a shell (only the first part of cmdline will apply!).
<i>decode</i>	If the result is to be decoded to a string.
<i>raise_on_errors</i>	If errors shall raise Exceptions (also specifies the exception text in this case).
<i>split_output_channels</i>	If to separately record output and error stream.

Returns

: Tuple of returncode, program output.

1.251.3.3 createtempdir()

```
def anise.features.milieus.Milieu.createtempdir (
    self,
    dirname = None )
```

Returns a fresh [anise.framework.files.TempDir](#).

Parameters

<i>dirname</i>	An optional name which the root directory should get; if not given, it gets a random name.
----------------	--

1.251.3.4 initialize()

```
def anise.features.milieus.Milieu.initialize (
    self )
```

Initializes the milieu (composite).

Override this method in custom subclasses or leave the default implementation.

1.251.4 Member Data Documentation

1.251.4.1 _initialized

```
anise.features.milieus.Milieu._initialized [private]
```

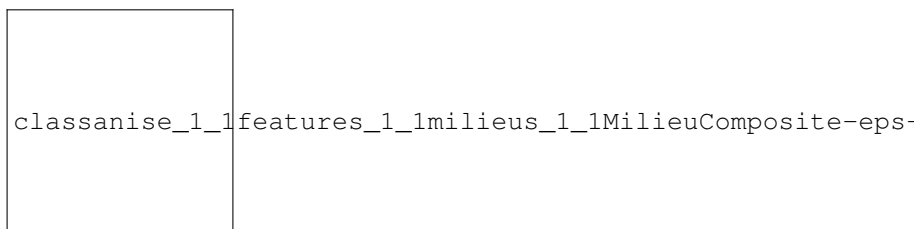
The documentation for this class was generated from the following file:

- [anise/features/milieus.py](#)

1.252 anise.features.milieus.MilieuComposite Class Reference

A milieu composite brings parts of functionality which can be stacked on top of a [Milieu](#).

Inheritance diagram for anise.features.milieus.MilieuComposite:



Public Member Functions

- def `__init__` (self)
- def `initialize` (self)
Initializes the milieu (composite).
- def `setinner` (self, inner)
Sets the inner milieu.
- def `__getattr__` (self, item)

Public Attributes

- `inner`

Private Member Functions

- def `_initialize` (self)
Called from outside for initialization.

Private Attributes

- `_initialized`

1.252.1 Detailed Description

A milieu composite brings parts of functionality which can be stacked on top of a [Milieu](#).

They behave largely equally to [Milieus](#).

1.252.2 Constructor & Destructor Documentation

1.252.2.1 `__init__()`

```
def anise.features.milieus.MilieuComposite.__init__ (
    self )
```

1.252.3 Member Function Documentation

1.252.3.1 `__getattr__()`

```
def anise.features.milieus.MilieuComposite.__getattr__ (
    self,
    item )
```

1.252.3.2 `_initialize()`

```
def anise.features.milieus.MilieuComposite._initialize (
    self ) [private]
```

Called from outside for initialization.

This is part of a particular piece of internal infrastructure and is typically not required to be used.

1.252.3.3 `initialize()`

```
def anise.features.milieus.MilieuComposite.initialize (
    self )
```

Initializes the milieu (composite).

Override this method in custom subclasses or leave the default implementation.

1.252.3.4 `setinner()`

```
def anise.features.milieus.MilieuComposite.setinner (
    self,
    inner )
```

Sets the inner milieu.

This is part of a particular piece of internal infrastructure and is typically not required to be used.

1.252.4 Member Data Documentation

1.252.4.1 `_initialized`

`anise.features.milieus.MilieuComposite._initialized` [private]

1.252.4.2 `inner`

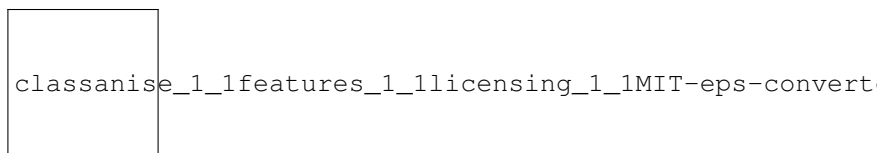
`anise.features.milieus.MilieuComposite.inner`

The documentation for this class was generated from the following file:

- [anise/features/milieus.py](#)

1.253 `anise.features.licensing.MIT` Class Reference

Inheritance diagram for `anise.features.licensing.MIT`:



Public Member Functions

- `def __init__ (self)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.253.1 Constructor & Destructor Documentation

1.253.1.1 `__init__()`

```
def anise.features.licensing.MIT.__init__ (
    self )
```

1.253.2 Member Function Documentation

1.253.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.253.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.253.3 Member Data Documentation

1.253.3.1 `filename`

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.253.3.2 `name`

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.253.3.3 `namedebian`

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.253.3.4 namepythonsetuputils

`anise.features.licensing.BaseLicense.namepythonsetuputils` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.254 anise.features.ui.Mode Class Reference

Enumeration of user interface modes.

Static Public Attributes

- [TERMINAL](#)
- [WEB](#)

1.254.1 Detailed Description

Enumeration of user interface modes.

1.254.2 Member Data Documentation

1.254.2.1 TERMINAL

`anise.features.ui.Mode.TERMINAL` [static]

1.254.2.2 WEB

`anise.features.ui.Mode.WEB` [static]

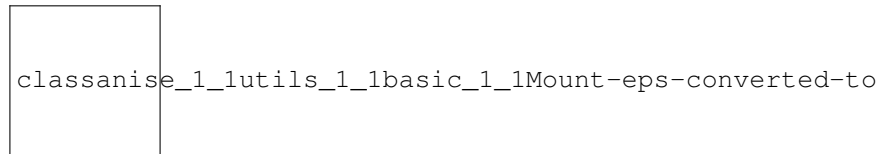
The documentation for this class was generated from the following file:

- [anise/features/ui.py](#)

1.255 anise.utils.basic.Mount Class Reference

Mounts a volume.

Inheritance diagram for anise.utils.basic.Mount:



Public Member Functions

- def `__init__` (self, `src`, `dst`, `options`=None, `mounter`=None, `unmounter`=None, `needsroot`=True)
- def `mount` (self)
 - Manually mount the volume.*
- def `__enter__` (self)
- def `unmount` (self)
 - Manually unmount the volume.*
- def `__exit__` (self, `etype`, `eval`, `etraceback`)

Public Attributes

- `su`
- `mounted`
- `src`
- `dst`
- `options`
- `mounter`
- `unmounter`
- `needsroot`

1.255.1 Detailed Description

Mounts a volume.

Can either be used automatically with the `with` keyword or manually.

1.255.2 Constructor & Destructor Documentation

1.255.2.1 `__init__()`

```

def anise.utils.basic.Mount.__init__ (
    self,
    src,
    dst,
    options = None,
    mounter = None,
    unmounter = None,
    needsroot = True )
  
```

Parameters

<i>src</i>	Mount source.
<i>dst</i>	Mount destination.
<i>options</i>	Optional additional mount options.
<i>mounter</i>	Optional string list overriding the default mount call.
<i>unmounter</i>	Optional string list overriding the default unmount call.
<i>needsroot</i>	If this mount action will need root permissions.

1.255.3 Member Function Documentation

1.255.3.1 `__enter__()`

```
def anise.utils.basic.Mount.__enter__ (
    self )
```

1.255.3.2 `__exit__()`

```
def anise.utils.basic.Mount.__exit__ (
    self,
    etype,
    evalue,
    etraceback )
```

1.255.3.3 `mount()`

```
def anise.utils.basic.Mount.mount (
    self )
```

Manually mount the volume.

1.255.3.4 `unmount()`

```
def anise.utils.basic.Mount.unmount (
    self )
```

Manually unmount the volume.

1.255.4 Member Data Documentation

1.255.4.1 dst

`anise.utils.basic.Mount.dst`

1.255.4.2 mounted

`anise.utils.basic.Mount.mounted`

1.255.4.3 mounter

`anise.utils.basic.Mount.mounter`

1.255.4.4 needsroot

`anise.utils.basic.Mount.needsroot`

1.255.4.5 options

`anise.utils.basic.Mount.options`

1.255.4.6 src

`anise.utils.basic.Mount.src`

1.255.4.7 su

`anise.utils.basic.Mount.su`

1.255.4.8 unmounter

`anise.utils.basic.Mount.unmounter`

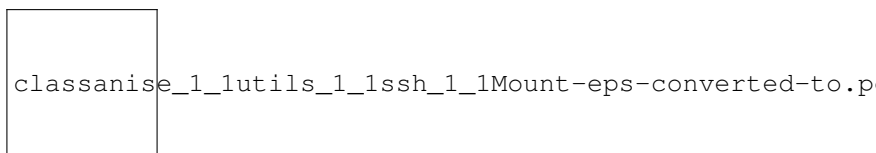
The documentation for this class was generated from the following file:

- [anise/utils/basic.py](#)

1.256 anise.utils.ssh.Mount Class Reference

Mounts a volume via ssh.

Inheritance diagram for `anise.utils.ssh.Mount`:



Public Member Functions

- `def __init__(self, src, dst, options=None, port=22, identityfile=None)`
- `def mount(self)`
Manually mount the volume.
- `def __enter__(self)`
- `def unmount(self)`
Manually unmount the volume.
- `def __exit__(self, etype, evalue, etraceback)`

Public Attributes

- `su`
- `mounted`
- `src`
- `dst`
- `options`
- `mounter`
- `unmounter`
- `needsroot`

1.256.1 Detailed Description

Mounts a volume via ssh.

1.256.2 Constructor & Destructor Documentation

1.256.2.1 __init__()

```
def anise.utils.ssh.Mount.__init__ (
    self,
    src,
    dst,
    options = None,
    port = 22,
    identityfile = None )
```

Parameters

<i>src</i>	The remote mount source (an ssh address like <code>user\@machine</code>).
<i>dst</i>	The local mount destination.
<i>options</i>	Optional additional mount options.
<i>port</i>	Optional tcp port.
<i>identityfile</i>	Optional identity file for authentication.

1.256.3 Member Function Documentation

1.256.3.1 __enter__()

```
def anise.utils.basic.Mount.__enter__ (
    self ) [inherited]
```

1.256.3.2 __exit__()

```
def anise.utils.basic.Mount.__exit__ (
    self,
    etype,
    evalue,
    etraceback ) [inherited]
```

1.256.3.3 mount()

```
def anise.utils.basic.Mount.mount (
    self ) [inherited]
```

Manually mount the volume.

1.256.3.4 unmount()

```
def anise.utils.basic.Mount.unmount (
    self ) [inherited]
```

Manually unmount the volume.

1.256.4 Member Data Documentation

1.256.4.1 dst

```
anise.utils.basic.Mount.dst [inherited]
```

1.256.4.2 mounted

```
anise.utils.basic.Mount.mounted [inherited]
```

1.256.4.3 mounter

```
anise.utils.basic.Mount.mounter [inherited]
```

1.256.4.4 needsroot

```
anise.utils.basic.Mount.needsroot [inherited]
```

1.256.4.5 options

```
anise.utils.basic.Mount.options [inherited]
```

1.256.4.6 src

```
anise.utils.basic.Mount.src [inherited]
```


1.256.4.7 su

`anise.utils.basic.Mount.su` [inherited]

1.256.4.8 unmounter

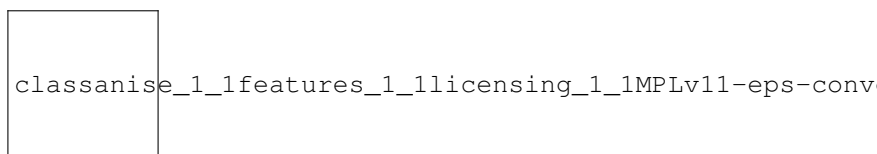
`anise.utils.basic.Mount.unmounter` [inherited]

The documentation for this class was generated from the following file:

- [anise/utils/ssh.py](#)

1.257 anise.features.licensing.MPLv11 Class Reference

Inheritance diagram for `anise.features.licensing.MPLv11`:



Public Member Functions

- `def __init__ (self)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.257.1 Constructor & Destructor Documentation

1.257.1.1 `__init__()`

```
def anise.features.licensing.MPLv11.__init__ (
    self )
```

1.257.2 Member Function Documentation

1.257.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.257.2.2 `getlicensetext()`

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.257.3 Member Data Documentation

1.257.3.1 `filename`

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.257.3.2 `name`

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.257.3.3 `namedebian`

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.257.3.4 namepythonsetuputils

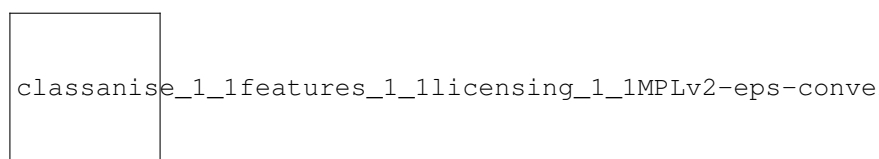
`anise.features.licensing.BaseLicense.namepythonsetuputils` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.258 anise.features.licensing.MPLv2 Class Reference

Inheritance diagram for `anise.features.licensing.MPLv2`:



Public Member Functions

- `def __init__ (self)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.258.1 Constructor & Destructor Documentation

1.258.1.1 `__init__()`

```
def anise.features.licensing.MPLv2.__init__ (
    self )
```

1.258.2 Member Function Documentation

1.258.2.1 getfilename()

```
def anise.features.licensing.BaseLicense.getfilename (
    self ) [inherited]
```

Returns the full path of the license text file.

1.258.2.2 getlicensetext()

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.258.3 Member Data Documentation

1.258.3.1 filename

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.258.3.2 name

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.258.3.3 namedebian

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.258.3.4 namepythonsetuputils

```
anise.features.licensing.BaseLicense.namepythonsetuputils [inherited]
```

The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.259 test_custom_processor.MyObject Class Reference

Public Member Functions

- def `__init__` (self, `s`)
- def `up` (self)
- def `lo` (self)

Public Attributes

- `s`

1.259.1 Constructor & Destructor Documentation

1.259.1.1 `__init__()`

```
def test_custom_processor.MyObject.__init__ (  
    self,  
    s )
```

1.259.2 Member Function Documentation

1.259.2.1 `lo()`

```
def test_custom_processor.MyObject.lo (  
    self )
```

1.259.2.2 `up()`

```
def test_custom_processor.MyObject.up (  
    self )
```

1.259.3 Member Data Documentation

1.259.3.1 s

```
test_custom_processor.MyObject.s
```

The documentation for this class was generated from the following file:

- [anise/data/PyDepsEngine/_meta/test_custom_processor.py](#)

1.260 test_custom_processor_filterobjects_completelistbyrequirements.MyProcessor Class Reference

Inheritance diagram for test_custom_processor_filterobjects_completelistbyrequirements.MyProcessor:



Public Member Functions

- [def __init__](#) (self)

Private Member Functions

- [def _filter_objects](#) (self, pobj)

1.260.1 Constructor & Destructor Documentation

1.260.1.1 __init__()

```
def test_custom_processor_filterobjects_completelistbyrequirements.MyProcessor.__init__ (
    self )
```

1.260.2 Member Function Documentation

1.260.2.1 _filter_objects()

```
def test_custom_processor_filterobjects_completelistbyrequirements.MyProcessor._filter_objects
(
    self,
    pobj ) [private]
```

The documentation for this class was generated from the following file:

- [anise/data/PyDepsEngine/_meta/test_custom_processor_filterobjects_completelistbyrequirements.py](#)

1.261 test_custom_processor.MyProcessor Class Reference

Inheritance diagram for test_custom_processor.MyProcessor:



Public Member Functions

- [def __init__](#) (self, reverse)

Private Member Functions

- [def _update_object_deps](#) (self, pobj)

Private Attributes

- [_reverse](#)

1.261.1 Constructor & Destructor Documentation

1.261.1.1 __init__()

```
def test_custom_processor.MyProcessor.__init__ (
    self,
    reverse )
```

1.261.2 Member Function Documentation

1.261.2.1 `_update_object_deps()`

```
def test_custom_processor.MyProcessor._update_object_deps (
    self,
    pobj ) [private]
```

1.261.3 Member Data Documentation

1.261.3.1 `_reverse`

```
test_custom_processor.MyProcessor._reverse [private]
```

The documentation for this class was generated from the following file:

- anise/data/PyDepsEngine/_meta/[test_custom_processor.py](#)

1.262 `test_custom_processor_filterobjects.MyProcessor` Class Reference

Inheritance diagram for `test_custom_processor_filterobjects.MyProcessor`:



Public Member Functions

- def [__init__](#) (self)

Private Member Functions

- def [_filter_objects](#) (self, pobj)

1.262.1 Constructor & Destructor Documentation

1.262.1.1 `__init__()`

```
def test_custom_processor_filterobjects.MyProcessor.__init__ (
    self )
```

1.262.2 Member Function Documentation

1.262.2.1 `_filter_objects()`

```
def test_custom_processor_filterobjects.MyProcessor._filter_objects (
    self,
    pobj ) [private]
```

The documentation for this class was generated from the following file:

- [anise/data/PyDepsEngine/_meta/test_custom_processor_filterobjects.py](#)

1.263 `anise.features.ui.ChoiceTree.Node` Class Reference

One node in a tree of possible choices.

Public Member Functions

- def `__init__` (self, [key](#), [val](#), [parent](#), [childs](#), [image](#))
- def `createchild` (self, [key](#), [val](#), [image](#)=None)
Creates a new node in tree as as child of this node.

Public Attributes

- [key](#)
- [val](#)
- [parent](#)
- [childs](#)
- [image](#)

1.263.1 Detailed Description

One node in a tree of possible choices.

1.263.2 Constructor & Destructor Documentation

1.263.2.1 `__init__()`

```
def anise.features.ui.ChoiceTree.Node.__init__ (
    self,
    key,
    val,
    parent,
    childs,
    image )
```

1.263.3 Member Function Documentation

1.263.3.1 `createchild()`

```
def anise.features.ui.ChoiceTree.Node.createchild (
    self,
    key,
    val,
    image = None )
```

Creates a new node in tree as as child of this node.

Parameters

<i>key</i>	The key string. This is displayed in tree.
<i>val</i>	The value description string. It is displayed in the tree as well.
<i>image</i>	Icon name.

Returns

: The new [ChoiceTree.Node](#).

1.263.4 Member Data Documentation

1.263.4.1 `childs`

```
anise.features.ui.ChoiceTree.Node.childs
```

1.263.4.2 `image`

```
anise.features.ui.ChoiceTree.Node.image
```

1.263.4.3 key

```
anise.features.ui.ChoiceTree.Node.key
```

1.263.4.4 parent

```
anise.features.ui.ChoiceTree.Node.parent
```

1.263.4.5 val

```
anise.features.ui.ChoiceTree.Node.val
```

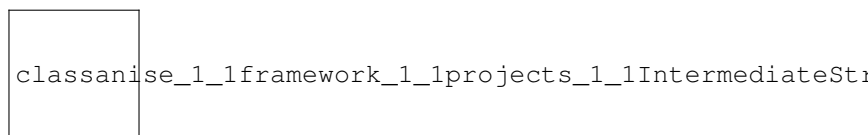
The documentation for this class was generated from the following file:

- [anise/features/ui.py](#)

1.264 anise.framework.projects.IntermediateStructure.Node Class Reference

A node in the intermediate structure.

Inheritance diagram for anise.framework.projects.IntermediateStructure.Node:



Public Member Functions

- def `__init__` (self, k, args)
- def `key` (self)
The key name.
- def `parent` (self)
Returns the parent node (either `IntermediateStructure` or `IntermediateStructure.Node`).
- def `args` (self)
Returns the list of stored argument keys.
- def `getarg` (self, key)
Returns an argument (as `IntermediateStructure.Value`).
- def `setarg` (self, key, node)
Sets an argument.
- def `removearg` (self, key)
Removes an argument.
- def `clearargs` (self)
Clears the list of arguments.
- def `remove` (self)
Removes this node.

Public Attributes

- [islet](#)
- [isvalue](#)

Private Attributes

- [_parent](#)
- [_args](#)
- [_key](#)

1.264.1 Detailed Description

A node in the intermediate structure.

1.264.2 Constructor & Destructor Documentation

1.264.2.1 `__init__()`

```
def anise.framework.projects.IntermediateStructure.Node.__init__ (
    self,
    k,
    args )
```

Parameters

<i>k</i>	The node key.
----------	---------------

1.264.3 Member Function Documentation

1.264.3.1 `args()`

```
def anise.framework.projects.IntermediateStructure.Node.args (
    self )
```

Returns the list of stored argument keys.

1.264.3.2 clearargs()

```
def anise.framework.projects.IntermediateStructure.Node.clearargs (
    self )
```

Clears the list of arguments.

1.264.3.3 getarg()

```
def anise.framework.projects.IntermediateStructure.Node.getarg (
    self,
    key )
```

Returns an argument (as [IntermediateStructure.Value](#)).

Parameters

<i>key</i>	The argument key.
------------	-------------------

1.264.3.4 key()

```
def anise.framework.projects.IntermediateStructure.Node.key (
    self )
```

The key name.

1.264.3.5 parent()

```
def anise.framework.projects.IntermediateStructure.Node.parent (
    self )
```

Returns the parent node (either [IntermediateStructure](#) or [IntermediateStructure.Node](#)).

1.264.3.6 remove()

```
def anise.framework.projects.IntermediateStructure.Node.remove (
    self )
```

Removes this node.

1.264.3.7 removearg()

```
def anise.framework.projects.IntermediateStructure.Node.removearg (
    self,
    key )
```

Removes an argument.

Parameters

<i>key</i>	The argument key.
------------	-------------------

1.264.3.8 setarg()

```
def anise.framework.projects.IntermediateStructure.Node.setarg (
    self,
    key,
    node )
```

Sets an argument.

Parameters

<i>key</i>	The argument key.
<i>node</i>	The argument value (as IntermediateStructure.Value).

1.264.4 Member Data Documentation

1.264.4.1 _args

```
anise.framework.projects.IntermediateStructure.Node._args [private]
```

1.264.4.2 _key

```
anise.framework.projects.IntermediateStructure.Node._key [private]
```

1.264.4.3 _parent

```
anise.framework.projects.IntermediateStructure.Node._parent [private]
```

1.264.4.4 islet

```
anise.framework.projects.IntermediateStructure.Node.islet
```

1.264.4.5 isvalue

```
anise.framework.projects.IntermediateStructure.Node.isvalue
```

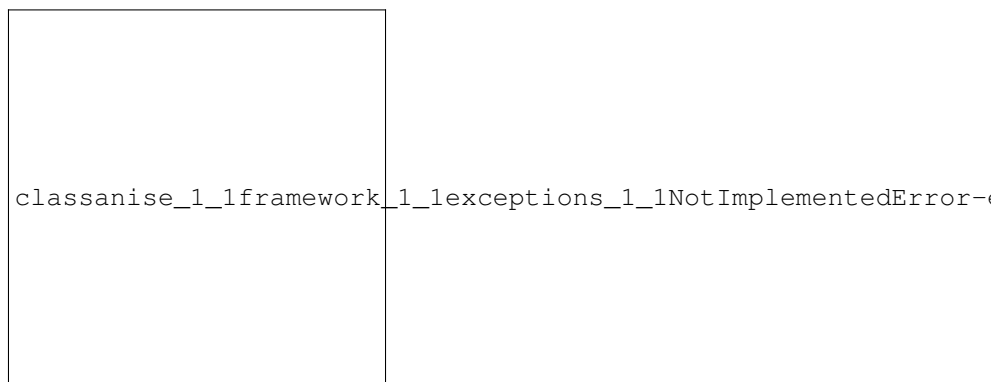
The documentation for this class was generated from the following file:

- [anise/framework/projects.py](#)

1.265 anise.framework.exceptions.NotImplementedError Class Reference

Some unimplemented functionality was called.

Inheritance diagram for anise.framework.exceptions.NotImplementedError:



Public Member Functions

- [def __init__](#) (self)
- [def __call__](#) (self, args)

1.265.1 Detailed Description

Some unimplemented functionality was called.

1.265.2 Constructor & Destructor Documentation

1.265.2.1 __init__()

```
def anise.framework.exceptions.NotImplementedError.__init__ (  
    self )
```

1.265.3 Member Function Documentation

1.265.3.1 `__call__()`

```
def anise.framework.exceptions.AniseError.__call__ (
    self,
    args ) [inherited]
```

The documentation for this class was generated from the following file:

- [anise/framework/exceptions.py](#)

1.266 `anise.data.PyDepsEngine.depsengine.Processor.Object` Class Reference

A processor object holds an entity object together with dependency details.

Public Member Functions

- `def __str__ (self)`
- `def __init__ (self, o, processor, own, afterrequired, beforerequired, afteroptional, beforeoptional, cacheable)`
See See [Engine.add_object](#) for details about the parameters.
- `def update (self, own, afterrequired, beforerequired, afteroptional, beforeoptional)`
Updates the dependency details for this processor object.

Public Attributes

- `o`
- `processor`
- `own`
- `afterrequired`
- `beforerequired`
- `afteroptional`
- `beforeoptional`
- `after`
- `before`
- `required`
- `cacheable`

Static Private Member Functions

- `def __getlist (l)`
Sanitizes `l` to an actual list.

1.266.1 Detailed Description

A processor object holds an entity object together with dependency details.

Instantiated by the [Processor](#) infrastructure.

1.266.2 Constructor & Destructor Documentation

1.266.2.1 `__init__()`

```
def anise.data.PyDepsEngine.depsengine.Processor.Object.__init__ (
    self,
    o,
    processor,
    own,
    afterrequired,
    beforerequired,
    afteroptional,
    beforeoptional,
    cacheable )
```

See See [Engine.add_object](#) for details about the parameters.

Parameters

<code>cacheable</code>	If the dependencies are cacheable (just a hint; internal code may restrict cacheability further).
------------------------	---

1.266.3 Member Function Documentation

1.266.3.1 `__str__()`

```
def anise.data.PyDepsEngine.depsengine.Processor.Object.__str__ (
    self )
```

1.266.3.2 `_getlist()`

```
def anise.data.PyDepsEngine.depsengine.Processor.Object._getlist (
    l ) [static], [private]
```

Sanitizes `l` to an actual list.

If input is a string, it gets split. If input is a function, it gets called.

Parameters

/	The list (or some other stuff - see above) to sanitize.
---	---

1.266.3.3 update()

```
def anise.data.PyDepsEngine.depsengine.Processor.Object.update (
    self,
    own,
    afterrequired,
    beforerequired,
    afteroptional,
    beforeoptional )
```

Updates the dependency details for this processor object.

See See [Engine.add_object](#) for details about the parameters.

1.266.4 Member Data Documentation**1.266.4.1 after**

```
anise.data.PyDepsEngine.depsengine.Processor.Object.after
```

1.266.4.2 afteroptional

```
anise.data.PyDepsEngine.depsengine.Processor.Object.afteroptional
```

1.266.4.3 afterrequired

```
anise.data.PyDepsEngine.depsengine.Processor.Object.afterrequired
```

1.266.4.4 before

```
anise.data.PyDepsEngine.depsengine.Processor.Object.before
```

1.266.4.5 beforeoptional

`anise.data.PyDepsEngine.depsengine.Processor.Object.beforeoptional`

1.266.4.6 beforerequired

`anise.data.PyDepsEngine.depsengine.Processor.Object.beforerequired`

1.266.4.7 cacheable

`anise.data.PyDepsEngine.depsengine.Processor.Object.cacheable`

1.266.4.8 o

`anise.data.PyDepsEngine.depsengine.Processor.Object.o`

1.266.4.9 own

`anise.data.PyDepsEngine.depsengine.Processor.Object.own`

1.266.4.10 processor

`anise.data.PyDepsEngine.depsengine.Processor.Object.processor`

1.266.4.11 required

`anise.data.PyDepsEngine.depsengine.Processor.Object.required`

The documentation for this class was generated from the following file:

- `anise/data/PyDepsEngine/depsengine.py`

1.267 anise.features.signing.OsslsigncodeConfiguration Class Reference

Signing configuration for signing a win32 binary with the 'osslsigncode' tool.

Public Member Functions

- `def __init__(self, p12path, intermediatepath=None, osslsigncodepath="osslsigncode", opensslpath="openssl", password="")`

Public Attributes

- `osslsigncodepath`
- `opensslpath`
- `p12path`
- `intermediatepath`
- `password`

1.267.1 Detailed Description

Signing configuration for signing a win32 binary with the 'osslsigncode' tool.

1.267.2 Constructor & Destructor Documentation

1.267.2.1 __init__()

```
def anise.features.signing.OsslsigncodeConfiguration.__init__(
    self,
    p12path,
    intermediatepath = None,
    osslsigncodepath = "osslsigncode",
    opensslpath = "openssl",
    password = "" )
```

Parameters

<i>p12path</i>	Path to the .p12 file.
<i>intermediatepath</i>	Path to intermediate certificates.
<i>osslsigncodepath</i>	Path to the osslsigncode tool.
<i>opensslpath</i>	Path to the openssl tool.
<i>password</i>	The certificate passphrase.

1.267.3 Member Data Documentation

1.267.3.1 intermediatepath

`anise.features.signing.OsslsigncodeConfiguration.intermediatepath`

1.267.3.2 opensslpath

`anise.features.signing.OsslsigncodeConfiguration.opensslpath`

1.267.3.3 osslsigncodepath

`anise.features.signing.OsslsigncodeConfiguration.osslsigncodepath`

1.267.3.4 p12path

`anise.features.signing.OsslsigncodeConfiguration.p12path`

1.267.3.5 password

`anise.features.signing.OsslsigncodeConfiguration.password`

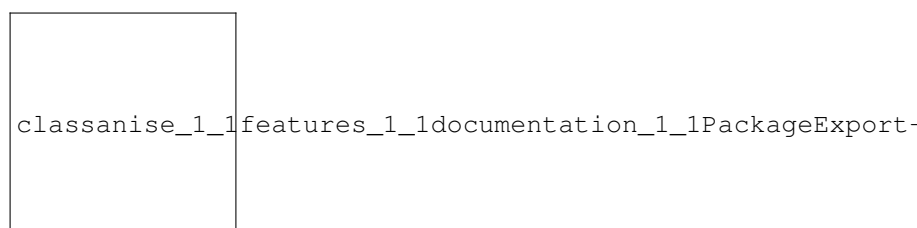
The documentation for this class was generated from the following file:

- [anise/features/signing.py](#)

1.268 anise.features.documentation.PackageExport Class Reference

Describes an export of an anise documentation into the distribution package(s).

Inheritance diagram for `anise.features.documentation.PackageExport`:



Public Member Functions

- `def __init__ (self, key, outputfile, format, name=None, heading=None, includedeveloperdoc=False, extractall=True, extractprivate=True, hideundocumented=False, imagepaths=None, additionaltext="", shownavbar=False, excludepatterns=None, predefineddocmacros=None, head3=None, head4=None, source=None, sources=None)`
- `def generate (self, targetpath)`
Generates the content.

Public Attributes

- `outputfile`
- `key`
- `format`
- `name`
- `heading`
- `includedeveloperdoc`
- `extractall`
- `extractprivate`
- `hideundocumented`
- `imagepaths`
- `additionaltext`
- `shownavbar`
- `excludepatterns`
- `predefineddocmacros`
- `head3`
- `head4`
- `source`
- `sources`

1.268.1 Detailed Description

Describes an export of an anise documentation into the distribution package(s).

1.268.2 Constructor & Destructor Documentation

1.268.2.1 __init__()

```
def anise.features.documentation.PackageExport.__init__ (
    self,
    key,
    outputfile,
    format,
    name = None,
    heading = None,
    includedeveloperdoc = False,
    extractall = True,
    extractprivate = True,
```

```

hideundocumented = False,
imagepaths = None,
additionaltext = "",
shownavbar = False,
excludepatterns = None,
predefineddocmacros = None,
head3 = None,
head4 = None,
source = None,
sources = None )

```

Parameters

<i>key</i>	One of the keys stored in pool. Defines which documentation to export.
<i>outputfile</i>	Target file path relative to the 'raw package's root.
<i>format</i>	The output format (one of documentation.ExportFormat).
<i>name</i>	The export definition name. Finding definitions by name is possible with <code>exports.getbyname</code> .
<i>heading</i>	The title.
<i>includedeveloperdoc</i>	If the output shall include the developer documentation as well (does not work for all output formats).
<i>extractall</i>	If a developer documentation should include all elements (not just the documented ones).
<i>extractprivate</i>	If a developer documentation should include also private members.
<i>hideundocumented</i>	If the develo documentation should entirely leave out undocumented members.
<i>imagepaths</i>	Directories from where images are used in the documentation.
<i>additionaltext</i>	Additional text to append to the selected documentation.
<i>shownavbar</i>	If to show navigation bar. Does not work in all output formats.
<i>excludepatterns</i>	File patterns to exclude from reading.
<i>predefineddocmacros</i>	List of additional macros to define for reading.
<i>head3</i>	3rd heading.
<i>head4</i>	4th heading.
<i>source</i>	The documentation source root.
<i>sources</i>	Optional list of source paths (default: ["/"]).

1.268.3 Member Function Documentation

1.268.3.1 generate()

```

def anise.features.documentation.internals.Export.generate (
    self,
    targetpath ) [inherited]

```

Generates the content.

Override this method in custom subclasses or leave the default implementation.

1.268.4 Member Data Documentation

1.268.4.1 additionaltext

`anise.features.documentation.internals.Export.additionaltext` [inherited]

1.268.4.2 excludepatterns

`anise.features.documentation.internals.Export.excludepatterns` [inherited]

1.268.4.3 extractall

`anise.features.documentation.internals.Export.extractall` [inherited]

1.268.4.4 extractprivate

`anise.features.documentation.internals.Export.extractprivate` [inherited]

1.268.4.5 format

`anise.features.documentation.internals.Export.format` [inherited]

1.268.4.6 head3

`anise.features.documentation.internals.Export.head3` [inherited]

1.268.4.7 head4

`anise.features.documentation.internals.Export.head4` [inherited]

1.268.4.8 heading

`anise.features.documentation.internals.Export.heading` [inherited]

1.268.4.9 hideundocumented

`anise.features.documentation.internals.Export.hideundocumented` [inherited]

1.268.4.10 imagepaths

`anise.features.documentation.internals.Export.imagepaths` [inherited]

1.268.4.11 includedeveloperdoc

`anise.features.documentation.internals.Export.includedeveloperdoc` [inherited]

1.268.4.12 key

`anise.features.documentation.internals.Export.key` [inherited]

1.268.4.13 name

`anise.features.documentation.internals.Export.name` [inherited]

1.268.4.14 outputfile

`anise.features.documentation.internals.AbstractFileExport.outputfile` [inherited]

1.268.4.15 predefineddocmacros

`anise.features.documentation.internals.Export.predefineddocmacros` [inherited]

1.268.4.16 shownavbar

```
anise.features.documentation.internals.Export.shownavbar [inherited]
```

1.268.4.17 source

```
anise.features.documentation.internals.Export.source [inherited]
```

1.268.4.18 sources

```
anise.features.documentation.internals.Export.sources [inherited]
```

The documentation for this class was generated from the following file:

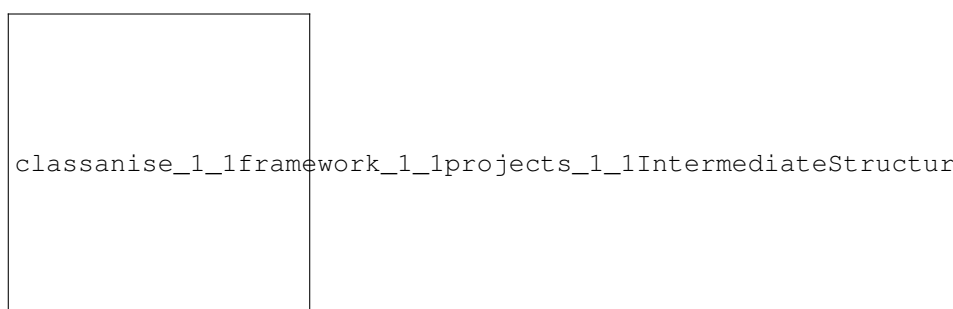
- anise/features/[documentation.py](#)

1.269 anise.framework.projects.IntermediateStructure.ParseProjectDescriptionError

Class Reference

Parsing error like bad input xml.

Inheritance diagram for anise.framework.projects.IntermediateStructure.ParseProjectDescriptionError:

**Public Member Functions**

- def [__call__](#) (self, args)

1.269.1 Detailed Description

Parsing error like bad input xml.

1.269.2 Member Function Documentation

1.269.2.1 __call__()

```
def anise.framework.exceptions.AniseError.__call__ (
    self,
    args ) [inherited]
```

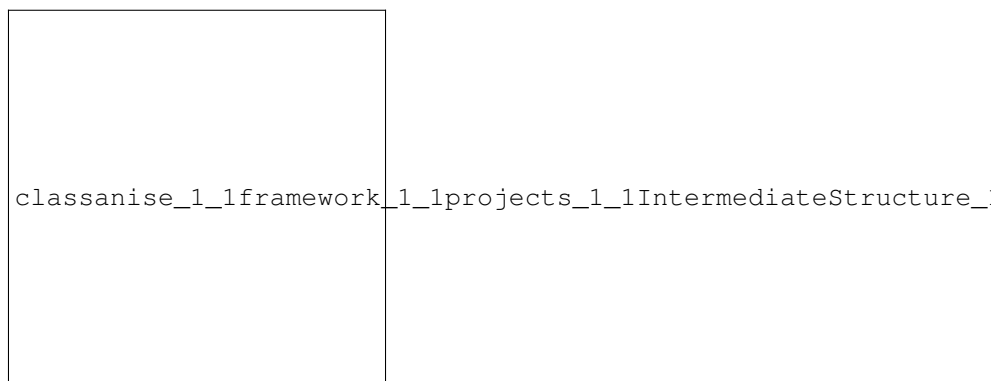
The documentation for this class was generated from the following file:

- [anise/framework/projects.py](#)

1.270 anise.framework.projects.IntermediateStructure.ParseProjectDescriptionInternalError Class Reference

Parsing error which seems like a bug.

Inheritance diagram for anise.framework.projects.IntermediateStructure.ParseProjectDescriptionInternalError:



Public Member Functions

- [def __call__](#) (self, args)

1.270.1 Detailed Description

Parsing error which seems like a bug.

1.270.2 Member Function Documentation

1.270.2.1 `__call__()`

```
def anise.framework.exceptions.AniseError.__call__ (
    self,
    args ) [inherited]
```

The documentation for this class was generated from the following file:

- [anise/framework/projects.py](#)

1.271 `anise.features.dependencies.internals.Pool` Class Reference

Storage of all dependencies.

Public Member Functions

- `def __init__ (self)`
- `def add (self, dependency)`
Adds a dependency.

Public Attributes

- [list](#)

1.271.1 Detailed Description

Storage of all dependencies.

1.271.2 Constructor & Destructor Documentation

1.271.2.1 `__init__()`

```
def anise.features.dependencies.internals.Pool.__init__ (
    self )
```

1.271.3 Member Function Documentation

1.271.3.1 `add()`

```
def anise.features.dependencies.internals.Pool.add (
    self,
    dependency )
```

Adds a dependency.

Parameters

<code>dependency</code>	A Dependency instance.
-------------------------	--

1.271.4 Member Data Documentation

1.271.4.1 list

```
anise.features.dependencies.internals.Pool.list
```

The documentation for this class was generated from the following file:

- [anise/features/dependencies.py](#)

1.272 anise.features.medialogalleries.internals.Pool Class Reference

Storage of all image galleries.

Public Member Functions

- `def __init__ (self)`
- `def add (self, gallery, name="")`
Adds an image gallery.

Public Attributes

- [list](#)

1.272.1 Detailed Description

Storage of all image galleries.

1.272.2 Constructor & Destructor Documentation

1.272.2.1 `__init__()`

```
def anise.features.medialogalleries.internals.Pool.__init__ (  
    self )
```

1.272.3 Member Function Documentation

1.272.3.1 add()

```
def anise.features.mediagalleries.internals.Pool.add (
    self,
    gallery,
    name = "" )
```

Adds an image gallery.

Parameters

<i>gallery</i>	An instance of MediaGallery specifying the gallery details.
<i>name</i>	Gallery name.

1.272.4 Member Data Documentation

1.272.4.1 list

```
anise.features.mediagalleries.internals.Pool.list
```

The documentation for this class was generated from the following file:

- anise/features/[mediagalleries.py](#)

1.273 anise.features.distributables.internals.Pool Class Reference

Storage of all dependencies.

Public Member Functions

- def [__init__](#) (self)
- def [addgroup](#) (self, sources, params)
Adds a group of distributables.
- def [find_by_name](#) (self, name)
Searches a [FileGroup](#) by name and returns it.
- def [get](#) (self)
Prepares the pool for further processing (i.e.

Public Attributes

- [list](#)
- [initialized](#)

1.273.1 Detailed Description

Storage of all dependencies.

1.273.2 Constructor & Destructor Documentation

1.273.2.1 `__init__()`

```
def anise.features.distributables.internals.Pool.__init__ (
    self )
```

1.273.3 Member Function Documentation

1.273.3.1 `addgroup()`

```
def anise.features.distributables.internals.Pool.addgroup (
    self,
    sources,
    params )
```

Adds a group of distributables.

Each group is stored as [FileGroup](#) instance.

Parameters

<i>sources</i>	The generator tasks, which returns the distributables.
<i>params</i>	Additional parameters for the generator tasks.

1.273.3.2 `find_by_name()`

```
def anise.features.distributables.internals.Pool.find_by_name (
    self,
    name )
```

Searches a [FileGroup](#) by name and returns it.

Parameters

<i>name</i>	The name of the group to find.
-------------	--------------------------------

1.273.3.3 get()

```
def anise.features.distributables.internals.Pool.get (
    self )
```

Prepares the pool for further processing (i.e.

firing HOOK_PREPARE_DISTRIBUTABLES_POOL) and returns the pool.

1.273.4 Member Data Documentation

1.273.4.1 initialized

```
anise.features.distributables.internals.Pool.initialized
```

1.273.4.2 list

```
anise.features.distributables.internals.Pool.list
```

The documentation for this class was generated from the following file:

- anise/features/[distributables.py](#)

1.274 anise.features.documentation.internals.Pool Class Reference

Storage of all documentation text generators.

Public Member Functions

- def [__init__](#) (self)
- def [add](#) (self, key, src)
Adds a documentation.
- def [gettext](#)s (self)
Returns all texts.
- def [load_anisedoc_source](#) (self, src)
Loads the content of anisedoc sources (which contain ANISEDOC tags) into the documentation store.

Public Attributes

- [list](#)

1.274.1 Detailed Description

Storage of all documentation text generators.

1.274.2 Constructor & Destructor Documentation

1.274.2.1 `__init__()`

```
def anise.features.documentation.internals.Pool.__init__ (
    self )
```

1.274.3 Member Function Documentation

1.274.3.1 `add()`

```
def anise.features.documentation.internals.Pool.add (
    self,
    key,
    src )
```

Adds a documentation.

Parameters

<i>key</i>	The documentation key for storage (later used for referencing this text).
<i>src</i>	The documentation source text (or a function which returns one).

1.274.3.2 `gettext()`

```
def anise.features.documentation.internals.Pool.gettexts (
    self )
```

Returns all texts.

Returns

: A list of key/value pairs. Both are always strings.

1.274.3.3 load_anisedoc_source()

```
def anise.features.documentation.internals.Pool.load_anisedoc_source (
    self,
    src )
```

Loads the content of anisedoc sources (which contain ANISEDOC tags) into the documentation store.

Parameters

<code>src</code>	File or directory with anisedoc sources.
------------------	--

1.274.4 Member Data Documentation**1.274.4.1 list**

```
anise.features.documentation.internals.Pool.list
```

The documentation for this class was generated from the following file:

- [anise/features/documentation.py](#)

1.275 anise.features.datainjections.internals.Pool Class Reference**Public Member Functions**

- def [add](#) (self, method, outfile, proppnames=None, do_in_source=False, params)
Add a data injection.

1.275.1 Detailed Description

Storage of all data injections.

1.275.2 Member Function Documentation

1.275.2.1 add()

```
def anise.features.datainjections.internals.Pool.add (
    self,
    method,
    outfile,
    propnames = None,
    do_in_source = False,
    params )
```

Add a data injection.

Parameters

<i>method</i>	The implementation for the injection. This is the name of a function with the parameters 'outfile' and 'data', writing the data (a dict with key/value pairs) to a given file in some format (as absolute path or relative to the project root directory).
<i>outfile</i>	Path to the target file for the injections (relative to anise project root path).
<i>propnames</i>	List of entry names you want to inject (beyond the few default ones).
<i>do_in_source</i>	If the source itself should be updated as well (instead of only the on-the-fly copies for packages).
<i>params</i>	Additional parameters for calling the injection implementation.

The documentation for this class was generated from the following file:

- [anise/features/datainjections.py](#)

1.276 anise.features.packages.appc.PortDescription Class Reference

Description for Appc network ports to be added to the container image.

Public Member Functions

- `def __init__(self, name, protocol, port)`

Public Attributes

- [name](#)
- [protocol](#)
- [port](#)

1.276.1 Detailed Description

Description for Appc network ports to be added to the container image.

1.276.2 Constructor & Destructor Documentation

1.276.2.1 __init__()

```
def anise.features.packages.appc.PortDescription.__init__ (
    self,
    name,
    protocol,
    port )
```

Parameters

<i>name</i>	The port name.
<i>protocol</i>	The protocol (typically "tcp" or "udp").
<i>port</i>	The port number as available inside the container.

1.276.3 Member Data Documentation

1.276.3.1 name

```
anise.features.packages.appc.PortDescription.name
```

1.276.3.2 port

```
anise.features.packages.appc.PortDescription.port
```

1.276.3.3 protocol

```
anise.features.packages.appc.PortDescription.protocol
```

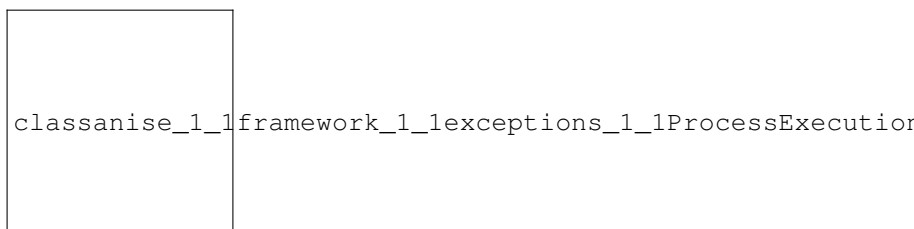
The documentation for this class was generated from the following file:

- [anise/features/packages/appc.py](#)

1.277 anise.framework.exceptions.ProcessExecutionFailedError Class Reference

The execution of an external process failed.

Inheritance diagram for anise.framework.exceptions.ProcessExecutionFailedError:



Public Member Functions

- `def __init__ (self, text, output=None)`
- `def __call__ (self, args)`

Public Attributes

- `output`

1.277.1 Detailed Description

The execution of an external process failed.

1.277.2 Constructor & Destructor Documentation

1.277.2.1 __init__()

```
def anise.framework.exceptions.ProcessExecutionFailedError.__init__ (
    self,
    text,
    output = None )
```

1.277.3 Member Function Documentation

1.277.3.1 `__call__()`

```
def anise.framework.exceptions.AniseError.__call__ (
    self,
    args ) [inherited]
```

1.277.4 Member Data Documentation

1.277.4.1 `output`

```
anise.framework.exceptions.ProcessExecutionFailedError.output
```

The documentation for this class was generated from the following file:

- [anise/framework/exceptions.py](#)

1.278 anise.data.PyDepsEngine.depsengine.Processor Class Reference

A dependency processor manages a list of entity objects, based on top of an [Engine](#), but sorted according to their dependencies.

Classes

- class [Object](#)

A processor object holds an entity object together with dependency details.

Public Member Functions

- def `__init__` (self, filter_by_owns=None, presort_keyfct=None)
- def `initialize` (self, [engine](#))
- def `set_filter_by_owns` (self, owns=None)
Filters this processor so it returns only those entity objects, which list at least one of `owns` in its 'own'.
- def `get_objects` (self)
Returns the list of entity objects, correctly sorted according their dependencies.
- def `get_processor_objects` (self)
Returns the list of processor objects as [Processor.Object](#), correctly sorted according their dependencies.

Public Attributes

- [engine](#)

Private Member Functions

- `def _ensure_filtering_sorting_valid (self)`
Ensures the objects to be sorted correctly according to their current dependencies.
- `def _filter_objects (self, l)`
Filters the list of all objects and returns that subset which is to be actually processed.
- `def _complete_list_by_requirements (self, l)`
Returns a new list of processor objects, containing all from the input list, but extended by their required dependencies.
- `def _update_object_deps (self, pobj)`
Computes dependencies for an object and calls `Processor.Object.update` in order to update dependency infos.
- `def _add_object (self, o, own=None, afterrequired=None, beforerequired=None, afteroptional=None, beforeoptional=None)`
Adds an object to this processor.
- `def _remove_object (self, o)`
Removes an object from this processor.

Private Attributes

- `_all_objects_raw`
- `_objects`
- `_filter_by_owns`
- `_presort_keyfct`
- `_sorted_filtered`

1.278.1 Detailed Description

A dependency processor manages a list of entity objects, based on top of an [Engine](#), but sorted according to their dependencies.

Each processor potentially has its own view on dependencies for the entity objects so two processors on top of the same engine can hold the entity objects in two different orders. Processors can also apply a filter and thereby returns just a subset of all entity objects.

Either use processors directly (construct one with [Engine.add_processor](#), then use [Processor.get_objects](#)) or use higher level constructs like compound objects (with [Engine.create_compound_object](#)).

1.278.2 Constructor & Destructor Documentation

1.278.2.1 `__init__()`

```
def anise.data.PyDepsEngine.depsengine.Processor.__init__ (
    self,
    filter_by_owns = None,
    presort_keyfct = None )
```


Parameters

<i>filter_by_owns</i>	The list of 'own' symbols (as string) to let the filter pass.
<i>presort_keyfct</i>	Optionally pre-sorts the list so it always has the same order regardless of the object insertion order.

1.278.3 Member Function Documentation

1.278.3.1 `_add_object()`

```
def anise.data.PyDepsEngine.depsengine.Processor._add_object (
    self,
    o,
    own = None,
    afterrequired = None,
    beforerequired = None,
    afteroptional = None,
    beforeoptional = None ) [private]
```

Adds an object to this processor.

Used by the infrastructure.

1.278.3.2 `_complete_list_by_requirements()`

```
def anise.data.PyDepsEngine.depsengine.Processor._complete_list_by_requirements (
    self,
    l ) [private]
```

Returns a new list of processor objects, containing all from the input list, but extended by their required dependencies.

This is to be used internally. It is not sorted and not checked to be valid!

Parameters

<i>l</i>	A list of processor objects as Processor.Object to complete.
----------	--

1.278.3.3 `_ensure_filtering_sorting_valid()`

```
def anise.data.PyDepsEngine.depsengine.Processor._ensure_filtering_sorting_valid (
    self ) [private]
```

Ensures the objects to be sorted correctly according to their current dependencies.

1.278.3.4 `_filter_objects()`

```
def anise.data.PyDepsEngine.depsengine.Processor._filter_objects (
    self,
    l ) [private]
```

Filters the list of all objects and returns that subset which is to be actually processed.

Parameters

<i>l</i>	A list of processor objects as Processor.Object .
----------	---

1.278.3.5 `_remove_object()`

```
def anise.data.PyDepsEngine.depsengine.Processor._remove_object (
    self,
    o ) [private]
```

Removes an object from this processor.

Used by the infrastructure.

1.278.3.6 `_update_object_deps()`

```
def anise.data.PyDepsEngine.depsengine.Processor._update_object_deps (
    self,
    pobj ) [private]
```

Computes dependencies for an object and calls [Processor.Object.update](#) in order to update dependency infos.

Subclasses may override it with custom behavior. See [Engine.add_object](#) for more details about the parts of a dependency tuple.

Parameters

<i>pobj</i>	The processor object as Processor.Object .
-------------	--

1.278.3.7 `get_objects()`

```
def anise.data.PyDepsEngine.depsengine.Processor.get_objects (
    self )
```

Returns the list of entity objects, correctly sorted according their dependencies.

1.278.3.8 get_processor_objects()

```
def anise.data.PyDepsEngine.depsengine.Processor.get_processor_objects (
    self )
```

Returns the list of processor objects as [Processor.Object](#), correctly sorted according their dependencies.

Each contains one entity object and its dependency details. Typically one should use [Processor.get_objects](#) instead, which returns just the plain entity objects as inserted.

1.278.3.9 initialize()

```
def anise.data.PyDepsEngine.depsengine.Processor.initialize (
    self,
    engine )
```

1.278.3.10 set_filter_by_owns()

```
def anise.data.PyDepsEngine.depsengine.Processor.set_filter_by_owns (
    self,
    owns = None )
```

Filters this processor so it returns only those entity objects, which list at least one of `owns` in its 'own'.

Parameters

<i>owns</i>	The list of 'own' symbols (as string) to let the filter pass.
-------------	---

1.278.4 Member Data Documentation

1.278.4.1 _all_objects_raw

```
anise.data.PyDepsEngine.depsengine.Processor._all_objects_raw [private]
```

1.278.4.2 _filter_by_owns

```
anise.data.PyDepsEngine.depsengine.Processor._filter_by_owns [private]
```

1.278.4.3 `_objects`

`anise.data.PyDepsEngine.depsengine.Processor._objects` [private]

1.278.4.4 `_presort_keyfct`

`anise.data.PyDepsEngine.depsengine.Processor._presort_keyfct` [private]

1.278.4.5 `_sorted_filtered`

`anise.data.PyDepsEngine.depsengine.Processor._sorted_filtered` [private]

1.278.4.6 `engine`

`anise.data.PyDepsEngine.depsengine.Processor.engine`

The documentation for this class was generated from the following file:

- [anise/data/PyDepsEngine/depsengine.py](#)

1.279 `anise.framework.report.ExecutionScope.Progress` Class Reference

[Progress](#) info for an execution scope.

Inheritance diagram for `anise.framework.report.ExecutionScope.Progress`:

class

`anise_1_1framework_1_1report_1_1ExecutionScope`

Public Member Functions

- `def __init__ (self)`
- `def tojson (self)`
Returns a simple object structure for json transfers.

1.279.1 Detailed Description

[Progress](#) info for an execution scope.

1.279.2 Constructor & Destructor Documentation

1.279.2.1 `__init__()`

```
def anise.framework.report.ExecutionScope.Progress.__init__ (
    self )
```

1.279.3 Member Function Documentation

1.279.3.1 `tojson()`

```
def anise.framework.report.ExecutionScope.Progress.tojson (
    self )
```

Returns a simple object structure for json transfers.

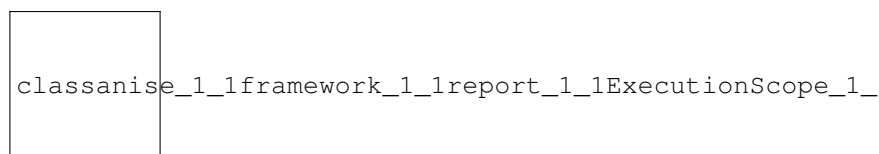
The documentation for this class was generated from the following file:

- [anise/framework/report.py](#)

1.280 anise.framework.report.ExecutionScope.ProgressDeterminate Class Reference

[Progress](#) info for a determinate progress.

Inheritance diagram for `anise.framework.report.ExecutionScope.ProgressDeterminate`:



Public Member Functions

- `def __init__(self, value)`
- `def tojson(self)`

Returns a simple object structure for json transfers.

Public Attributes

- [value](#)
- [kind](#)
- [stopped](#)

1.280.1 Detailed Description

[Progress](#) info for a determinate progress.

1.280.2 Constructor & Destructor Documentation

1.280.2.1 `__init__()`

```
def anise.framework.report.ExecutionScope.ProgressDeterminate.__init__ (
    self,
    value )
```

Parameters

<i>value</i>	The progress between 0.0 and 1.0.
--------------	-----------------------------------

1.280.3 Member Function Documentation

1.280.3.1 `tojson()`

```
def anise.framework.report.ExecutionScope.Progress.tojson (
    self ) [inherited]
```

Returns a simple object structure for json transfers.

1.280.4 Member Data Documentation

1.280.4.1 `kind`

```
anise.framework.report.ExecutionScope.ProgressDeterminate.kind
```

1.280.4.2 `stopped`

```
anise.framework.report.ExecutionScope.ProgressDeterminate.stopped
```

1.280.4.3 value

```
anise.framework.report.ExecutionScope.ProgressDeterminate.value
```

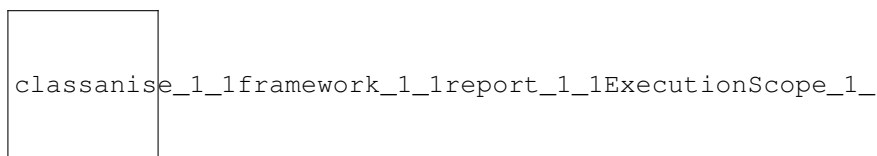
The documentation for this class was generated from the following file:

- [anise/framework/report.py](#)

1.281 anise.framework.report.ExecutionScope.ProgressFailed Class Reference

[Progress](#) info for a failed execution.

Inheritance diagram for `anise.framework.report.ExecutionScope.ProgressFailed`:



Public Member Functions

- `def __init__(self)`
- `def tojson(self)`
Returns a simple object structure for json transfers.

Public Attributes

- [kind](#)
- [stopped](#)

1.281.1 Detailed Description

[Progress](#) info for a failed execution.

1.281.2 Constructor & Destructor Documentation

1.281.2.1 __init__()

```
def anise.framework.report.ExecutionScope.ProgressFailed.__init__(
    self )
```

1.281.3 Member Function Documentation

1.281.3.1 tojson()

```
def anise.framework.report.ExecutionScope.Progress.tojson (
    self ) [inherited]
```

Returns a simple object structure for json transfers.

1.281.4 Member Data Documentation

1.281.4.1 kind

```
anise.framework.report.ExecutionScope.ProgressFailed.kind
```

1.281.4.2 stopped

```
anise.framework.report.ExecutionScope.ProgressFailed.stopped
```

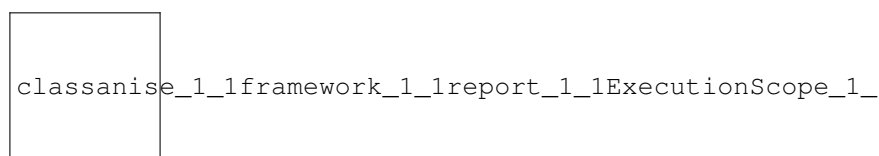
The documentation for this class was generated from the following file:

- [anise/framework/report.py](#)

1.282 anise.framework.report.ExecutionScope.ProgressFinished Class Reference

[Progress](#) info for a finished execution.

Inheritance diagram for `anise.framework.report.ExecutionScope.ProgressFinished`:



Public Member Functions

- `def __init__ (self)`
- `def tojson (self)`

Returns a simple object structure for json transfers.

Public Attributes

- [kind](#)
- [stopped](#)

1.282.1 Detailed Description

[Progress](#) info for a finished execution.

1.282.2 Constructor & Destructor Documentation

1.282.2.1 `__init__()`

```
def anise.framework.report.ExecutionScope.ProgressFinished.__init__ (
    self )
```

1.282.3 Member Function Documentation

1.282.3.1 `tojson()`

```
def anise.framework.report.ExecutionScope.Progress.tojson (
    self ) [inherited]
```

Returns a simple object structure for json transfers.

1.282.4 Member Data Documentation

1.282.4.1 `kind`

```
anise.framework.report.ExecutionScope.ProgressFinished.kind
```

1.282.4.2 `stopped`

```
anise.framework.report.ExecutionScope.ProgressFinished.stopped
```

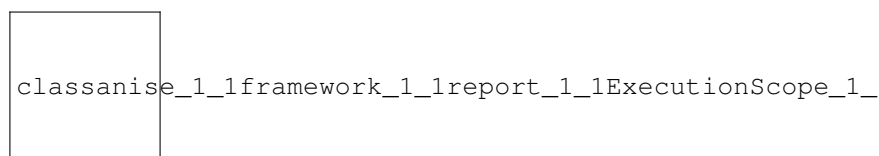
The documentation for this class was generated from the following file:

- [anise/framework/report.py](#)

1.283 anise.framework.report.ExecutionScope.ProgressIndeterminate Class Reference

[Progress](#) info for an indeterminate progress.

Inheritance diagram for anise.framework.report.ExecutionScope.ProgressIndeterminate:



Public Member Functions

- `def __init__ (self)`
- `def tojson (self)`
Returns a simple object structure for json transfers.

Public Attributes

- [kind](#)
- [stopped](#)

1.283.1 Detailed Description

[Progress](#) info for an indeterminate progress.

1.283.2 Constructor & Destructor Documentation

1.283.2.1 `__init__()`

```
def anise.framework.report.ExecutionScope.ProgressIndeterminate.__init__ (
    self )
```

1.283.3 Member Function Documentation

1.283.3.1 `tojson()`

```
def anise.framework.report.ExecutionScope.Progress.tojson (
    self ) [inherited]
```

Returns a simple object structure for json transfers.

1.283.4 Member Data Documentation

1.283.4.1 kind

`anise.framework.report.ExecutionScope.ProgressIndeterminate.kind`

1.283.4.2 stopped

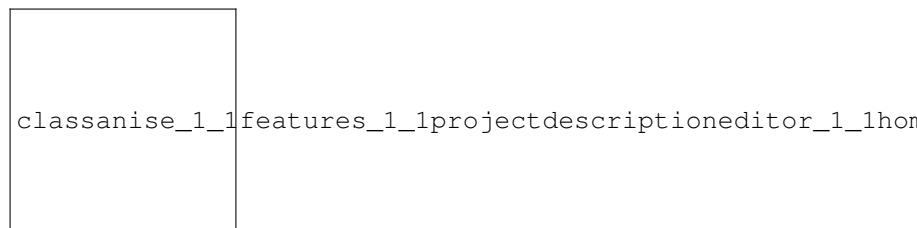
`anise.framework.report.ExecutionScope.ProgressIndeterminate.stopped`

The documentation for this class was generated from the following file:

- `anise/framework/report.py`

1.284 anise.features.projectdescriptioneditor.homepage.ProjectDescriptionEditorAssistant Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.homepage.ProjectDescriptionEditorAssistant`:



Public Member Functions

- `def __init__(self)`
- `def getcustomactions(self, ais)`
- `def getentries(self, ais)`
- `def getentrylabel(self, ais, entry)`
- `def execute(self)`
- `def manageentry(self, ais, node)`
Called for managing one of the entries.
- `def visible(self)`
Checks if this feature action shall be visible.

Public Attributes

- [header](#)
- [label](#)
- [description](#)
- [forfeature](#)

Private Member Functions

- [def _add_homepage_destination](#) (self, ais)

1.284.1 Constructor & Destructor Documentation

1.284.1.1 __init__()

```
def anise.features.projectdescriptioneditor.homepage.ProjectDescriptionEditorAssistant.__↵↵
init__ (
    self )
```

1.284.2 Member Function Documentation

1.284.2.1 _add_homepage_destination()

```
def anise.features.projectdescriptioneditor.homepage.ProjectDescriptionEditorAssistant._add_↵↵
homepage_destination (
    self,
    ais ) [private]
```

1.284.2.2 execute()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (
    self ) [inherited]
```

1.284.2.3 getcustomactions()

```
def anise.features.projectdescriptioneditor.homepage.ProjectDescriptionEditorAssistant.↵↵
getcustomactions (
    self,
    ais )
```

1.284.2.4 getentries()

```
def anise.features.projectdescriptioneditor.homepage.ProjectDescriptionEditorAssistant.↵  
getentries (   
    self,   
    ais )
```

1.284.2.5 getentrylabel()

```
def anise.features.projectdescriptioneditor.homepage.ProjectDescriptionEditorAssistant.↵  
getentrylabel (   
    self,   
    ais,   
    entry )
```

1.284.2.6 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (   
    self,   
    ais,   
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.284.2.7 visible()

```
def anise.features.diagnostics.FeatureAction.visible (   
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.284.3 Member Data Documentation

1.284.3.1 description

`anise.features.diagnostics.FeatureAction.description` [inherited]

1.284.3.2 forfeature

`anise.features.diagnostics.FeatureAction.forfeature` [inherited]

1.284.3.3 header

`anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header` [inherited]

1.284.3.4 label

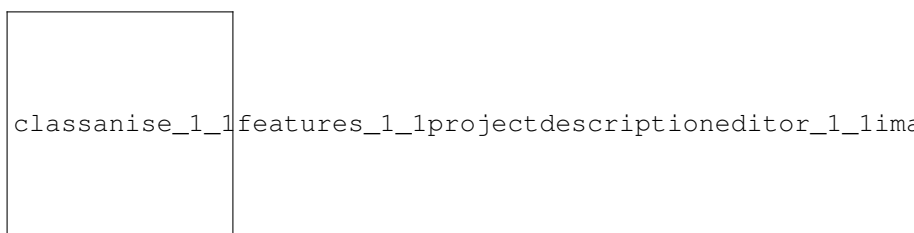
`anise.features.diagnostics.FeatureAction.label` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/homepage.py](#)

1.285 `anise.features.projectdescriptioneditor.imagegalleries.ProjectDescriptionEditorAssistant` Class Reference ↩

Inheritance diagram for `anise.features.projectdescriptioneditor.imagegalleries.ProjectDescriptionEditorAssistant`:



Public Member Functions

- `def __init__ (self)`
- `def getcustomactions (self, ais)`
- `def getentries (self, ais)`
- `def getentrylabel (self, ais, entry)`
- `def execute (self)`
- `def manageentry (self, ais, node)`
Called for managing one of the entries.
- `def visible (self)`
Checks if this feature action shall be visible.

Public Attributes

- [header](#)
- [label](#)
- [description](#)
- [forfeature](#)

Private Member Functions

- [def _add_imagegallery](#) (self, ais)

1.285.1 Constructor & Destructor Documentation

1.285.1.1 __init__()

```
def anise.features.projectdescriptioneditor.imagegalleries.ProjectDescriptionEditorAssistant.↔  
__init__ (   
            self )
```

1.285.2 Member Function Documentation

1.285.2.1 _add_imagegallery()

```
def anise.features.projectdescriptioneditor.imagegalleries.ProjectDescriptionEditorAssistant.↔  
_add_imagegallery (   
            self,  
            ais ) [private]
```

1.285.2.2 execute()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (   
            self ) [inherited]
```

1.285.2.3 getcustomactions()

```
def anise.features.projectdescriptioneditor.imagegalleries.ProjectDescriptionEditorAssistant.↔  
getcustomactions (   
            self,  
            ais )
```

1.285.2.4 getentries()

```
def anise.features.projectdescriptioneditor.imagegalleries.ProjectDescriptionEditorAssistant.↔
getentries (
    self,
    ais )
```

1.285.2.5 getentrylabel()

```
def anise.features.projectdescriptioneditor.imagegalleries.ProjectDescriptionEditorAssistant.↔
getentrylabel (
    self,
    ais,
    entry )
```

1.285.2.6 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (
    self,
    ais,
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.285.2.7 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.285.3 Member Data Documentation

1.285.3.1 description

`anise.features.diagnostics.FeatureAction.description` [inherited]

1.285.3.2 forfeature

`anise.features.diagnostics.FeatureAction.forfeature` [inherited]

1.285.3.3 header

`anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header` [inherited]

1.285.3.4 label

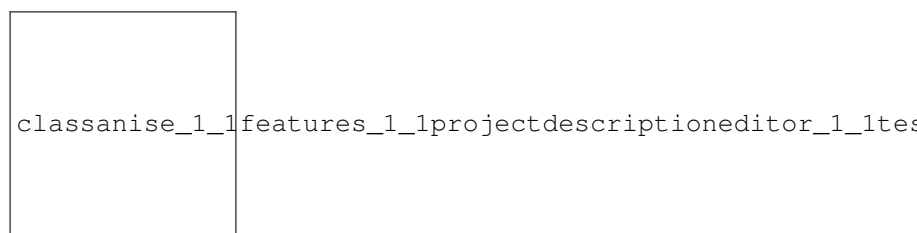
`anise.features.diagnostics.FeatureAction.label` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/imagegalleries.py](#)

1.286 anise.features.projectdescriptioneditor.testing.ProjectDescriptionEditorAssistant Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.testing.ProjectDescriptionEditorAssistant`:



Public Member Functions

- `def __init__ (self)`
- `def getcustomactions (self, ais)`
- `def getentries (self, ais)`
- `def getentrylabel (self, ais, entry)`
- `def execute (self)`
- `def manageentry (self, ais, node)`
Called for managing one of the entries.
- `def visible (self)`
Checks if this feature action shall be visible.

Public Attributes

- [header](#)
- [label](#)
- [description](#)
- [forfeature](#)

Private Member Functions

- [def _add_testpackage](#) (self, ais)

1.286.1 Constructor & Destructor Documentation

1.286.1.1 __init__()

```
def anise.features.projectdescriptioneditor.testing.ProjectDescriptionEditorAssistant.__init__↵
__ (
    self )
```

1.286.2 Member Function Documentation

1.286.2.1 _add_testpackage()

```
def anise.features.projectdescriptioneditor.testing.ProjectDescriptionEditorAssistant._add_↵
testpackage (
    self,
    ais ) [private]
```

1.286.2.2 execute()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (
    self ) [inherited]
```

1.286.2.3 getcustomactions()

```
def anise.features.projectdescriptioneditor.testing.ProjectDescriptionEditorAssistant.getcustomactions
(
    self,
    ais )
```

1.286.2.4 getentries()

```
def anise.features.projectdescriptioneditor.testing.ProjectDescriptionEditorAssistant.getentries
(
    self,
    ais )
```

1.286.2.5 getentrylabel()

```
def anise.features.projectdescriptioneditor.testing.ProjectDescriptionEditorAssistant.getentrylabel
(
    self,
    ais,
    entry )
```

1.286.2.6 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (
    self,
    ais,
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.286.2.7 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.286.3 Member Data Documentation

1.286.3.1 description

```
anise.features.diagnostics.FeatureAction.description [inherited]
```

1.286.3.2 forfeature

```
anise.features.diagnostics.FeatureAction.forfeature [inherited]
```

1.286.3.3 header

```
anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header [inherited]
```

1.286.3.4 label

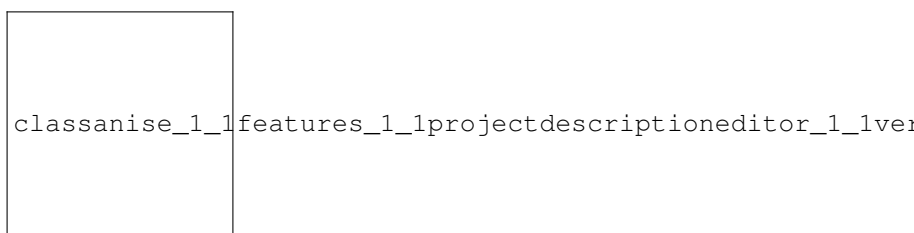
```
anise.features.diagnostics.FeatureAction.label [inherited]
```

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/testing.py](#)

1.287 **anise.features.projectdescriptioneditor.versioncontrol.ProjectDescriptionEditorAssistant Class Reference** ↩

Inheritance diagram for `anise.features.projectdescriptioneditor.versioncontrol.ProjectDescriptionEditorAssistant`:

**Public Member Functions**

- `def __init__ (self)`
- `def getcustomactions (self, ais)`
- `def getentries (self, ais)`
- `def getentrylabel (self, ais, entry)`
- `def execute (self)`
- `def manageentry (self, ais, node)`
Called for managing one of the entries.
- `def visible (self)`
Checks if this feature action shall be visible.

Public Attributes

- [header](#)
- [label](#)
- [description](#)
- [forfeature](#)

Private Member Functions

- [def _add_versioncontrolsystem](#) (self, ais)

1.287.1 Constructor & Destructor Documentation

1.287.1.1 __init__()

```
def anise.features.projectdescriptioneditor.versioncontrol.ProjectDescriptionEditorAssistant.↔  
__init__ (   
            self )
```

1.287.2 Member Function Documentation

1.287.2.1 _add_versioncontrolsystem()

```
def anise.features.projectdescriptioneditor.versioncontrol.ProjectDescriptionEditorAssistant.↔  
_add_versioncontrolsystem (   
            self,  
            ais ) [private]
```

1.287.2.2 execute()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (   
            self ) [inherited]
```

1.287.2.3 getcustomactions()

```
def anise.features.projectdescriptioneditor.versioncontrol.ProjectDescriptionEditorAssistant.↔  
getcustomactions (   
            self,  
            ais )
```

1.287.2.4 getentries()

```
def anise.features.projectdescriptioneditor.versioncontrol.ProjectDescriptionEditorAssistant.↔
getentries (
    self,
    ais )
```

1.287.2.5 getentrylabel()

```
def anise.features.projectdescriptioneditor.versioncontrol.ProjectDescriptionEditorAssistant.↔
getentrylabel (
    self,
    ais,
    entry )
```

1.287.2.6 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (
    self,
    ais,
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.287.2.7 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.287.3 Member Data Documentation

1.287.3.1 description

`anise.features.diagnostics.FeatureAction.description` [inherited]

1.287.3.2 forfeature

`anise.features.diagnostics.FeatureAction.forfeature` [inherited]

1.287.3.3 header

`anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header` [inherited]

1.287.3.4 label

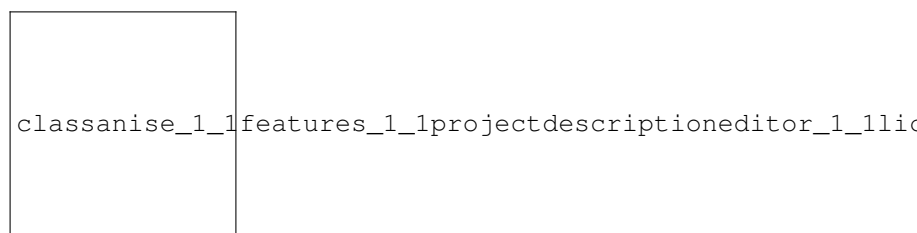
`anise.features.diagnostics.FeatureAction.label` [inherited]

The documentation for this class was generated from the following file:

- `anise/features/projectdescriptioneditor/versioncontrol.py`

1.288 anise.features.projectdescriptioneditor.licensing.ProjectDescriptionEditorAssistant Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.licensing.ProjectDescriptionEditorAssistant`:



Public Member Functions

- `def __init__ (self)`
- `def getcustomactions (self, ais)`
- `def getentries (self, ais)`
- `def getentrylabel (self, ais, entry)`
- `def execute (self)`
- `def manageentry (self, ais, node)`
Called for managing one of the entries.
- `def visible (self)`
Checks if this feature action shall be visible.

Public Attributes

- [header](#)
- [label](#)
- [description](#)
- [forfeature](#)

Private Member Functions

- [def _set_license](#) (self, ais)

1.288.1 Constructor & Destructor Documentation

1.288.1.1 __init__()

```
def anise.features.projectdescriptioneditor.licensing.ProjectDescriptionEditorAssistant.__↵  
init__ (   
    self )
```

1.288.2 Member Function Documentation

1.288.2.1 _set_license()

```
def anise.features.projectdescriptioneditor.licensing.ProjectDescriptionEditorAssistant._set↵  
_license (   
    self,  
    ais ) [private]
```

1.288.2.2 execute()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (   
    self ) [inherited]
```

1.288.2.3 getcustomactions()

```
def anise.features.projectdescriptioneditor.licensing.ProjectDescriptionEditorAssistant.↵  
getcustomactions (   
    self,  
    ais )
```


1.288.2.4 getentries()

```
def anise.features.projectdescriptioneditor.licensing.ProjectDescriptionEditorAssistant.↔  
getentries (   
    self,  
    ais )
```

1.288.2.5 getentrylabel()

```
def anise.features.projectdescriptioneditor.licensing.ProjectDescriptionEditorAssistant.↔  
getentrylabel (   
    self,  
    ais,  
    entry )
```

1.288.2.6 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (   
    self,  
    ais,  
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.288.2.7 visible()

```
def anise.features.diagnostics.FeatureAction.visible (   
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.288.3 Member Data Documentation

1.288.3.1 description

```
anise.features.diagnostics.FeatureAction.description [inherited]
```

1.288.3.2 forfeature

```
anise.features.diagnostics.FeatureAction.forfeature [inherited]
```

1.288.3.3 header

```
anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header [inherited]
```

1.288.3.4 label

```
anise.features.diagnostics.FeatureAction.label [inherited]
```

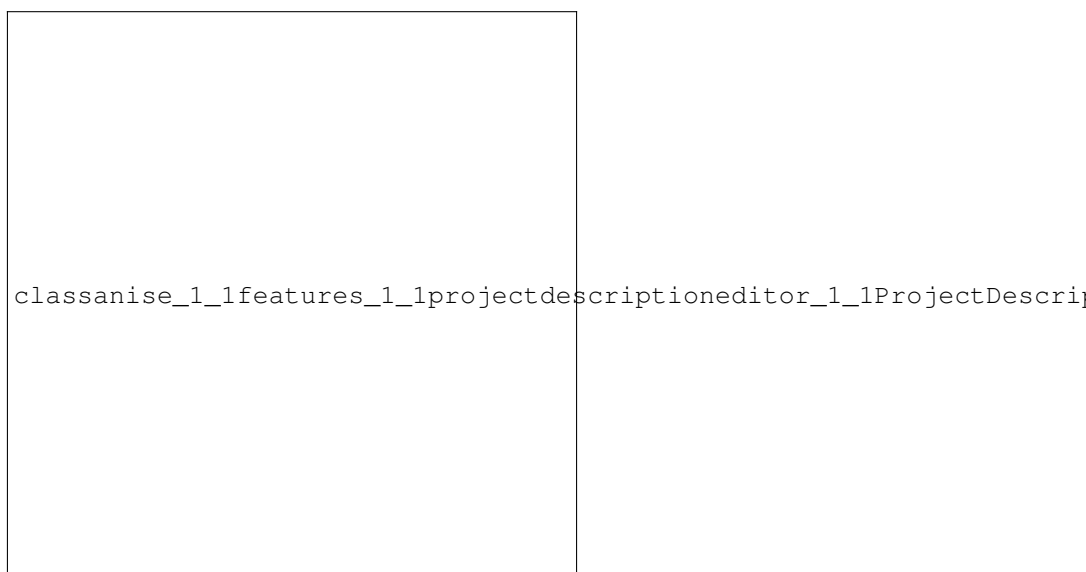
The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/licensing.py](#)

1.289 anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant Class Reference

Interface class for a project description editor assistant.

Inheritance diagram for `anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant`:



Public Member Functions

- def `__init__` (self, `label`, `description`, `forfeature`="", `header`=None)
 - def `execute` (self)
 - def `getcustomactions` (self, ais)
 - Returns a list of `CustomAction`, controlling which additional actions are offered in the overview.
 - def `getentries` (self, ais)
 - Returns a list of nodes to show in the overview.
 - def `getentrylabel` (self, ais, node)
 - Returns a label text for a node.
 - def `manageentry` (self, ais, node)
 - Called for managing one of the entries.
 - def `visible` (self)
 - Checks if this feature action shall be visible.

Public Attributes

- `header`
- `label`
- `description`
- `forfeature`

1.289.1 Detailed Description

Interface class for a project description editor assistant.

Represents one entry in the main list.

1.289.2 Constructor & Destructor Documentation

1.289.2.1 `__init__()`

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.__init__ (
    self,
    label,
    description,
    forfeature = "",
    header = None )
```

Parameters

<code>label</code>	The action label text.
<code>description</code>	The action description text.
<code>forfeature</code>	Full name of the feature to associate this action with (e.g. 'foo.bar').
<code>header</code>	Header text for the overview screen.

1.289.3 Member Function Documentation

1.289.3.1 execute()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (
    self )
```

1.289.3.2 getcustomactions()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.getcustomactions
(
    self,
    ais )
```

Returns a list of [CustomAction](#), controlling which additional actions are offered in the overview.

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
------------	---

1.289.3.3 getentries()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.getentries (
    self,
    ais )
```

Returns a list of nodes to show in the overview.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
------------	---

1.289.3.4 getentrylabel()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.getentrylabel (
    self,
```

```
        ais,  
        node )
```

Returns a label text for a node.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.289.3.5 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (  
    self,  
    ais,  
    node )
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.289.3.6 visible()

```
def anise.features.diagnostics.FeatureAction.visible (  
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.289.4 Member Data Documentation

1.289.4.1 description

`anise.features.diagnostics.FeatureAction.description` [inherited]

1.289.4.2 forfeature

`anise.features.diagnostics.FeatureAction.forfeature` [inherited]

1.289.4.3 header

`anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header`

1.289.4.4 label

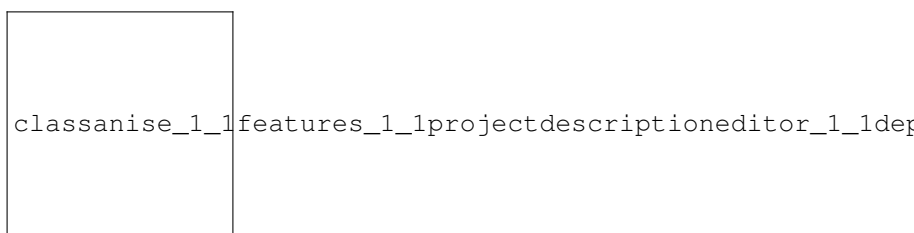
`anise.features.diagnostics.FeatureAction.label` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor.py](#)

1.290 `anise.features.projectdescriptioneditor.dependencies.ProjectDescriptionEditorAssistant` Class Reference ↩

Inheritance diagram for `anise.features.projectdescriptioneditor.dependencies.ProjectDescriptionEditorAssistant`:



Public Member Functions

- `def __init__ (self)`
- `def getcustomactions (self, ais)`
- `def getentries (self, ais)`
- `def getentrylabel (self, ais, entry)`
- `def execute (self)`
- `def manageentry (self, ais, node)`
Called for managing one of the entries.
- `def visible (self)`
Checks if this feature action shall be visible.

Public Attributes

- [header](#)
- [label](#)
- [description](#)
- [forfeature](#)

Private Member Functions

- [def _add_dependency](#) (self, ais)

1.290.1 Constructor & Destructor Documentation

1.290.1.1 __init__()

```
def anise.features.projectdescriptioneditor.dependencies.ProjectDescriptionEditorAssistant.__↵
__init__ (
    self )
```

1.290.2 Member Function Documentation

1.290.2.1 _add_dependency()

```
def anise.features.projectdescriptioneditor.dependencies.ProjectDescriptionEditorAssistant.__↵
add_dependency (
    self,
    ais ) [private]
```

1.290.2.2 execute()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (
    self ) [inherited]
```

1.290.2.3 getcustomactions()

```
def anise.features.projectdescriptioneditor.dependencies.ProjectDescriptionEditorAssistant.↵
getcustomactions (
    self,
    ais )
```

1.290.2.4 getentries()

```
def anise.features.projectdescriptioneditor.dependencies.ProjectDescriptionEditorAssistant.↵
getentries (
    self,
    ais )
```

1.290.2.5 getentrylabel()

```
def anise.features.projectdescriptioneditor.dependencies.ProjectDescriptionEditorAssistant.↵
getentrylabel (
    self,
    ais,
    entry )
```

1.290.2.6 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (
    self,
    ais,
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.290.2.7 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.290.3 Member Data Documentation

1.290.3.1 description

`anise.features.diagnostics.FeatureAction.description` [inherited]

1.290.3.2 forfeature

`anise.features.diagnostics.FeatureAction.forfeature` [inherited]

1.290.3.3 header

`anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header` [inherited]

1.290.3.4 label

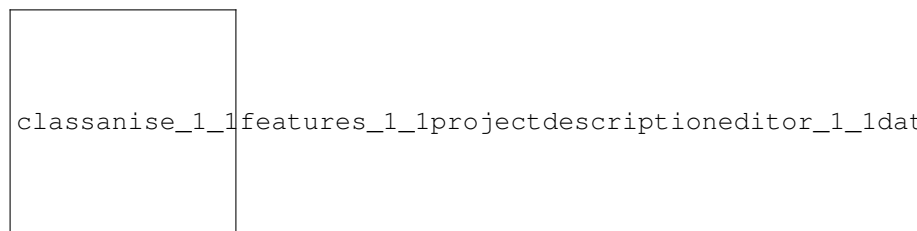
`anise.features.diagnostics.FeatureAction.label` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/dependencies.py](#)

1.291 anise.features.projectdescriptioneditor.datainjections.ProjectDescriptionEditorAssistant Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.datainjections.ProjectDescriptionEditorAssistant`:



Public Member Functions

- `def __init__ (self)`
- `def getcustomactions (self, ais)`
- `def getentries (self, ais)`
- `def getentrylabel (self, ais, entry)`
- `def execute (self)`
- `def manageentry (self, ais, node)`
Called for managing one of the entries.
- `def visible (self)`
Checks if this feature action shall be visible.

Public Attributes

- [header](#)
- [label](#)
- [description](#)
- [forfeature](#)

Private Member Functions

- [def _add_datainjection](#) (self, ais)

1.291.1 Constructor & Destructor Documentation

1.291.1.1 __init__()

```
def anise.features.projectdescriptioneditor.datainjections.ProjectDescriptionEditorAssistant.  
__init__ (   
    self )
```

1.291.2 Member Function Documentation

1.291.2.1 _add_datainjection()

```
def anise.features.projectdescriptioneditor.datainjections.ProjectDescriptionEditorAssistant.  
_add_datainjection (   
    self,  
    ais ) [private]
```

1.291.2.2 execute()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (   
    self ) [inherited]
```

1.291.2.3 getcustomactions()

```
def anise.features.projectdescriptioneditor.datainjections.ProjectDescriptionEditorAssistant.  
getcustomactions (   
    self,  
    ais )
```

1.291.2.4 getentries()

```
def anise.features.projectdescriptioneditor.datainjections.ProjectDescriptionEditorAssistant.↵
getentries (
    self,
    ais )
```

1.291.2.5 getentrylabel()

```
def anise.features.projectdescriptioneditor.datainjections.ProjectDescriptionEditorAssistant.↵
getentrylabel (
    self,
    ais,
    entry )
```

1.291.2.6 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (
    self,
    ais,
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.291.2.7 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.291.3 Member Data Documentation

1.291.3.1 description

`anise.features.diagnostics.FeatureAction.description` [inherited]

1.291.3.2 forfeature

`anise.features.diagnostics.FeatureAction.forfeature` [inherited]

1.291.3.3 header

`anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header` [inherited]

1.291.3.4 label

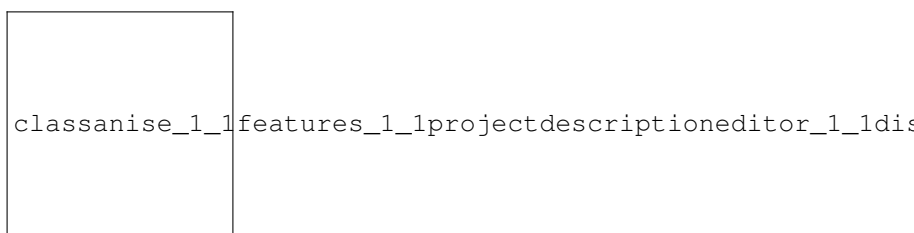
`anise.features.diagnostics.FeatureAction.label` [inherited]

The documentation for this class was generated from the following file:

- `anise/features/projectdescriptioneditor/datainjections.py`

1.292 `anise.features.projectdescriptioneditor.distributables.ProjectDescriptionEditorAssistant` Class Reference ↩

Inheritance diagram for `anise.features.projectdescriptioneditor.distributables.ProjectDescriptionEditorAssistant`:



Public Member Functions

- `def __init__ (self)`
- `def getcustomactions (self, ais)`
- `def getentries (self, ais)`
- `def getentrylabel (self, ais, entry)`
- `def execute (self)`
- `def manageentry (self, ais, node)`
Called for managing one of the entries.
- `def visible (self)`
Checks if this feature action shall be visible.

Public Attributes

- [header](#)
- [label](#)
- [description](#)
- [forfeature](#)

Private Member Functions

- [def _add_distributable](#) (self, ais)

1.292.1 Constructor & Destructor Documentation

1.292.1.1 __init__()

```
def anise.features.projectdescriptioneditor.distributables.ProjectDescriptionEditorAssistant.  
__init__ (   
    self )
```

1.292.2 Member Function Documentation

1.292.2.1 _add_distributable()

```
def anise.features.projectdescriptioneditor.distributables.ProjectDescriptionEditorAssistant.  
_add_distributable (   
    self,  
    ais ) [private]
```

1.292.2.2 execute()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (   
    self ) [inherited]
```

1.292.2.3 getcustomactions()

```
def anise.features.projectdescriptioneditor.distributables.ProjectDescriptionEditorAssistant.  
getcustomactions (   
    self,  
    ais )
```

1.292.2.4 getentries()

```
def anise.features.projectdescriptioneditor.distributables.ProjectDescriptionEditorAssistant.↔
getentries (
    self,
    ais )
```

1.292.2.5 getentrylabel()

```
def anise.features.projectdescriptioneditor.distributables.ProjectDescriptionEditorAssistant.↔
getentrylabel (
    self,
    ais,
    entry )
```

1.292.2.6 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (
    self,
    ais,
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.292.2.7 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.292.3 Member Data Documentation

1.292.3.1 description

`anise.features.diagnostics.FeatureAction.description` [inherited]

1.292.3.2 forfeature

`anise.features.diagnostics.FeatureAction.forfeature` [inherited]

1.292.3.3 header

`anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header` [inherited]

1.292.3.4 label

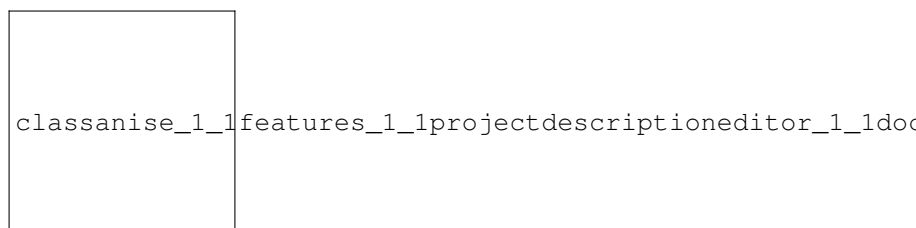
`anise.features.diagnostics.FeatureAction.label` [inherited]

The documentation for this class was generated from the following file:

- `anise/features/projectdescriptioneditor/distributables.py`

1.293 anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistantDevdoc Class Reference↩

Inheritance diagram for `anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistantDevdoc`:↩



Public Member Functions

- def `__init__` (self)
- def `getcustomactions` (self, ais)
- def `execute` (self)
- def `getentries` (self, ais)
Returns a list of nodes to show in the overview.
- def `getentrylabel` (self, ais, node)
Returns a label text for a node.
- def `manageentry` (self, ais, node)
Called for managing one of the entries.
- def `visible` (self)
Checks if this feature action shall be visible.

Public Attributes

- `header`
- `label`
- `description`
- `forfeature`

Private Member Functions

- def `_disable_devdoc` (self, ais)
- def `_enable_devdoc` (self, ais)

1.293.1 Constructor & Destructor Documentation

1.293.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant←
Devdoc.__init__ (
    self )
```

1.293.2 Member Function Documentation

1.293.2.1 `_disable_devdoc()`

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant←
Devdoc._disable_devdoc (
    self,
    ais ) [private]
```


1.293.2.2 _enable_devdoc()

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant←  
Devdoc._enable_devdoc (   
    self,   
    ais ) [private]
```

1.293.2.3 execute()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (   
    self ) [inherited]
```

1.293.2.4 getcustomactions()

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant←  
Devdoc.getcustomactions (   
    self,   
    ais )
```

1.293.2.5 getentries()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.getentries (   
    self,   
    ais ) [inherited]
```

Returns a list of nodes to show in the overview.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
------------	---

1.293.2.6 getentrylabel()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.getentrylabel (   
    self,
```

```
        ais,  
        node ) [inherited]
```

Returns a label text for a node.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.293.2.7 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (  
    self,  
    ais,  
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.293.2.8 visible()

```
def anise.features.diagnostics.FeatureAction.visible (  
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.293.3 Member Data Documentation

1.293.3.1 description

`anise.features.diagnostics.FeatureAction.description` [inherited]

1.293.3.2 forfeature

`anise.features.diagnostics.FeatureAction.forfeature` [inherited]

1.293.3.3 header

`anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header` [inherited]

1.293.3.4 label

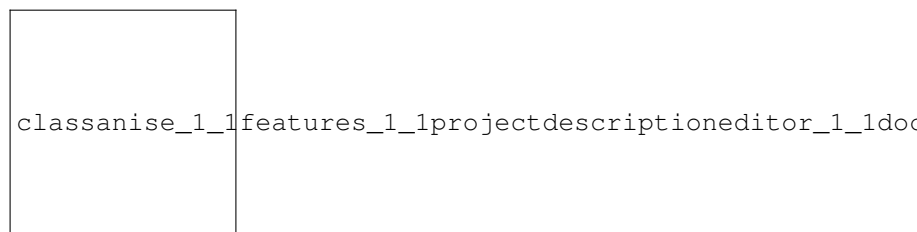
`anise.features.diagnostics.FeatureAction.label` [inherited]

The documentation for this class was generated from the following file:

- `anise/features/projectdescriptioneditor/documentation.py`

1.294 anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistantExports Class Reference ↩

Inheritance diagram for `anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistantExports`: ↩



Public Member Functions

- `def __init__ (self)`
- `def getcustomactions (self, ais)`
- `def getentries (self, ais)`
- `def getentrylabel (self, ais, entry)`
- `def execute (self)`
- `def manageentry (self, ais, node)`
Called for managing one of the entries.
- `def visible (self)`
Checks if this feature action shall be visible.

Public Attributes

- [header](#)
- [label](#)
- [description](#)
- [forfeature](#)

Private Member Functions

- [def _add_export](#) (self, ais)

1.294.1 Constructor & Destructor Documentation

1.294.1.1 __init__()

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant↵
Exports.__init__ (
    self )
```

1.294.2 Member Function Documentation

1.294.2.1 _add_export()

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant↵
Exports._add_export (
    self,
    ais ) [private]
```

1.294.2.2 execute()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (
    self ) [inherited]
```

1.294.2.3 getcustomactions()

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant↵
Exports.getcustomactions (
    self,
    ais )
```

1.294.2.4 getentries()

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant↵
Exports.getentries (
    self,
    ais )
```

1.294.2.5 getentrylabel()

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant↵
Exports.getentrylabel (
    self,
    ais,
    entry )
```

1.294.2.6 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (
    self,
    ais,
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.294.2.7 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.294.3 Member Data Documentation

1.294.3.1 description

```
anise.features.diagnostics.FeatureAction.description [inherited]
```

1.294.3.2 forfeature

```
anise.features.diagnostics.FeatureAction.forfeature [inherited]
```

1.294.3.3 header

```
anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header [inherited]
```

1.294.3.4 label

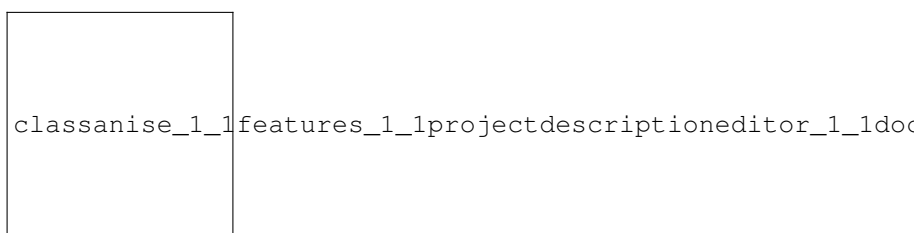
```
anise.features.diagnostics.FeatureAction.label [inherited]
```

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/documentation.py](#)

1.295 **anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistantPool Class Reference**↔

Inheritance diagram for `anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistantPool`:↔

**Public Member Functions**

- def [__init__](#) (self)
- def [getcustomactions](#) (self, ais)
- def [getentries](#) (self, ais)
- def [getentrylabel](#) (self, ais, entry)
- def [execute](#) (self)
- def [manageentry](#) (self, ais, node)
 - Called for managing one of the entries.*
- def [visible](#) (self)
 - Checks if this feature action shall be visible.*

Public Attributes

- [header](#)
- [label](#)
- [description](#)
- [forfeature](#)

Private Member Functions

- [def _add_pool](#) (self, ais)

1.295.1 Constructor & Destructor Documentation

1.295.1.1 __init__()

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant↵
Pool.__init__ (
    self )
```

1.295.2 Member Function Documentation

1.295.2.1 _add_pool()

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant↵
Pool._add_pool (
    self,
    ais ) [private]
```

1.295.2.2 execute()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (
    self ) [inherited]
```

1.295.2.3 getcustomactions()

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant↵
Pool.getcustomactions (
    self,
    ais )
```

1.295.2.4 getentries()

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant←
Pool.getentries (
    self,
    ais )
```

1.295.2.5 getentrylabel()

```
def anise.features.projectdescriptioneditor.documentation.ProjectDescriptionEditorAssistant←
Pool.getentrylabel (
    self,
    ais,
    entry )
```

1.295.2.6 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (
    self,
    ais,
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.295.2.7 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.295.3 Member Data Documentation

1.295.3.1 description

`anise.features.diagnostics.FeatureAction.description` [inherited]

1.295.3.2 forfeature

`anise.features.diagnostics.FeatureAction.forfeature` [inherited]

1.295.3.3 header

`anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header` [inherited]

1.295.3.4 label

`anise.features.diagnostics.FeatureAction.label` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/documentation.py](#)

1.296 anise.framework.features.PseudoFeature Class Reference

This is an auxiliary class, which fixes the gap which appears, e.g.

1.296.1 Detailed Description

This is an auxiliary class, which fixes the gap which appears, e.g.

if a feature 'foo.bar' exists, but no 'foo' one.

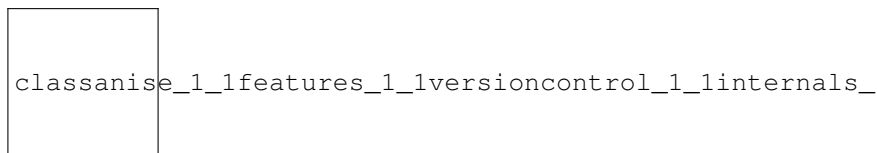
The documentation for this class was generated from the following file:

- [anise/framework/features.py](#)

1.297 anise.features.versioncontrol.internals.PseudoVersionControlSystem Class Reference

A pseudo version control system implementation which does nothing.

Inheritance diagram for anise.features.versioncontrol.internals.PseudoVersionControlSystem:



Public Member Functions

- def `__init__` (self)
- def `syncversioncontrolsystem` (self, commitmessage="", forceskipped=True, forceunchanged=False)
- def `getfullversion` (self)
Returns a complete version string.
- def `createrawpackage` (self)
Generates a raw package.
- def `findvcs` (self)
Checks if the current project is bound to a version control system of this kind.
- def `bindvcs` (self, l)
Binds the current project to this version control system.
- def `fetchrawcommitmessage` (self, revision)
Fetches a raw commit message.
- def `getcommitmessagefromrawcommitmessage` (self, s)
Gets the actual commit message from the raw 'fetchrawcommitmessage' commit message output.
- def `getpreviousrevision` (self, revision)
Gets the predecessor revision for a given revision.
- def `getheadrevision` (self)
Gets the head revision.
- def `getcommitmessage` (self, revision)
Returns the raw commit message (as formatted by the vcs tool) for a revision.
- def `getchangelog` (self, asstring=True)
Computes the project change log from information in specially formatted vcs commit messages.
- def `addchange` (self, changetext=None)
Adds a change to the changelog.
- def `addlabel` (self, label=None)
Adds a version label for the current state.
- def `storechange` (self, changetext)
Stores a new entry to the changelog.
- def `storelabel` (self, label)
Stores a new version label.
- def `branchfromhere` (self, branchname=None)
Creates a new branch from here and switches to it.

Public Attributes

- [label](#)

1.297.1 Detailed Description

A pseudo version control system implementation which does nothing.

This is the default as long as no real version control system is configured.

1.297.2 Constructor & Destructor Documentation

1.297.2.1 `__init__()`

```
def anise.features.versioncontrol.internals.PseudoVersionControlSystem.__init__ (
    self )
```

1.297.3 Member Function Documentation

1.297.3.1 `addchange()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.addchange (
    self,
    changetext = None ) [inherited]
```

Adds a change to the changelog.

Parameters

<i>changetext</i>	The changelog entry text.
-------------------	---------------------------

1.297.3.2 `addlabel()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.addlabel (
    self,
    label = None ) [inherited]
```

Adds a version label for the current state.

Parameters

<i>label</i>	The label string.
--------------	-------------------

1.297.3.3 bindvcs()

```
def anise.features.versioncontrol.internals.VersionControlSystem.bindvcs (
    self,
    l ) [inherited]
```

Binds the current project to this version control system.

Uses user interaction for getting details like urls.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>l</i>	The let configuration node for storing additional parameters.
----------	---

1.297.3.4 branchfromhere()

```
def anise.features.versioncontrol.internals.VersionControlSystem.branchfromhere (
    self,
    branchname = None ) [inherited]
```

Creates a new branch from here and switches to it.

Unpushed changes will become part of that new branch.

Override this method in custom subclasses.

Parameters

<i>branchname</i>	The branch name (None: input dialog).
-------------------	---------------------------------------

1.297.3.5 createrawpackage()

```
def anise.features.versioncontrol.internals.VersionControlSystem.createrawpackage (
    self ) [inherited]
```

Generates a raw package.

This package is the basis for most (or all) of the other package types.

Override this method in custom subclasses.

1.297.3.6 fetchrawcommitmessage()

```
def anise.features.versioncontrol.internals.VersionControlSystem.fetchrawcommitmessage (
    self,
    revision ) [inherited]
```

Fetches a raw commit message.

Do not use it directly, but getcommitmessage instead.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>revision</i>	The revision.
-----------------	---------------

Returns

: The raw commit message as string in a custom format.

1.297.3.7 findvcs()

```
def anise.features.versioncontrol.internals.VersionControlSystem.findvcs (
    self ) [inherited]
```

Checks if the current project is bound to a version control system of this kind.

Override this method in custom subclasses or leave the default implementation.

1.297.3.8 getchangelog()

```
def anise.features.versioncontrol.internals.VersionControlSystem.getchangelog (
    self,
    asstring = True ) [inherited]
```

Computes the project change log from information in specially formatted vcs commit messages.

Parameters

<i>asstring</i>	If to output the changelog as string (instead of an internal data structure).
-----------------	---

1.297.3.9 `getcommitmessage()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.getcommitmessage (
    self,
    revision ) [inherited]
```

Returns the raw commit message (as formatted by the vcs tool) for a revision.

Parameters

<i>revision</i>	The revision.
-----------------	---------------

1.297.3.10 `getcommitmessagefromrawcommitmessage()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.getcommitmessagefromrawcommitmessage (
    self,
    s ) [inherited]
```

Gets the actual commit message from the raw 'fetchrawcommitmessage' commit message output.

Do not use it directly, but `getcommitmessage` instead.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>s</i>	The raw commit message string.
----------	--------------------------------

Returns

: An instance of [CommitMessage](#).

1.297.3.11 `getfullversion()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.getfullversion (
    self ) [inherited]
```

Returns a complete version string.

Override this method in custom subclasses.

1.297.3.12 getheadrevision()

```
def anise.features.versioncontrol.internals.VersionControlSystem.getheadrevision (
    self ) [inherited]
```

Gets the head revision.

Override this method in custom subclasses or leave the default implementation.

1.297.3.13 getpreviousrevision()

```
def anise.features.versioncontrol.internals.VersionControlSystem.getpreviousrevision (
    self,
    revision ) [inherited]
```

Gets the predecessor revision for a given revision.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>revision</i>	A revision.
-----------------	-------------

Returns

: The predeccessing revision.

1.297.3.14 storechange()

```
def anise.features.versioncontrol.internals.VersionControlSystem.storechange (
    self,
    changetext ) [inherited]
```

Stores a new entry to the changelog.

Do not use this directly, but addchange instead.

Parameters

<i>changetext</i>	The changelog entry text.
-------------------	---------------------------

1.297.3.15 storelabel()

```
def anise.features.versioncontrol.internals.VersionControlSystem.storelabel (
    self,
    label ) [inherited]
```

Stores a new version label.

Do not use this directly, but addlabel instead.

Parameters

<i>label</i>	The label string.
--------------	-------------------

1.297.3.16 syncversioncontrolsystem()

```
def anise.features.versioncontrol.internals.PseudoVersionControlSystem.syncversioncontrolsystem
(
    self,
    commitmessage = "",
    forceskipped = True,
    forceunchanged = False )
```

1.297.4 Member Data Documentation**1.297.4.1 label**

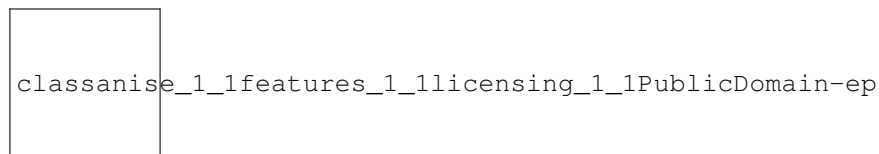
```
anise.features.versioncontrol.internals.VersionControlSystem.label [inherited]
```

The documentation for this class was generated from the following file:

- [anise/features/versioncontrol.py](#)

1.298 anise.features.licensing.PublicDomain Class Reference

Inheritance diagram for anise.features.licensing.PublicDomain:



Public Member Functions

- `def __init__ (self)`
- `def getfilename (self)`
Returns the full path of the license text file.
- `def getlicensetext (self)`
Returns the full license text.

Public Attributes

- [name](#)
- [namedebian](#)
- [namepythonsetuputils](#)
- [filename](#)

1.298.1 Constructor & Destructor Documentation

1.298.1.1 `__init__()`

```
def anise.features.licensing.PublicDomain.__init__ (  
    self )
```

1.298.2 Member Function Documentation

1.298.2.1 `getfilename()`

```
def anise.features.licensing.BaseLicense.getfilename (  
    self ) [inherited]
```

Returns the full path of the license text file.

1.298.2.2 getlicensetext()

```
def anise.features.licensing.BaseLicense.getlicensetext (
    self ) [inherited]
```

Returns the full license text.

1.298.3 Member Data Documentation**1.298.3.1 filename**

```
anise.features.licensing.BaseLicense.filename [inherited]
```

1.298.3.2 name

```
anise.features.licensing.BaseLicense.name [inherited]
```

1.298.3.3 namedebian

```
anise.features.licensing.BaseLicense.namedebian [inherited]
```

1.298.3.4 namepythonsetuputils

```
anise.features.licensing.BaseLicense.namepythonsetuputils [inherited]
```

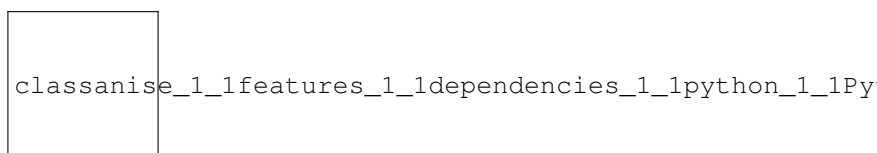
The documentation for this class was generated from the following file:

- [anise/features/licensing.py](#)

1.299 anise.features.dependencies.python.PythonDependency Class Reference

A dependency describing the required Python interpreter.

Inheritance diagram for `anise.features.dependencies.python.PythonDependency`:



Public Member Functions

- `def __init__` (self, [type](#), version, kwargs)

Public Attributes

- [deptype](#)
- [type](#)
- [objectname](#)
- [comment](#)
- [icon](#)
- [visible](#)
- [displayname](#)

1.299.1 Detailed Description

A dependency describing the required Python interpreter.

1.299.2 Constructor & Destructor Documentation

1.299.2.1 __init__()

```
def anise.features.dependencies.python.PythonDependency.__init__ (
    self,
    type,
    version,
    kwargs )
```

Parameters

<i>type</i>	The dependency type (see dependencies.Type).
<i>version</i>	The version of the required Python interpreter (as string).
<i>kwargs</i>	Additional dependency infos.

1.299.3 Member Data Documentation

1.299.3.1 comment

`anise.features.dependencies.Dependency.comment` [inherited]

1.299.3.2 deptype

`anise.features.dependencies.Dependency.deptype` [inherited]

1.299.3.3 displayname

`anise.features.dependencies.Dependency.displayname` [inherited]

1.299.3.4 icon

`anise.features.dependencies.Dependency.icon` [inherited]

1.299.3.5 objectname

`anise.features.dependencies.Dependency.objectname` [inherited]

1.299.3.6 type

`anise.features.dependencies.Dependency.type` [inherited]

1.299.3.7 visible

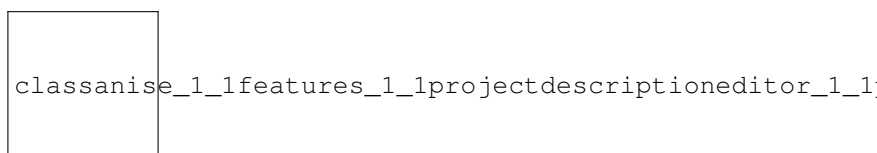
`anise.features.dependencies.Dependency.visible` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/dependencies/python.py](#)

1.300 anise.features.projectdescriptioneditor.packages.PythonWheelPackage Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.packages.PythonWheelPackage`:



Public Member Functions

- def [__init__](#) (self)
- def [ask](#) (self)
- def [gettaskref](#) (self)

Public Attributes

- [label](#)
- [taskref](#)

1.300.1 Constructor & Destructor Documentation

1.300.1.1 [__init__\(\)](#)

```
def anise.features.projectdescriptioneditor.packages.PythonWheelPackage.__init__ (  
    self )
```

1.300.2 Member Function Documentation

1.300.2.1 [ask\(\)](#)

```
def anise.features.projectdescriptioneditor.packages.PythonWheelPackage.ask (  
    self )
```

1.300.2.2 [gettaskref\(\)](#)

```
def anise.features.projectdescriptioneditor.packages.AbstractPackage.gettaskref (  
    self ) [inherited]
```

1.300.3 Member Data Documentation

1.300.3.1 [label](#)

```
anise.features.projectdescriptioneditor.packages.PythonWheelPackage.label
```

1.300.3.2 taskref

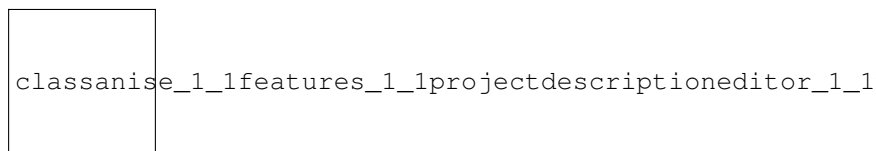
`anise.features.projectdescriptioneditor.packages.PythonWheelPackage.taskref`

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/packages.py](#)

1.301 `anise.features.projectdescriptioneditor.packages.PythonWheelPackageAddExecLinkCustomAction` Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.packages.PythonWheelPackageAddExecLinkCustomAction`:



Public Member Functions

- `def __init__ (self)`
- `def visible (self, ais, node)`
- `def execute (self, ais, node)`

Public Attributes

- `label`
- `function`
- `icon`

1.301.1 Constructor & Destructor Documentation

1.301.1.1 `__init__()`

```

def anise.features.projectdescriptioneditor.packages.PythonWheelPackageAddExecLinkCustom↵
Action.__init__ (
    self )

```

1.301.2 Member Function Documentation

1.301.2.1 execute()

```
def anise.features.projectdescriptioneditor.packages.PythonWheelPackageAddExecLinkCustom↵  
Action.execute (   
    self,   
    ais,   
    node )
```

1.301.2.2 visible()

```
def anise.features.projectdescriptioneditor.packages.PythonWheelPackageAddExecLinkCustom↵  
Action.visible (   
    self,   
    ais,   
    node )
```

1.301.3 Member Data Documentation

1.301.3.1 function

anise.features.projectdescriptioneditor.CustomAction.function [inherited]

1.301.3.2 icon

anise.features.projectdescriptioneditor.CustomAction.icon [inherited]

1.301.3.3 label

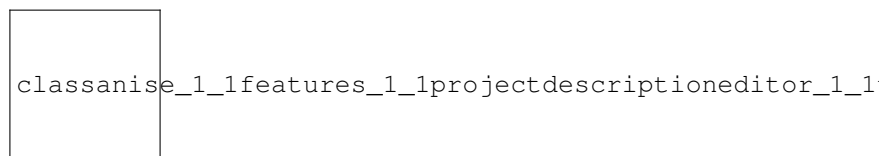
anise.features.projectdescriptioneditor.CustomAction.label [inherited]

The documentation for this class was generated from the following file:

- anise/features/projectdescriptioneditor/[packages.py](#)

1.302 anise.features.projectdescriptioneditor.testing.PyUnitTestingImplementation Class Reference

Inheritance diagram for anise.features.projectdescriptioneditor.testing.PyUnitTestingImplementation:



Public Member Functions

- def [__init__](#) (self)
- def [execute](#) (self, params)

Public Attributes

- [label](#)

1.302.1 Constructor & Destructor Documentation

1.302.1.1 [__init__](#)()

```
def anise.features.projectdescriptioneditor.testing.PyUnitTestingImplementation.__init__ (
    self )
```

1.302.2 Member Function Documentation

1.302.2.1 [execute](#)()

```
def anise.features.projectdescriptioneditor.testing.PyUnitTestingImplementation.execute (
    self,
    params )
```

1.302.3 Member Data Documentation

1.302.3.1 label

`anise.features.projectdescriptioneditor.testing.TestingImplementation.label` [inherited]

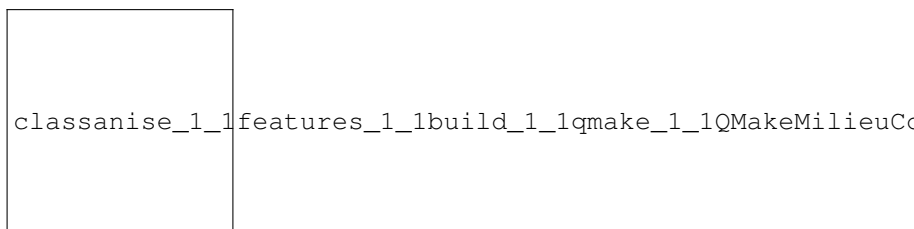
The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/testing.py](#)

1.303 anise.features.build.qmake.QMakeMilieuComposite Class Reference

Build environment composite for Qt and qmake builds.

Inheritance diagram for `anise.features.build.qmake.QMakeMilieuComposite`:



Public Member Functions

- `def __init__ (self, qtenv=None, qmakepath=None, bindir=None, pluginsdir=None, datadir=None, makepath="make")`
- `def initialize (self)`
- `def call_qmake (self, args, kwargs)`
Calls the qmake tool.
- `def call_make (self, args, kwargs)`
- `def make (self, targetname=None, makefile=None)`
- `def setinner (self, inner)`
Sets the inner milieu.
- `def __getattr__ (self, item)`

Public Attributes

- [qtenv](#)
- [qmakepath](#)
- [bindir](#)
- [pluginsdir](#)
- [datadir](#)
- [qmakecmd](#)
- [makepath](#)
- [processcount](#)
- [makecmd](#)
- [inner](#)

1.303.1 Detailed Description

Build environment composite for Qt and qmake builds.

See [milieus.MilieuComposite](#) for general infos.

1.303.2 Constructor & Destructor Documentation

1.303.2.1 `__init__()`

```
def anise.features.build.qmake.QMakeMilieuComposite.__init__ (
    self,
    qtenv = None,
    qmakepath = None,
    bindir = None,
    pluginsdir = None,
    datadir = None,
    makepath = "make" )
```

Parameters

<i>qtenv</i>	Optionally choose an qt environment by name (use <code>qtchooser -l</code> for a list).
<i>qmakepath</i>	The path to the Qt qmake tool.
<i>bindir</i>	The Qt bin directory.
<i>pluginsdir</i>	The Qt plugins directory.
<i>datadir</i>	The Qt data directory.
<i>makepath</i>	The path to the make tool.

1.303.3 Member Function Documentation

1.303.3.1 `__getattr__()`

```
def anise.features.milieus.MilieuComposite.__getattr__ (
    self,
    item ) [inherited]
```

1.303.3.2 `call_make()`

```
def anise.features.build.make.MakeMilieuComposite.call_make (
    self,
    args,
    kwargs ) [inherited]
```

1.303.3.3 call_qmake()

```
def anise.features.build.qmake.QMakeMilieuComposite.call_qmake (
    self,
    args,
    kwargs )
```

Calls the qmake tool.

1.303.3.4 initialize()

```
def anise.features.build.qmake.QMakeMilieuComposite.initialize (
    self )
```

1.303.3.5 make()

```
def anise.features.build.make.MakeMilieuComposite.make (
    self,
    targetname = None,
    makefile = None ) [inherited]
```

1.303.3.6 setinner()

```
def anise.features.milieus.MilieuComposite.setinner (
    self,
    inner ) [inherited]
```

Sets the inner milieu.

This is part of a particular piece of internal infrastructure and is typically not required to be used.

1.303.4 Member Data Documentation

1.303.4.1 bindir

```
anise.features.build.qmake.QMakeMilieuComposite.bindir
```

1.303.4.2 datadir

`anise.features.build.qmake.QMakeMilieuComposite.datadir`

1.303.4.3 inner

`anise.features.milieus.MilieuComposite.inner` [inherited]

1.303.4.4 makecmd

`anise.features.build.make.MakeMilieuComposite.makecmd` [inherited]

1.303.4.5 makepath

`anise.features.build.make.MakeMilieuComposite.makepath` [inherited]

1.303.4.6 pluginsdir

`anise.features.build.qmake.QMakeMilieuComposite.pluginsdir`

1.303.4.7 processcount

`anise.features.build.make.MakeMilieuComposite.processcount` [inherited]

1.303.4.8 qmakecmd

`anise.features.build.qmake.QMakeMilieuComposite.qmakecmd`

1.303.4.9 qmakepath

`anise.features.build.qmake.QMakeMilieuComposite.qmakepath`

1.303.4.10 qtenv

```
anise.features.build.qmake.QMakeMilieuComposite.qtenv
```

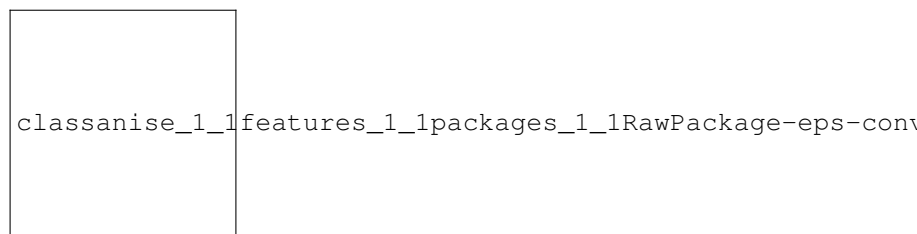
The documentation for this class was generated from the following file:

- [anise/features/build/qmake.py](#)

1.304 anise.features.packages.RawPackage Class Reference

A [anise.framework.files.Filestructure](#) subclass which contains the raw package, which is the typical starting point for any further packaging steps or for fetching parts of the project file structure for some processing.

Inheritance diagram for `anise.features.packages.RawPackage`:



Public Member Functions

- `def __init__ (self)`
- `def datakeys (self)`
Returns a list of keys for all stored metadata properties.
- `def setdata (self, k, v)`
Sets a metadata property.
- `def getdata (self, k, deflt)`
Gets a metadata property.
- `def initialize (self)`
Initializes this [Filestructure](#).
- `def path (self)`
The path to this structure in the filesystem.
- `def dl (self, subpath="", to=None, progresscallback=None)`
Copies the complete filestructure or a subdirectory to a new destination.
- `def with_modified_rootname (self, newname)`
Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.
- `def mv (self, newname)`
Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.
- `def setautoopen (self, path, interminal=False)`
Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Public Attributes

- [task](#)
- [params](#)
- [inner](#)

1.304.1 Detailed Description

A [anise.framework.files.Filestructure](#) subclass which contains the raw package, which is the typical starting point for any further packaging steps or for fetching parts of the project file structure for some processing.

This is a convenience shortcut, which is equivalent to a [anise.framework.files.TaskExecution](#) with `makerawpackage` as content generator.

1.304.2 Constructor & Destructor Documentation

1.304.2.1 `__init__()`

```
def anise.features.packages.RawPackage.__init__ (
    self )
```

1.304.3 Member Function Documentation

1.304.3.1 `datakeys()`

```
def anise.framework.files.Filestructure.datakeys (
    self ) [inherited]
```

Returns a list of keys for all stored metadata properties.

1.304.3.2 `dl()`

```
def anise.framework.files.Filestructure.dl (
    self,
    subpath = "",
    to = None,
    progresscallback = None ) [inherited]
```

Copies the complete filestructure or a subdirectory to a new destination.

Parameters

<i>subpath</i>	If given, only this subdirectory will be copied.
<i>to</i>	If given, it specifies the destination path; if not given, a new directory in the current working directory will be used.
<i>progresscallback</i>	Callback function for retrieving progress information.

1.304.3.3 `getdata()`

```
def anise.framework.files.Filestructure.getdata (
    self,
    k,
    deflt ) [inherited]
```

Gets a metadata property.

Parameters

<i>k</i>	The key name.
<i>deflt</i>	The default value (returned if the key doesn't exist).

1.304.3.4 `initialize()`

```
def anise.framework.files.Filestructure.initialize (
    self ) [inherited]
```

Initializes this Filestructure.

When this is called, the dynamic content (if any) is created and written to disc. It is automatically called for typical usage and is not required to be called manually.

1.304.3.5 `mv()`

```
def anise.framework.files.Filestructure.mv (
    self,
    newname ) [inherited]
```

Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.

Parameters

<i>newname</i>	The new name (absolute or relative).
----------------	--------------------------------------

1.304.3.6 path()

```
def anise.framework.files.Filestructure.path (
    self ) [inherited]
```

The path to this structure in the filesystem.

1.304.3.7 setautoopen()

```
def anise.framework.files.Filestructure.setautoopen (
    self,
    path,
    interterminal = False ) [inherited]
```

Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Parameters

<i>path</i>	The relative path to the file which should automatically be opened.
<i>interterminal</i>	If this hint is for terminal (non-graphical) usage.

1.304.3.8 setdata()

```
def anise.framework.files.Filestructure.setdata (
    self,
    k,
    v ) [inherited]
```

Sets a metadata property.

Parameters

<i>k</i>	The key name.
<i>v</i>	The value.

1.304.3.9 with_modified_rootname()

```
def anise.framework.files.Filestructure.with_modified_rootname (
    self,
    newname ) [inherited]
```

Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.

Parameters

<code>newname</code>	The name which the new root directory should get.
----------------------	---

1.304.4 Member Data Documentation

1.304.4.1 inner

`anise.framework.files.TaskExecution.inner` [inherited]

1.304.4.2 params

`anise.framework.files.TaskExecution.params` [inherited]

1.304.4.3 task

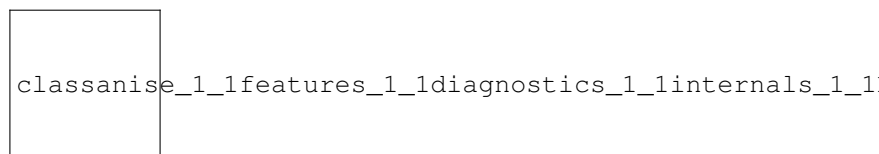
`anise.framework.files.TaskExecution.task` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/packages.py](#)

1.305 anise.features.diagnostics.internals.ReadManualApplication Class Reference

Inheritance diagram for `anise.features.diagnostics.internals.ReadManualApplication`:



Public Member Functions

- `def __init__ (self)`
- `def execute (self)`
- `def visible (self)`

Checks if this feature action shall be visible.

Public Attributes

- [label](#)
- [description](#)
- [forfeature](#)

Private Member Functions

- [def _manualpath](#) (self)

1.305.1 Constructor & Destructor Documentation

1.305.1.1 __init__()

```
def anise.features.diagnostics.internals.ReadManualApplication.__init__ (  
    self )
```

1.305.2 Member Function Documentation

1.305.2.1 _manualpath()

```
def anise.features.diagnostics.internals.ReadManualApplication._manualpath (  
    self ) [private]
```

1.305.2.2 execute()

```
def anise.features.diagnostics.internals.ReadManualApplication.execute (  
    self )
```

1.305.2.3 visible()

```
def anise.features.diagnostics.FeatureAction.visible (  
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.305.3 Member Data Documentation

1.305.3.1 description

`anise.features.diagnostics.FeatureAction.description` [inherited]

1.305.3.2 forfeature

`anise.features.diagnostics.FeatureAction.forfeature` [inherited]

1.305.3.3 label

`anise.features.diagnostics.FeatureAction.label` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/diagnostics.py](#)

1.306 anise.framework.projects.Universe.RegisteredTask Class Reference

A task which is registered to the universe object.

Public Member Functions

- `def __init__(self, parentname, taskname, label, important, invisible, sortindex, function)`

Public Attributes

- [parentname](#)
- [taskname](#)
- [name](#)
- [label](#)
- [important](#)
- [invisible](#)
- [sortindex](#)
- [function](#)

1.306.1 Detailed Description

A task which is registered to the universe object.

You typically don't need to work with this structure directly. Use [Universe.addtask](#) to add new tasks.

1.306.2 Constructor & Destructor Documentation

1.306.2.1 `__init__()`

```
def anise.framework.projects.Universe.RegisteredTask.__init__ (
    self,
    parentname,
    taskname,
    label,
    important,
    invisible,
    sortindex,
    function )
```

1.306.3 Member Data Documentation

1.306.3.1 `function`

```
anise.framework.projects.Universe.RegisteredTask.function
```

1.306.3.2 `important`

```
anise.framework.projects.Universe.RegisteredTask.important
```

1.306.3.3 `invisible`

```
anise.framework.projects.Universe.RegisteredTask.invisible
```

1.306.3.4 label

```
anise.framework.projects.Universe.RegisteredTask.label
```

1.306.3.5 name

```
anise.framework.projects.Universe.RegisteredTask.name
```

1.306.3.6 parentname

```
anise.framework.projects.Universe.RegisteredTask.parentname
```

1.306.3.7 sortindex

```
anise.framework.projects.Universe.RegisteredTask.sortindex
```

1.306.3.8 taskname

```
anise.framework.projects.Universe.RegisteredTask.taskname
```

The documentation for this class was generated from the following file:

- anise/framework/[projects.py](#)

1.307 anise.features.releasing.internals.Releasetasks Class Reference

Storage of all actions to execute for releasing.

Public Member Functions

- def [add](#) (self, task, params)
Specifies a new task and adds it into the releasing chain.

1.307.1 Detailed Description

Storage of all actions to execute for releasing.

1.307.2 Member Function Documentation

1.307.2.1 add()

```
def anise.features.releasing.internals.Releasetasks.add (  
    self,  
    task,  
    params )
```

Specifies a new task and adds it into the releasing chain.

Parameters

<i>task</i>	The task implementation to add.
<i>params</i>	Additional parameters for calling that task.

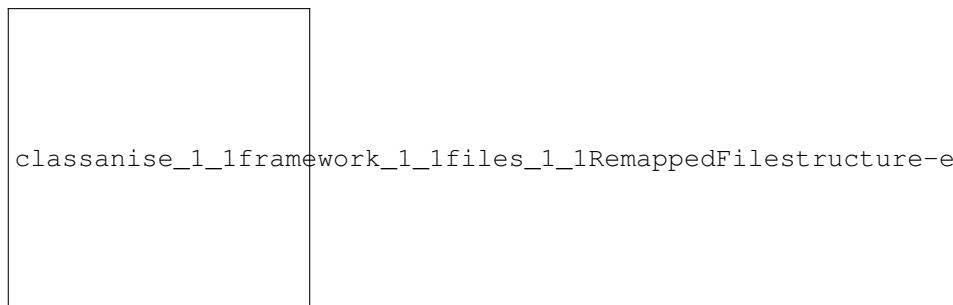
The documentation for this class was generated from the following file:

- anise/features/[releasing.py](#)

1.308 anise.framework.files.RemappedFilestructure Class Reference

Remaps a [Filestructure](#)'s inner directory hierarchy into a new tree.

Inheritance diagram for anise.framework.files.RemappedFilestructure:



Public Member Functions

- def `__init__` (self, [source](#)=None, [frommap](#)=None)
- def [getdests](#) (self)
- def [getsource](#) (self, desttuple)
- def [getmerged](#) (self)
- def [datakeys](#) (self)

Returns a list of keys for all stored metadata properties.
- def [setdata](#) (self, k, v)

Sets a metadata property.
- def [getdata](#) (self, k, deflt)

Gets a metadata property.
- def [initialize](#) (self)

Initializes this [Filestructure](#).
- def [path](#) (self)

The path to this structure in the filesystem.
- def [dl](#) (self, subpath="", to=None, progresscallback=None)

Copies the complete filestructure or a subdirectory to a new destination.
- def [with_modified_rootname](#) (self, newname)

Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.
- def [mv](#) (self, newname)

Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.
- def [setautoopen](#) (self, [path](#), interminal=False)

Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Public Attributes

- [source](#)
- [frommap](#)
- [task](#)
- [params](#)
- [inner](#)

1.308.1 Detailed Description

Remaps a [Filestructure](#)'s inner directory hierarchy into a new tree.

1.308.2 Constructor & Destructor Documentation

1.308.2.1 `__init__()`

```
def anise.framework.files.RemappedFilestructure.__init__ (
    self,
    source = None,
    frommap = None )
```

Parameters

<i>source</i>	The source anise.framework.files.Filestructure .
<i>frommap</i>	A dict of string/Filestructure. Values are relative source paths with the dict key specifying a target subpath.

1.308.3 Member Function Documentation

1.308.3.1 `datakeys()`

```
def anise.framework.files.Filestructure.datakeys (
    self ) [inherited]
```

Returns a list of keys for all stored metadata properties.

1.308.3.2 dl()

```
def anise.framework.files.Filestructure.dl (
    self,
    subpath = "",
    to = None,
    progresscallback = None ) [inherited]
```

Copies the complete filestructure or a subdirectory to a new destination.

Parameters

<i>subpath</i>	If given, only this subdirectory will be copied.
<i>to</i>	If given, it specifies the destination path; if not given, a new directory in the current working directory will be used.
<i>progresscallback</i>	Callback function for retrieving progress information.

1.308.3.3 `getdata()`

```
def anise.framework.files.Filestructure.getdata (
    self,
    k,
    deflt ) [inherited]
```

Gets a metadata property.

Parameters

<i>k</i>	The key name.
<i>deflt</i>	The default value (returned if the key doesn't exist).

1.308.3.4 `getdests()`

```
def anise.framework.files.RemappedFilestructure.getdests (
    self )
```

1.308.3.5 `getmerged()`

```
def anise.framework.files.AbstractRebuildDirectoriesFilestructure.getmerged (
    self ) [inherited]
```

1.308.3.6 `getsource()`

```
def anise.framework.files.RemappedFilestructure.getsource (
    self,
    desttuple )
```

1.308.3.7 initialize()

```
def anise.framework.files.Filestructure.initialize (
    self ) [inherited]
```

Initializes this [Filestructure](#).

When this is called, the dynamic content (if any) is created and written to disc. It is automatically called for typical usage and is not required to be called manually.

1.308.3.8 mv()

```
def anise.framework.files.Filestructure.mv (
    self,
    newname ) [inherited]
```

Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.

Parameters

<i>newname</i>	The new name (absolute or relative).
----------------	--------------------------------------

1.308.3.9 path()

```
def anise.framework.files.Filestructure.path (
    self ) [inherited]
```

The path to this structure in the filesystem.

1.308.3.10 setautoopen()

```
def anise.framework.files.Filestructure.setautoopen (
    self,
    path,
    interminal = False ) [inherited]
```

Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Parameters

<i>path</i>	The relative path to the file which should automatically be opened.
<i>interminal</i>	If this hint is for terminal (non-graphical) usage.

1.308.3.11 setdata()

```
def anise.framework.files.Filestructure.setdata (
    self,
    k,
    v ) [inherited]
```

Sets a metadata property.

Parameters

<i>k</i>	The key name.
<i>v</i>	The value.

1.308.3.12 with_modified_rootname()

```
def anise.framework.files.Filestructure.with_modified_rootname (
    self,
    newname ) [inherited]
```

Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.

Parameters

<i>newname</i>	The name which the new root directory should get.
----------------	---

1.308.4 Member Data Documentation

1.308.4.1 frommap

```
anise.framework.files.RemappedFilestructure.frommap
```

1.308.4.2 inner

```
anise.framework.files.TaskExecution.inner [inherited]
```

1.308.4.3 params

`anise.framework.files.TaskExecution.params` [inherited]

1.308.4.4 source

`anise.framework.files.RemappedFilestructure.source`

1.308.4.5 task

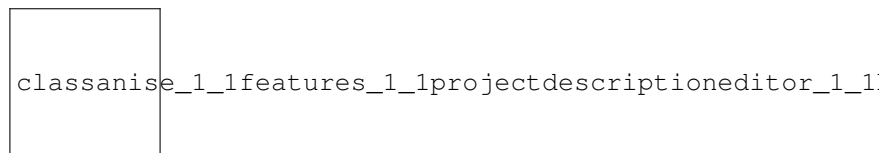
`anise.framework.files.TaskExecution.task` [inherited]

The documentation for this class was generated from the following file:

- `anise/framework/files.py`

1.309 anise.features.projectdescriptioneditor.base.RenameTaskCustomAction Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.base.RenameTaskCustomAction`:



Public Member Functions

- `def __init__(self)`
- `def visible(self, ais, node)`
- `def execute(self, ais, node)`

Public Attributes

- `label`
- `function`
- `icon`

1.309.1 Constructor & Destructor Documentation

1.309.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.base.RenameTaskCustomAction.__init__ (
    self )
```

1.309.2 Member Function Documentation

1.309.2.1 `execute()`

```
def anise.features.projectdescriptioneditor.base.RenameTaskCustomAction.execute (
    self,
    ais,
    node )
```

1.309.2.2 `visible()`

```
def anise.features.projectdescriptioneditor.base.RenameTaskCustomAction.visible (
    self,
    ais,
    node )
```

1.309.3 Member Data Documentation

1.309.3.1 `function`

```
anise.features.projectdescriptioneditor.CustomAction.function [inherited]
```

1.309.3.2 `icon`

```
anise.features.projectdescriptioneditor.CustomAction.icon [inherited]
```

1.309.3.3 `label`

```
anise.features.projectdescriptioneditor.CustomAction.label [inherited]
```

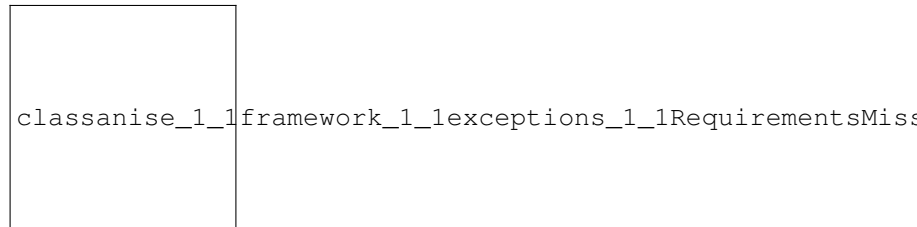
The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/base.py](#)

1.310 anise.framework.exceptions.RequirementsMissingError Class Reference

Something required is not in place.

Inheritance diagram for anise.framework.exceptions.RequirementsMissingError:



Public Member Functions

- def [__call__](#) (self, args)

1.310.1 Detailed Description

Something required is not in place.

1.310.2 Member Function Documentation

1.310.2.1 [__call__](#)()

```
def anise.framework.exceptions.AniseError.__call__ (
    self,
    args ) [inherited]
```

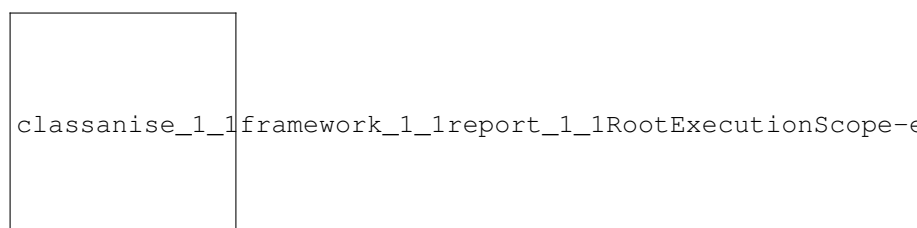
The documentation for this class was generated from the following file:

- anise/framework/[exceptions.py](#)

1.311 anise.framework.report.RootExecutionScope Class Reference

The root execution scope.

Inheritance diagram for anise.framework.report.RootExecutionScope:



Public Member Functions

- def `__init__` (self)
- def `appendchild` (self, child)
Appends a child scope to this scope.
- def `progress` (self)
Returns the current progress state of this scope.
- def `setprogress` (self, value)
Sets the progress of this scope.
- def `title` (self)
Returns the title string of this scope.
- def `rawsubparts` (self)
Returns a list of all direct subparts of this scope.
- def `newsubscope` (self, title)
Adds a new sub scope to the current top scope.
- def `topscope` (self, i=0)
Returns a scope from the top of the stack (without removing it).
- def `__enter__` (self)
- def `__exit__` (self, etype, evalue, etraceback)
- def `id` (self)
Returns the ID of this part.
- def `value` (self)
Returns the content value of this execution scope.
- def `hints` (self)
Returns the content hint.
- def `address` (self)
Returns the address of this part.
- def `getscopepartsbyidrange` (self, minid=None, maxid=None)
Gets a list of execution scope parts by a range of IDs.

Private Attributes

- `_lock`

1.311.1 Detailed Description

The root execution scope.

1.311.2 Constructor & Destructor Documentation

1.311.2.1 `__init__()`

```
def anise.framework.report.RootExecutionScope.__init__ (
    self )
```


1.311.3 Member Function Documentation

1.311.3.1 `__enter__()`

```
def anise.framework.report.ExecutionScope.__enter__ (
    self ) [inherited]
```

1.311.3.2 `__exit__()`

```
def anise.framework.report.ExecutionScope.__exit__ (
    self,
    etype,
    evalue,
    etraceback ) [inherited]
```

1.311.3.3 `address()`

```
def anise.framework.report.ExecutionScopePart.address (
    self ) [inherited]
```

Returns the address of this part.

1.311.3.4 `appendchild()`

```
def anise.framework.report.ExecutionScope.appendchild (
    self,
    child ) [inherited]
```

Appends a child scope to this scope.

1.311.3.5 `getscopepartsbyidrange()`

```
def anise.framework.report.ExecutionScopePart.getscopepartsbyidrange (
    self,
    minid = None,
    maxid = None ) [inherited]
```

Gets a list of execution scope parts by a range of IDs.

1.311.3.6 hints()

```
def anise.framework.report.ExecutionScopePart.hints (
    self ) [inherited]
```

Returns the content hint.

This is typically used for storing the severity of log messages, so the presentation can differ.

1.311.3.7 id()

```
def anise.framework.report.ExecutionScopePart.id (
    self ) [inherited]
```

Returns the ID of this part.

1.311.3.8 newsubscope()

```
def anise.framework.report.ExecutionScope.newsubscope (
    self,
    title ) [inherited]
```

Adds a new sub scope to the current top scope.

Parameters

<i>title</i>	The scope title text.
--------------	-----------------------

1.311.3.9 progress()

```
def anise.framework.report.ExecutionScope.progress (
    self ) [inherited]
```

Returns the current progress state of this scope.

Returns

: An instance of [ExecutionScope.Progress](#).

1.311.3.10 rawsubparts()

```
def anise.framework.report.ExecutionScope.rawsubparts (
    self ) [inherited]
```

Returns a list of all direct subparts of this scope.

1.311.3.11 setprogress()

```
def anise.framework.report.ExecutionScope.setprogress (
    self,
    value ) [inherited]
```

Sets the progress of this scope.

Parameters

<i>value</i>	An instance of ExecutionScope.Progress .
--------------	--

1.311.3.12 title()

```
def anise.framework.report.ExecutionScope.title (
    self ) [inherited]
```

Returns the title string of this scope.

1.311.3.13 topscope()

```
def anise.framework.report.ExecutionScope.topscope (
    self,
    i = 0 ) [inherited]
```

Returns a scope from the top of the stack (without removing it).

Parameters

<i>i</i>	Which scope to return (counting from top).
----------	--

1.311.3.14 value()

```
def anise.framework.report.ExecutionScopePart.value (
    self ) [inherited]
```

Returns the content value of this execution scope.

1.311.4 Member Data Documentation

1.311.4.1 `_lock`

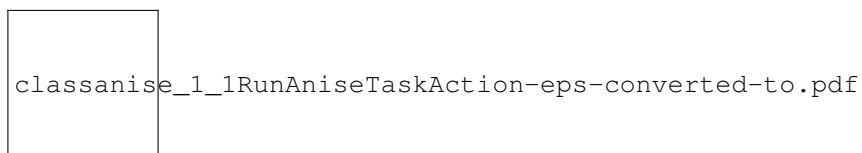
```
anise.framework.report.RootExecutionScope._lock [private]
```

The documentation for this class was generated from the following file:

- [anise/framework/report.py](#)

1.312 `anise.RunAniseTaskAction` Class Reference

Inheritance diagram for `anise.RunAniseTaskAction`:



Public Member Functions

- `def __init__` (self, [rootpath](#), task)
- `def action` (self, info)

Public Attributes

- [rootpath](#)
- [taskname](#)

1.312.1 Constructor & Destructor Documentation

1.312.1.1 `__init__()`

```
def anise.RunAniseTaskAction.__init__ (
    self,
    rootpath,
    task )
```

1.312.2 Member Function Documentation

1.312.2.1 action()

```
def anise.RunAniseTaskAction.action (
    self,
    info )
```

1.312.3 Member Data Documentation

1.312.3.1 rootpath

```
anise.RunAniseTaskAction.rootpath
```

1.312.3.2 taskname

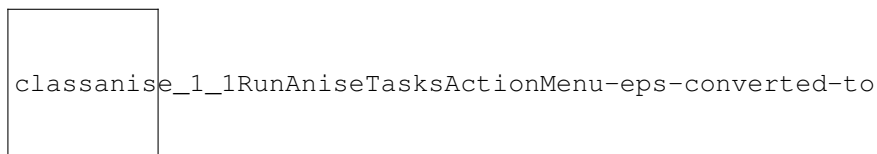
```
anise.RunAniseTaskAction.taskname
```

The documentation for this class was generated from the following file:

- [_meta/shallot_plugin/anise.py](#)

1.313 anise.RunAniseTasksActionMenu Class Reference

Inheritance diagram for anise.RunAniseTasksActionMenu:



Public Member Functions

- def [__init__](#) (self, nodes)
- def [initialize](#) (self)

Public Attributes

- [eurl](#)

1.313.1 Constructor & Destructor Documentation

1.313.1.1 `__init__()`

```
def anise.RunAniseTasksActionMenu.__init__ (
    self,
    nodes )
```

1.313.2 Member Function Documentation

1.313.2.1 `initialize()`

```
def anise.RunAniseTasksActionMenu.initialize (
    self )
```

1.313.3 Member Data Documentation

1.313.3.1 `eurl`

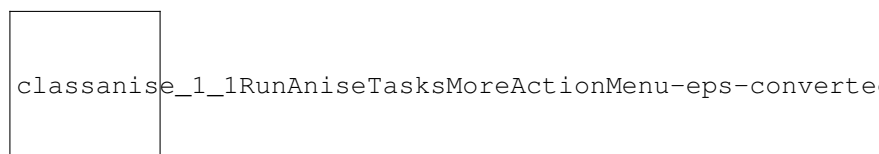
```
anise.RunAniseTasksActionMenu.eurl
```

The documentation for this class was generated from the following file:

- `_meta/shallot_plugin/anise.py`

1.314 `anise.RunAniseTasksMoreActionMenu` Class Reference

Inheritance diagram for `anise.RunAniseTasksMoreActionMenu`:



Public Member Functions

- `def __init__` (self, subitems)

1.314.1 Constructor & Destructor Documentation

1.314.1.1 `__init__()`

```
def anise.RunAniseTasksMoreActionMenu.__init__ (
    self,
    subitems )
```

The documentation for this class was generated from the following file:

- `_meta/shallot_plugin/anise.py`

1.315 anise.features.homepage.internals.Sections Class Reference

Storage of all homepage sections.

Public Member Functions

- def `__init__` (self)
- def `add` (self, index, key, header, textfct, additionalparams)
Adds a homepage section.
- def `getsectionbykey` (self, key)
Gets the section record for a key.
- def `gettextfct` (self, key)
Gets the generator function for a key, which returns the section Markdown source.
- def `replacetextfct` (self, key, textfct)
Replaces a generator function for a key.
- def `addicon` (self, index, path)
Adds an icon somewhere in the section menu, e.g.

Public Attributes

- `list`

1.315.1 Detailed Description

Storage of all homepage sections.

1.315.2 Constructor & Destructor Documentation

1.315.2.1 `__init__()`

```
def anise.features.homepage.internals.Sections.__init__ (
    self )
```

1.315.3 Member Function Documentation

1.315.3.1 `add()`

```
def anise.features.homepage.internals.Sections.add (
    self,
    index,
    key,
    header,
    textfct,
    additionalparams )
```

Adds a homepage section.

Parameters

<i>index</i>	A position index number. This controls the ordering of sections.
<i>key</i>	The section name. Also used as heading.
<i>header</i>	The section header text.
<i>textfct</i>	A function returning the section Markdown source as string.
<i>additionalparams</i>	Additional section parameters (advanced).

1.315.3.2 `addicon()`

```
def anise.features.homepage.internals.Sections.addicon (
    self,
    index,
    path )
```

Adds an icon somewhere in the section menu, e.g.

as splitter or for decoration.

Parameters

<i>index</i>	A position index number. This controls the ordering of sections.
<i>path</i>	Path to the icon file. Relative to the project root directory.

1.315.3.3 getsectionbykey()

```
def anise.features.homepage.internals.Sections.getsectionbykey (
    self,
    key )
```

Gets the section record for a key.

This is part of a particular piece of internal infrastructure and is typically not required to be used.

Parameters

<i>key</i>	The section name.
------------	-------------------

1.315.3.4 gettextfct()

```
def anise.features.homepage.internals.Sections.gettextfct (
    self,
    key )
```

Gets the generator function for a key, which returns the section Markdown source.

Parameters

<i>key</i>	The section name.
------------	-------------------

Returns

: The generator function.

1.315.3.5 replacetextfct()

```
def anise.features.homepage.internals.Sections.replacetextfct (
    self,
    key,
    textfct )
```

Replaces a generator function for a key.

Parameters

<i>key</i>	The section name.
<i>textfct</i>	The new generator function.

1.315.4 Member Data Documentation

1.315.4.1 list

```
anise.features.homepage.internals.Sections.list
```

The documentation for this class was generated from the following file:

- [anise/features/homepage.py](#)

1.316 anise.features.packages.debian.ServiceDescription Class Reference

Description for Debian services to be included in a package.

Public Member Functions

- `def __init__(self, name, command)`

Public Attributes

- `name`
- `command`

1.316.1 Detailed Description

Description for Debian services to be included in a package.

1.316.2 Constructor & Destructor Documentation

1.316.2.1 __init__()

```
def anise.features.packages.debian.ServiceDescription.__init__(
    self,
    name,
    command )
```

Parameters

<i>name</i>	The display name.
<i>command</i>	The command to be executed.

1.316.3 Member Data Documentation

1.316.3.1 command

`anise.features.packages.debian.ServiceDescription.command`

1.316.3.2 name

`anise.features.packages.debian.ServiceDescription.name`

The documentation for this class was generated from the following file:

- [anise/features/packages/debian.py](#)

1.317 anise.features.ui.internals.SetImplicitStopAllowed Class Reference

Controls if implicit application stop is enabled (e.g.

Public Member Functions

- `def __init__ (self, val)`
- `def __enter__ (self)`
- `def __exit__ (self, exc_type, exc_val, exc_tb)`

1.317.1 Detailed Description

Controls if implicit application stop is enabled (e.g.

not asking user for confirmation).

This is part of a particular piece of internal infrastructure and is typically not required to be used.

1.317.2 Constructor & Destructor Documentation

1.317.2.1 `__init__()`

```
def anise.features.ui.internals.SetImplicitStopAllowed.__init__ (
    self,
    val )
```

1.317.3 Member Function Documentation

1.317.3.1 `__enter__()`

```
def anise.features.ui.internals.SetImplicitStopAllowed.__enter__ (
    self )
```

1.317.3.2 `__exit__()`

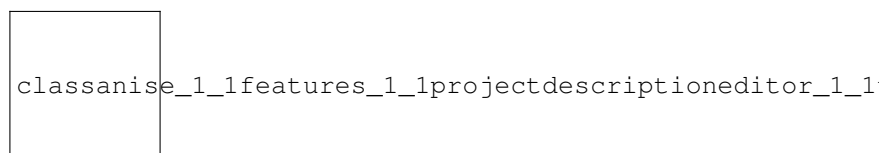
```
def anise.features.ui.internals.SetImplicitStopAllowed.__exit__ (
    self,
    exc_type,
    exc_val,
    exc_tb )
```

The documentation for this class was generated from the following file:

- [anise/features/ui.py](#)

1.318 `anise.features.projectdescriptioneditor.testing.SetReleaseCriticalCustomAction` Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.testing.SetReleaseCriticalCustomAction`:



Public Member Functions

- def `__init__` (self)
- def `visible` (self, ais, node)
- def `execute` (self, ais, node)

Public Attributes

- [label](#)
- [function](#)
- [icon](#)

1.318.1 Constructor & Destructor Documentation

1.318.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.testing.SetReleaseCriticalCustomAction.__init__ (
    self )
```

1.318.2 Member Function Documentation

1.318.2.1 `execute()`

```
def anise.features.projectdescriptioneditor.testing.SetReleaseCriticalCustomAction.execute (
    self,
    ais,
    node )
```

1.318.2.2 `visible()`

```
def anise.features.projectdescriptioneditor.testing.SetReleaseCriticalCustomAction.visible (
    self,
    ais,
    node )
```

1.318.3 Member Data Documentation

1.318.3.1 `function`

```
anise.features.projectdescriptioneditor.CustomAction.function [inherited]
```

1.318.3.2 icon

`anise.features.projectdescriptioneditor.CustomAction.icon` [inherited]

1.318.3.3 label

`anise.features.projectdescriptioneditor.CustomAction.label` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/testing.py](#)

1.319 anise.utils.logging.Severity Class Reference

Enumeration of log message severities.

Static Public Attributes

- [Debug](#)
- [Info](#)
- [Warning](#)
- [Error](#)

1.319.1 Detailed Description

Enumeration of log message severities.

1.319.2 Member Data Documentation

1.319.2.1 Debug

`anise.utils.logging.Severity.Debug` [static]

1.319.2.2 Error

`anise.utils.logging.Severity.Error` [static]

1.319.2.3 Info

```
anise.utils.logging.Severity.Info [static]
```

1.319.2.4 Warning

```
anise.utils.logging.Severity.Warning [static]
```

The documentation for this class was generated from the following file:

- [anise/utils/logging.py](#)

1.320 anise.features.documentation.internals.ShortSnippets Class Reference

Storage of text snippets for use in the documentation texts like @a_foo.

Public Member Functions

- `def __init__ (self)`
- `def add (self, name, text)`
Adds a text snippet to the storage, so it can be used.
- `def toaliases (self)`

Public Attributes

- [storage](#)

1.320.1 Detailed Description

Storage of text snippets for use in the documentation texts like @a_foo.

1.320.2 Constructor & Destructor Documentation

1.320.2.1 `__init__()`

```
def anise.features.documentation.internals.ShortSnippets.__init__ (  
    self )
```

1.320.3 Member Function Documentation

1.320.3.1 `add()`

```
def anise.features.documentation.internals.ShortSnippets.add (  
    self,  
    name,  
    text )
```

Adds a text snippet to the storage, so it can be used.

Parameters

<i>name</i>	The snippet name. This is the <code>foo</code> in <code>@a_foo</code> .
<i>text</i>	The snippet text.

1.320.3.2 `toaliases()`

```
def anise.features.documentation.internals.ShortSnippets.toaliases (
    self )
```

1.320.4 Member Data Documentation

1.320.4.1 `storage`

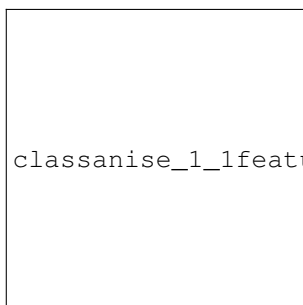
```
anise.features.documentation.internals.ShortSnippets.storage
```

The documentation for this class was generated from the following file:

- [anise/features/documentation.py](#)

1.321 `anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication` Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication`:



Public Member Functions

- def `__init__` (self, `forfeature=""`)
- def `getcustomactions` (self, ais)
- def `getentries` (self, ais)
- def `getentrylabel` (self, ais, entry)
- def `getsample` (self, tname, sampletype)
- def `execute` (self)
- def `manageentry` (self, ais, node)
Called for managing one of the entries.
- def `visible` (self)
Checks if this feature action shall be visible.

Static Public Member Functions

- def `getnewkey` (defaultname)

Public Attributes

- `header`
- `label`
- `description`
- `forfeature`

Static Public Attributes

- string `sample_function`
- string `sample_class`

Private Member Functions

- def `_add` (self, ais)

Static Private Member Functions

- def `_getallhooks` ()

1.321.1 Constructor & Destructor Documentation

1.321.1.1 `__init__` ()

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.__init__ (
    self,
    forfeature = "" )
```

1.321.2 Member Function Documentation

1.321.2.1 `_add()`

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication._add (
    self,
    ais ) [private]
```

1.321.2.2 `_getallhooks()`

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication._getallhooks
( ) [static], [private]
```

1.321.2.3 `execute()`

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (
    self ) [inherited]
```

1.321.2.4 `getcustomactions()`

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.getcustomactions
(
    self,
    ais )
```

1.321.2.5 `getentries()`

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.getentries (
    self,
    ais )
```

1.321.2.6 getentrylabel()

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.getentrylabel (
    (
        self,
        ais,
        entry )
```

1.321.2.7 getnewkey()

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.getnewkey (
    defaultname ) [static]
```

1.321.2.8 getsample()

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.getsample (
    self,
    tname,
    sampletype )
```

1.321.2.9 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (
    self,
    ais,
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.321.2.10 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.321.3 Member Data Documentation

1.321.3.1 description

```
anise.features.diagnostics.FeatureAction.description [inherited]
```

1.321.3.2 forfeature

```
anise.features.diagnostics.FeatureAction.forfeature [inherited]
```

1.321.3.3 header

```
anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header [inherited]
```

1.321.3.4 label

```
anise.features.diagnostics.FeatureAction.label [inherited]
```

1.321.3.5 sample_class

```
string anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.sample_↵  
class [static]
```

Initial value:

```
= """class HookHandler_{ttname}{sbaseclass}:  
# This is sample content. Change it as needed!  
  
{body}  
"""
```

1.321.3.6 sample_function

```
string anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.sample_↵  
function [static]
```

Initial value:

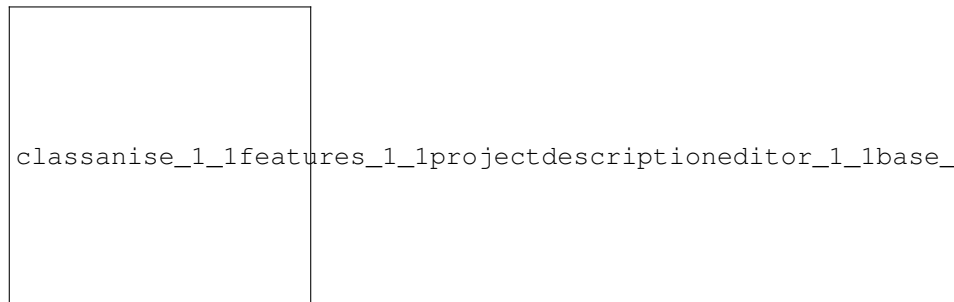
```
= """def on_{ttname}({signature}):  
# This is sample content. Change it as needed!  
{body}  
"""
```

The documentation for this class was generated from the following file:

- anise/features/projectdescriptioneditor/[base.py](#)

1.322 anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication_↵
InBase Class Reference

Inheritance diagram for anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication_InBase:



Public Member Functions

- def [__init__](#) (self)
- def [getcustomactions](#) (self, ais)
- def [getentries](#) (self, ais)
- def [getentrylabel](#) (self, ais, entry)
- def [getsample](#) (self, tname, sampletype)
- def [execute](#) (self)
- def [manageentry](#) (self, ais, node)
Called for managing one of the entries.
- def [visible](#) (self)
Checks if this feature action shall be visible.

Static Public Member Functions

- def [getnewkey](#) (defaultname)

Public Attributes

- [header](#)
- [label](#)
- [description](#)
- [forfeature](#)

Static Public Attributes

- string [sample_function](#)
- string [sample_class](#)

1.322.1 Constructor & Destructor Documentation

1.322.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication_InBase.__↵  
init__ (   
    self )
```

1.322.2 Member Function Documentation

1.322.2.1 `execute()`

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (   
    self ) [inherited]
```

1.322.2.2 `getcustomactions()`

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.getcustomactions   
(   
    self,   
    ais ) [inherited]
```

1.322.2.3 `getentries()`

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.getentries (   
    self,   
    ais ) [inherited]
```

1.322.2.4 getentrylabel()

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.getentrylabel (
    self,
    ais,
    entry ) [inherited]
```

1.322.2.5 getnewkey()

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.getnewkey (
    defaultname ) [static], [inherited]
```

1.322.2.6 getsample()

```
def anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.getsample (
    self,
    tname,
    sampletype ) [inherited]
```

1.322.2.7 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (
    self,
    ais,
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.322.2.8 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.322.3 Member Data Documentation

1.322.3.1 description

```
anise.features.diagnostics.FeatureAction.description [inherited]
```

1.322.3.2 forfeature

```
anise.features.diagnostics.FeatureAction.forfeature [inherited]
```

1.322.3.3 header

```
anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header [inherited]
```

1.322.3.4 label

```
anise.features.diagnostics.FeatureAction.label [inherited]
```

1.322.3.5 sample_class

```
string anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.sample_↵
class [static], [inherited]
```

Initial value:

```
= """class HookHandler_{ttname}{sbaseclass}:
# This is sample content. Change it as needed!

{body}
"""
```


1.322.3.6 sample_function

```
string anise.features.projectdescriptioneditor.base.SimpleScriptAssistantApplication.sample_↵
function [static], [inherited]
```

Initial value:

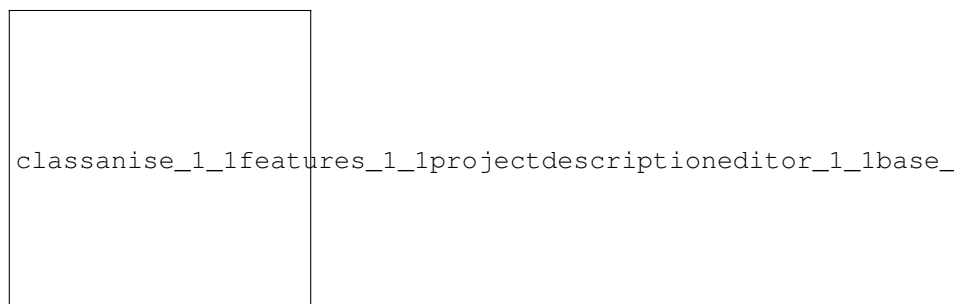
```
= """def on_{ttname}({signature}):
# This is sample content. Change it as needed!
{body}
"""
```

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/base.py](#)

1.323 anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication Class Reference

Inheritance diagram for anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication:

**Public Member Functions**

- def [__init__](#) (self, [forfeature](#)=[""](#))
- def [getcustomactions](#) (self, ais)
- def [getentries](#) (self, ais)
- def [getentrylabel](#) (self, ais, entry)
- def [execute](#) (self)
- def [manageentry](#) (self, ais, node)
Called for managing one of the entries.
- def [visible](#) (self)
Checks if this feature action shall be visible.

Static Public Member Functions

- def [getnewkey](#) (defaultname, ais)

Public Attributes

- [header](#)
- [label](#)
- [description](#)
- [forfeature](#)

Private Member Functions

- [def `_add`](#) (self, ais)

1.323.1 Constructor & Destructor Documentation

1.323.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication.__init__ (
    self,
    forfeature = "" )
```

1.323.2 Member Function Documentation

1.323.2.1 `_add()`

```
def anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication._add (
    self,
    ais ) [private]
```

1.323.2.2 `execute()`

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (
    self ) [inherited]
```

1.323.2.3 `getcustomactions()`

```
def anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication.getcustomactions
(
    self,
    ais )
```

1.323.2.4 getentries()

```
def anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication.getentries (
    self,
    ais )
```

1.323.2.5 getentrylabel()

```
def anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication.getentrylabel
(
    self,
    ais,
    entry )
```

1.323.2.6 getnewkey()

```
def anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication.getnewkey (
    defaultname,
    ais ) [static]
```

1.323.2.7 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (
    self,
    ais,
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.323.2.8 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.323.3 Member Data Documentation

1.323.3.1 description

`anise.features.diagnostics.FeatureAction.description` [inherited]

1.323.3.2 forfeature

`anise.features.diagnostics.FeatureAction.forfeature` [inherited]

1.323.3.3 header

`anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header` [inherited]

1.323.3.4 label

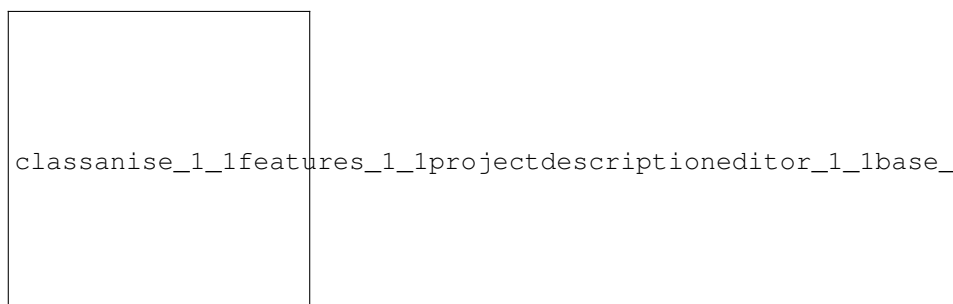
`anise.features.diagnostics.FeatureAction.label` [inherited]

The documentation for this class was generated from the following file:

- `anise/features/projectdescriptioneditor/base.py`

1.324 `anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication_In`↵ Base Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication_InBase`:



Public Member Functions

- def `__init__` (self)
- def `getcustomactions` (self, ais)
- def `getentries` (self, ais)
- def `getentrylabel` (self, ais, entry)
- def `execute` (self)
- def `manageentry` (self, ais, node)
Called for managing one of the entries.
- def `visible` (self)
Checks if this feature action shall be visible.

Static Public Member Functions

- def `getnewkey` (defaultname, ais)

Public Attributes

- `header`
- `label`
- `description`
- `forfeature`

1.324.1 Constructor & Destructor Documentation

1.324.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication_InBase.__↵  
init__ (   
    self )
```

1.324.2 Member Function Documentation

1.324.2.1 `execute()`

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.execute (   
    self ) [inherited]
```

1.324.2.2 getcustomactions()

```
def anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication.getcustomactions
(
    self,
    ais ) [inherited]
```

1.324.2.3 getentries()

```
def anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication.getentries (
    self,
    ais ) [inherited]
```

1.324.2.4 getentrylabel()

```
def anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication.getentrylabel
(
    self,
    ais,
    entry ) [inherited]
```

1.324.2.5 getnewkey()

```
def anise.features.projectdescriptioneditor.base.SimpleTaskAssistantApplication.getnewkey (
    defaultname,
    ais ) [static], [inherited]
```

1.324.2.6 manageentry()

```
def anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.manageentry (
    self,
    ais,
    node ) [inherited]
```

Called for managing one of the entries.

The default implementation opens the standard node editor (as the raw mode editor would provide it).

Parameters

<i>ais</i>	An instance of anise.framework.projects.IntermediateStructure .
<i>node</i>	An instance of anise.framework.projects.IntermediateStructure.Node .

1.324.2.7 visible()

```
def anise.features.diagnostics.FeatureAction.visible (
    self ) [inherited]
```

Checks if this feature action shall be visible.

Override this method in custom subclasses or leave the default implementation.

1.324.3 Member Data Documentation

1.324.3.1 description

```
anise.features.diagnostics.FeatureAction.description [inherited]
```

1.324.3.2 forfeature

```
anise.features.diagnostics.FeatureAction.forfeature [inherited]
```

1.324.3.3 header

```
anise.features.projectdescriptioneditor.ProjectDescriptionEditorAssistant.header [inherited]
```

1.324.3.4 label

```
anise.features.diagnostics.FeatureAction.label [inherited]
```

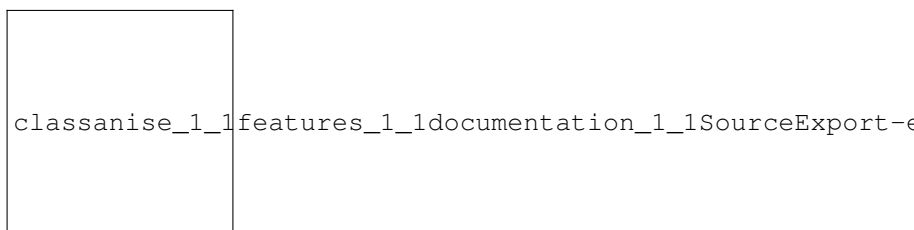
The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/base.py](#)

1.325 anise.features.documentation.SourceExport Class Reference

Describes an export of an anise documentation back into the source tree.

Inheritance diagram for anise.features.documentation.SourceExport:



Public Member Functions

- `def __init__ (self, key, outputfile, format, name=None, heading=None, includedeveloperdoc=False, extractall=True, extractprivate=True, hideundocumented=False, imagepaths=None, additionaltext="", shownavbar=False, excludepatterns=None, predefineddocmacros=None, head3=None, head4=None, source=None, sources=None)`
- `def generate (self, targetpath)`
Generates the content.

Public Attributes

- `outputfile`
- `key`
- `format`
- `name`
- `heading`
- `includedeveloperdoc`
- `extractall`
- `extractprivate`
- `hideundocumented`
- `imagepaths`
- `additionaltext`
- `shownavbar`
- `excludepatterns`
- `predefineddocmacros`
- `head3`
- `head4`
- `source`
- `sources`

1.325.1 Detailed Description

Describes an export of an anise documentation back into the source tree.

1.325.2 Constructor & Destructor Documentation

1.325.2.1 `__init__()`

```
def anise.features.documentation.SourceExport.__init__ (
    self,
    key,
    outputfile,
    format,
    name = None,
    heading = None,
    includedeveloperdoc = False,
    extractall = True,
    extractprivate = True,
    hideundocumented = False,
    imagepaths = None,
    additionaltext = "",
    shownavbar = False,
    excludepatterns = None,
    predefineddocmacros = None,
    head3 = None,
    head4 = None,
    source = None,
    sources = None )
```

Parameters

<i>key</i>	One of the keys stored in pool. Defines which documentation to export.
<i>outputfile</i>	Target file path relative to the 'raw package's root.
<i>format</i>	The output format (one of documentation.ExportFormat).
<i>name</i>	The export definition name. Finding definitions by name is possible with <code>exports.getbyname</code> .
<i>heading</i>	The title.
<i>includedeveloperdoc</i>	If the output shall include the developer documentation as well (does not work for all output formats).
<i>extractall</i>	If a developer documentation should include all elements (not just the documented ones).
<i>extractprivate</i>	If a developer documentation should include also private members.
<i>hideundocumented</i>	If the develo documentation should entirely leave out undocumented members.
<i>imagepaths</i>	Directories from where images are used in the documentation.
<i>additionaltext</i>	Additional text to append to the selected documentation.
<i>shownavbar</i>	If to show navigation bar. Does not work in all output formats.
<i>excludepatterns</i>	File patterns to exclude from reading.
<i>predefineddocmacros</i>	List of additional macros to define for reading.
<i>head3</i>	3rd heading.
<i>head4</i>	4th heading.
<i>source</i>	The documentation source root.
<i>sources</i>	Optional list of source paths (default: ["/"]).

1.325.3 Member Function Documentation

1.325.3.1 generate()

```
def anise.features.documentation.internals.Export.generate (
    self,
    targetpath ) [inherited]
```

Generates the content.

Override this method in custom subclasses or leave the default implementation.

1.325.4 Member Data Documentation

1.325.4.1 additionaltext

```
anise.features.documentation.internals.Export.additionaltext [inherited]
```

1.325.4.2 excludepatterns

```
anise.features.documentation.internals.Export.excludepatterns [inherited]
```

1.325.4.3 extractall

```
anise.features.documentation.internals.Export.extractall [inherited]
```

1.325.4.4 extractprivate

```
anise.features.documentation.internals.Export.extractprivate [inherited]
```

1.325.4.5 format

`anise.features.documentation.internals.Export.format` [inherited]

1.325.4.6 head3

`anise.features.documentation.internals.Export.head3` [inherited]

1.325.4.7 head4

`anise.features.documentation.internals.Export.head4` [inherited]

1.325.4.8 heading

`anise.features.documentation.internals.Export.heading` [inherited]

1.325.4.9 hideundocumented

`anise.features.documentation.internals.Export.hideundocumented` [inherited]

1.325.4.10 imagepaths

`anise.features.documentation.internals.Export.imagepaths` [inherited]

1.325.4.11 includedeveloperdoc

`anise.features.documentation.internals.Export.includedeveloperdoc` [inherited]

1.325.4.12 key

`anise.features.documentation.internals.Export.key` [inherited]

1.325.4.13 name

`anise.features.documentation.internals.Export.name` [inherited]

1.325.4.14 outputfile

`anise.features.documentation.internals.AbstractFileExport.outputfile` [inherited]

1.325.4.15 predefineddocmacros

`anise.features.documentation.internals.Export.predefineddocmacros` [inherited]

1.325.4.16 shownavbar

`anise.features.documentation.internals.Export.shownavbar` [inherited]

1.325.4.17 source

`anise.features.documentation.internals.Export.source` [inherited]

1.325.4.18 sources

`anise.features.documentation.internals.Export.sources` [inherited]

The documentation for this class was generated from the following file:

- [anise/features/documentation.py](#)

1.326 anise.framework.engine.SpecialPaths Class Reference

A data structure of some particular special paths.

Public Member Functions

- `def __init__(self, projectfile, projectdir)`

Public Attributes

- [projectfile](#)
- [projectdir](#)
- [relativeprojectfile](#)
- [anisedir](#)
- [anisedatadir](#)
- [startdir](#)

1.326.1 Detailed Description

A data structure of some particular special paths.

1.326.2 Constructor & Destructor Documentation

1.326.2.1 `__init__()`

```
def anise.framework.engine.SpecialPaths.__init__ (
    self,
    projectfile,
    projectdir )
```

1.326.3 Member Data Documentation

1.326.3.1 `anisedatadir`

```
anise.framework.engine.SpecialPaths.anisedatadir
```

1.326.3.2 `anisedir`

```
anise.framework.engine.SpecialPaths.anisedir
```

1.326.3.3 `projectdir`

```
anise.framework.engine.SpecialPaths.projectdir
```

1.326.3.4 projectfile

```
anise.framework.engine.SpecialPaths.projectfile
```

1.326.3.5 relativeprojectfile

```
anise.framework.engine.SpecialPaths.relativeprojectfile
```

1.326.3.6 startdir

```
anise.framework.engine.SpecialPaths.startdir
```

The documentation for this class was generated from the following file:

- anise/framework/[engine.py](#)

1.327 anise.features.projectdescriptioneditor.filetransfer.SshFiletransfer Class Reference

Public Member Functions

- def [__init__](#) (self)
- def [ask](#) (self)

Public Attributes

- [label](#)

1.327.1 Constructor & Destructor Documentation

1.327.1.1 __init__()

```
def anise.features.projectdescriptioneditor.filetransfer.SshFiletransfer.__init__ (
    self )
```

1.327.2 Member Function Documentation

1.327.2.1 ask()

```
def anise.features.projectdescriptioneditor.filetransfer.SshFiletransfer.ask (
    self )
```

1.327.3 Member Data Documentation

1.327.3.1 label

```
anise.features.projectdescriptioneditor.filetransfer.SshFiletransfer.label
```

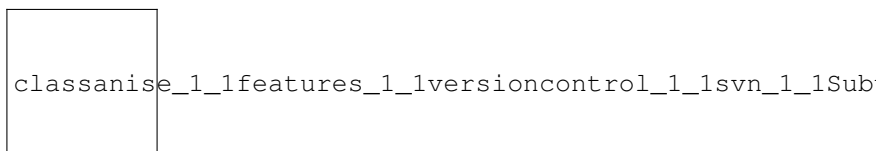
The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/filetransfer.py](#)

1.328 anise.features.versioncontrol.svn.SubversionVersionControlSystem Class Reference

Subversion version control system adapter.

Inheritance diagram for anise.features.versioncontrol.svn.SubversionVersionControlSystem:



Public Member Functions

- def [__init__](#) (self)
- def [getfullversion](#) (self)
- def [createrawpackage](#) (self)
- def [syncversioncontrolsystem](#) (self, commitmessage="", forceskipped=True, forceunchanged=False)
- def [findvcs](#) (self)
- def [bindvcs](#) (self, l)
- def [getheadrevision](#) (self)
- def [fetchrawcommitmessage](#) (self, revision)
- def [getcommitmessagefromrawcommitmessage](#) (self, s)
- def [getpreviousrevision](#) (self, revision)

Gets the predecessor revision for a given revision.
- def [getcommitmessage](#) (self, revision)

Returns the raw commit message (as formatted by the vcs tool) for a revision.
- def [getchangelog](#) (self, asstring=True)

Computes the project change log from information in specially formatted vcs commit messages.

- def `addchange` (self, changetext=None)
Adds a change to the changelog.
- def `addlabel` (self, label=None)
Adds a version label for the current state.
- def `storechange` (self, changetext)
Stores a new entry to the changelog.
- def `storelabel` (self, label)
Stores a new version label.
- def `branchfromhere` (self, branchname=None)
Creates a new branch from here and switches to it.

Public Attributes

- `label`

1.328.1 Detailed Description

Subversion version control system adapter.

1.328.2 Constructor & Destructor Documentation

1.328.2.1 `__init__()`

```
def anise.features.versioncontrol.svn.SubversionVersionControlSystem.__init__ (
    self )
```

1.328.3 Member Function Documentation

1.328.3.1 `addchange()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.addchange (
    self,
    changetext = None ) [inherited]
```

Adds a change to the changelog.

Parameters

<code>changetext</code>	The changelog entry text.
-------------------------	---------------------------

1.328.3.2 addlabel()

```
def anise.features.versioncontrol.internals.VersionControlSystem.addlabel (
    self,
    label = None ) [inherited]
```

Adds a version label for the current state.

Parameters

<i>label</i>	The label string.
--------------	-------------------

1.328.3.3 bindvcs()

```
def anise.features.versioncontrol.svn.SubversionVersionControlSystem.bindvcs (
    self,
    l )
```

1.328.3.4 branchfromhere()

```
def anise.features.versioncontrol.internals.VersionControlSystem.branchfromhere (
    self,
    branchname = None ) [inherited]
```

Creates a new branch from here and switches to it.

Unpushed changes will become part of that new branch.

Override this method in custom subclasses.

Parameters

<i>branchname</i>	The branch name (None: input dialog).
-------------------	---------------------------------------

1.328.3.5 createrawpackage()

```
def anise.features.versioncontrol.svn.SubversionVersionControlSystem.createrawpackage (
    self )
```

1.328.3.6 fetchrawcommitmessage()

```
def anise.features.versioncontrol.svn.SubversionVersionControlSystem.fetchrawcommitmessage (
    self,
    revision )
```

1.328.3.7 findvcs()

```
def anise.features.versioncontrol.svn.SubversionVersionControlSystem.findvcs (
    self )
```

1.328.3.8 getchangelog()

```
def anise.features.versioncontrol.internals.VersionControlSystem.getchangelog (
    self,
    asstring = True ) [inherited]
```

Computes the project change log from information in specially formatted vcs commit messages.

Parameters

<i>asstring</i>	If to output the changelog as string (instead of an internal data structure).
-----------------	---

1.328.3.9 getcommitmessage()

```
def anise.features.versioncontrol.internals.VersionControlSystem.getcommitmessage (
    self,
    revision ) [inherited]
```

Returns the raw commit message (as formatted by the vcs tool) for a revision.

Parameters

<i>revision</i>	The revision.
-----------------	---------------

1.328.3.10 getcommitmessagefromrawcommitmessage()

```
def anise.features.versioncontrol.svn.SubversionVersionControlSystem.getcommitmessagefromrawcommitmessage (
```

```
self,  
s )
```

1.328.3.11 getfullversion()

```
def anise.features.versioncontrol.svn.SubversionVersionControlSystem.getfullversion (  
    self )
```

1.328.3.12 getheadrevision()

```
def anise.features.versioncontrol.svn.SubversionVersionControlSystem.getheadrevision (  
    self )
```

1.328.3.13 getpreviousrevision()

```
def anise.features.versioncontrol.internals.VersionControlSystem.getpreviousrevision (  
    self,  
    revision ) [inherited]
```

Gets the predecessor revision for a given revision.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>revision</i>	A revision.
-----------------	-------------

Returns

: The predecassing revision.

1.328.3.14 storechange()

```
def anise.features.versioncontrol.internals.VersionControlSystem.storechange (  
    self,  
    changetext ) [inherited]
```

Stores a new entry to the changelog.

Do not use this directly, but addchange instead.

Parameters

<i>changetext</i>	The changelog entry text.
-------------------	---------------------------

1.328.3.15 storelabel()

```
def anise.features.versioncontrol.internals.VersionControlSystem.storelabel (
    self,
    label ) [inherited]
```

Stores a new version label.

Do not use this directly, but addlabel instead.

Parameters

<i>label</i>	The label string.
--------------	-------------------

1.328.3.16 syncversioncontrolsystem()

```
def anise.features.versioncontrol.svn.SubversionVersionControlSystem.syncversioncontrolsystem
(
    self,
    commitmessage = "",
    forceskipped = True,
    forceunchanged = False )
```

1.328.4 Member Data Documentation

1.328.4.1 label

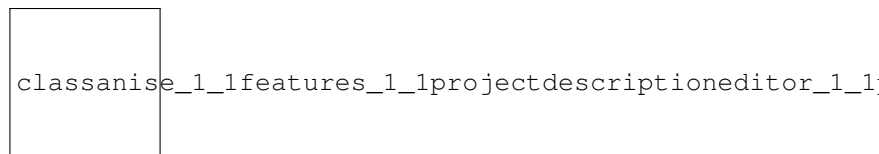
```
anise.features.versioncontrol.internals.VersionControlSystem.label [inherited]
```

The documentation for this class was generated from the following file:

- [anise/features/versioncontrol/svn.py](#)

1.329 anise.features.projectdescriptioneditor.packages.TarPackage Class Reference

Inheritance diagram for anise.features.projectdescriptioneditor.packages.TarPackage:



Public Member Functions

- def [__init__](#) (self)
- def [ask](#) (self)
- def [gettaskref](#) (self)

Public Attributes

- [label](#)
- [taskref](#)

1.329.1 Constructor & Destructor Documentation

1.329.1.1 [__init__\(\)](#)

```
def anise.features.projectdescriptioneditor.packages.TarPackage.__init__ (
    self )
```

1.329.2 Member Function Documentation

1.329.2.1 [ask\(\)](#)

```
def anise.features.projectdescriptioneditor.packages.TarPackage.ask (
    self )
```

1.329.2.2 [gettaskref\(\)](#)

```
def anise.features.projectdescriptioneditor.packages.AbstractPackage.gettaskref (
    self ) [inherited]
```

1.329.3 Member Data Documentation

1.329.3.1 label

`anise.features.projectdescriptioneditor.packages.TarPackage.label`

1.329.3.2 taskref

`anise.features.projectdescriptioneditor.packages.TarPackage.taskref`

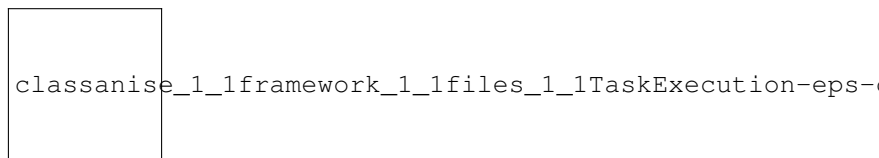
The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/packages.py](#)

1.330 anise.framework.files.TaskExecution Class Reference

Holds a function and some parameters for calling later on (like a delegate; or a more explicit, specialized and 'xml-compatible' version of a lambda) in order to get a [Filestructure](#) as function result.

Inheritance diagram for `anise.framework.files.TaskExecution`:



Public Member Functions

- `def __init__ (self, task, kwargs)`
- `def datakeys (self)`
Returns a list of keys for all stored metadata properties.
- `def setdata (self, k, v)`
Sets a metadata property.
- `def getdata (self, k, deflt)`
Gets a metadata property.
- `def initialize (self)`
Initializes this [Filestructure](#).
- `def path (self)`
The path to this structure in the filesystem.
- `def dl (self, subpath="", to=None, progresscallback=None)`
Copies the complete filestructure or a subdirectory to a new destination.
- `def with_modified_rootname (self, newname)`
Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.
- `def mv (self, newname)`
Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.
- `def setautoopen (self, path, interminal=False)`
Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Public Attributes

- [task](#)
- [params](#)
- [inner](#)

Private Member Functions

- `def _initialize (self)`

Private Attributes

- [_path](#)
- [_data](#)

1.330.1 Detailed Description

Holds a function and some parameters for calling later on (like a delegate; or a more explicit, specialized and 'xml-compatible' version of a lambda) in order to get a [Filestructure](#) as function result.

It acts as a proxy on top of this [Filestructure](#).

1.330.2 Constructor & Destructor Documentation

1.330.2.1 `__init__()`

```
def anise.framework.files.TaskExecution.__init__ (
    self,
    task,
    kwargs )
```

Parameters

<i>task</i>	The task implementation for execution.
-------------	--

1.330.3 Member Function Documentation

1.330.3.1 `_initialize()`

```
def anise.framework.files.TaskExecution._initialize (
    self ) [private]
```

1.330.3.2 datakeys()

```
def anise.framework.files.Filestructure.datakeys (
    self ) [inherited]
```

Returns a list of keys for all stored metadata properties.

1.330.3.3 dl()

```
def anise.framework.files.Filestructure.dl (
    self,
    subpath = "",
    to = None,
    progresscallback = None ) [inherited]
```

Copies the complete filestructure or a subdirectory to a new destination.

Parameters

<i>subpath</i>	If given, only this subdirectory will be copied.
<i>to</i>	If given, it specifies the destination path; if not given, a new directory in the current working directory will be used.
<i>progresscallback</i>	Callback function for retrieving progress information.

1.330.3.4 getdata()

```
def anise.framework.files.Filestructure.getdata (
    self,
    k,
    deflt ) [inherited]
```

Gets a metadata property.

Parameters

<i>k</i>	The key name.
<i>deflt</i>	The default value (returned if the key doesn't exist).

1.330.3.5 initialize()

```
def anise.framework.files.Filestructure.initialize (
    self ) [inherited]
```


Initializes this [Filestructure](#).

When this is called, the dynamic content (if any) is created and written to disc. It is automatically called for typical usage and is not required to be called manually.

1.330.3.6 mv()

```
def anise.framework.files.Filestructure.mv (
    self,
    newname ) [inherited]
```

Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.

Parameters

<i>newname</i>	The new name (absolute or relative).
----------------	--------------------------------------

1.330.3.7 path()

```
def anise.framework.files.Filestructure.path (
    self ) [inherited]
```

The path to this structure in the filesystem.

1.330.3.8 setautoopen()

```
def anise.framework.files.Filestructure.setautoopen (
    self,
    path,
    interminal = False ) [inherited]
```

Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Parameters

<i>path</i>	The relative path to the file which should automatically be opened.
<i>interminal</i>	If this hint is for terminal (non-graphical) usage.

1.330.3.9 setdata()

```
def anise.framework.files.Filestructure.setdata (
    self,
```

```
        k,  
        v ) [inherited]
```

Sets a metadata property.

Parameters

<i>k</i>	The key name.
<i>v</i>	The value.

1.330.3.10 with_modified_rootname()

```
def anise.framework.files.Filestructure.with_modified_rootname (  
    self,  
    newname ) [inherited]
```

Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.

Parameters

<i>newname</i>	The name which the new root directory should get.
----------------	---

1.330.4 Member Data Documentation

1.330.4.1 _data

```
anise.framework.files.TaskExecution._data [private]
```

1.330.4.2 _path

```
anise.framework.files.TaskExecution._path [private]
```

1.330.4.3 inner

```
anise.framework.files.TaskExecution.inner
```

1.330.4.4 params

```
anise.framework.files.TaskExecution.params
```

1.330.4.5 task

```
anise.framework.files.TaskExecution.task
```

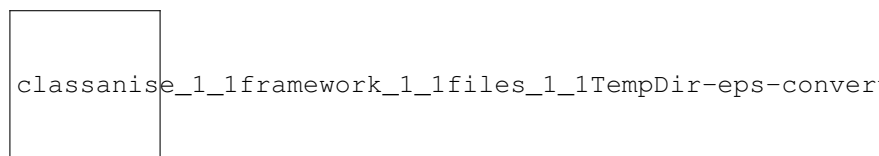
The documentation for this class was generated from the following file:

- [anise/framework/files.py](#)

1.331 anise.framework.files.TempDir Class Reference

A special [anise.framework.files.Filestructure](#) which automatically creates a temporary directory in background.

Inheritance diagram for `anise.framework.files.TempDir`:



Public Member Functions

- `def __init__ (self, dirname=None, temprootpath=None, namepattern="anise.{pid}.{i}")`
- `def __enter__ (self)`
- `def delete (self)`
Deletes the content from the hard drive.
- `def __exit__ (self, etype, evalue, etraceback)`
- `def datakeys (self)`
Returns a list of keys for all stored metadata properties.
- `def setdata (self, k, v)`
Sets a metadata property.
- `def getdata (self, k, deflt)`
Gets a metadata property.
- `def initialize (self)`
Initializes this [Filestructure](#).
- `def path (self)`
The path to this structure in the filesystem.
- `def dl (self, subpath="", to=None, progresscallback=None)`
Copies the complete filestructure or a subdirectory to a new destination.
- `def with_modified_rootname (self, newname)`
Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.
- `def mv (self, newname)`
Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.
- `def setautoopen (self, path, interminal=False)`
Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Public Attributes

- [origpath](#)

1.331.1 Detailed Description

A special [anise.framework.files.Filestructure](#) which automatically creates a temporary directory in background.

Example:

```
def sometask():
    mytmp = files.TempDir() # creates a temporary directory somewhere in no-man's-land
    myfile = mytmp.path() + "/somefile"
    with open(myfile, "w") as f:
        f.write("some content for our new file")
    return mytmp
```

1.331.2 Constructor & Destructor Documentation

1.331.2.1 `__init__()`

```
def anise.framework.files.TempDir.__init__ (
    self,
    dirname = None,
    temprootpath = None,
    namepattern = "anise.{pid}.{i}" )
```

Parameters

<i>dirname</i>	An optional name which the root directory should get; if not given, it gets a random name.
<i>temprootpath</i>	An optional alternate temp root directory path.
<i>namepattern</i>	The directory name pattern.

1.331.3 Member Function Documentation

1.331.3.1 `__enter__()`

```
def anise.framework.files.TempDir.__enter__ (
    self )
```

1.331.3.2 `__exit__()`

```
def anise.framework.files.TempDir.__exit__ (
    self,
    etype,
    evalue,
    etraceback )
```

1.331.3.3 `datakeys()`

```
def anise.framework.files.Filestructure.datakeys (
    self ) [inherited]
```

Returns a list of keys for all stored metadata properties.

1.331.3.4 `delete()`

```
def anise.framework.files.TempDir.delete (
    self )
```

Deletes the content from the hard drive.

1.331.3.5 `dl()`

```
def anise.framework.files.Filestructure.dl (
    self,
    subpath = "",
    to = None,
    progresscallback = None ) [inherited]
```

Copies the complete filestructure or a subdirectory to a new destination.

Parameters

<i>subpath</i>	If given, only this subdirectory will be copied.
<i>to</i>	If given, it specifies the destination path; if not given, a new directory in the current working directory will be used.
<i>progresscallback</i>	Callback function for retrieving progress information.

1.331.3.6 `getdata()`

```
def anise.framework.files.Filestructure.getdata (
```

```

        self,
        k,
        default ) [inherited]

```

Gets a metadata property.

Parameters

<i>k</i>	The key name.
<i>default</i>	The default value (returned if the key doesn't exist).

1.331.3.7 initialize()

```

def anise.framework.files.Filestructure.initialize (
    self ) [inherited]

```

Initializes this [Filestructure](#).

When this is called, the dynamic content (if any) is created and written to disc. It is automatically called for typical usage and is not required to be called manually.

1.331.3.8 mv()

```

def anise.framework.files.Filestructure.mv (
    self,
    newname ) [inherited]

```

Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.

Parameters

<i>newname</i>	The new name (absolute or relative).
----------------	--------------------------------------

1.331.3.9 path()

```

def anise.framework.files.Filestructure.path (
    self ) [inherited]

```

The path to this structure in the filesystem.

1.331.3.10 setautoopen()

```
def anise.framework.files.Filestructure.setautoopen (
    self,
    path,
    interterminal = False ) [inherited]
```

Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Parameters

<i>path</i>	The relative path to the file which should automatically be opened.
<i>interterminal</i>	If this hint is for terminal (non-graphical) usage.

1.331.3.11 setdata()

```
def anise.framework.files.Filestructure.setdata (
    self,
    k,
    v ) [inherited]
```

Sets a metadata property.

Parameters

<i>k</i>	The key name.
<i>v</i>	The value.

1.331.3.12 with_modified_rootname()

```
def anise.framework.files.Filestructure.with_modified_rootname (
    self,
    newname ) [inherited]
```

Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.

Parameters

<i>newname</i>	The name which the new root directory should get.
----------------	---

1.331.4 Member Data Documentation

1.331.4.1 origpath

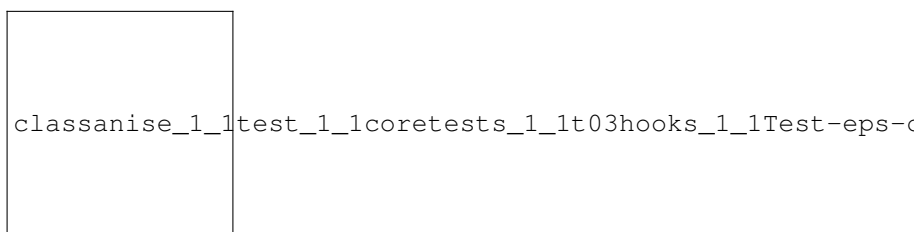
```
anise.framework.files.TempDir.origpath
```

The documentation for this class was generated from the following file:

- [anise/framework/files.py](#)

1.332 anise.test.coretests.t03hooks.Test Class Reference

Inheritance diagram for anise.test.coretests.t03hooks.Test:



Public Member Functions

- `def __init__ (self, a, kwa)`
- `def test_a (self)`
- `def setUp (self)`
- `def runanise (self, taskname, values=None, loadfeaturesfrom=None)`
- `def tearDown (self)`

Public Attributes

- [projectdesc](#)
- [rootdir](#)
- [packagestoredir](#)
- [mydir](#)
- [projdescfile](#)

1.332.1 Constructor & Destructor Documentation

1.332.1.1 `__init__()`

```
def anise.test.coretests.t03hooks.Test.__init__ (
    self,
    a,
    kwa )
```


1.332.2 Member Function Documentation

1.332.2.1 runanise()

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.332.2.2 setUp()

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.332.2.3 tearDown()

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.332.2.4 test_a()

```
def anise.test.coretests.t03hooks.Test.test_a (
    self )
```

1.332.3 Member Data Documentation

1.332.3.1 mydir

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.332.3.2 packagestoredir

```
anise.test.anisetest.AniseTestCase.packagestoredir [inherited]
```

1.332.3.3 projdescfile

```
anise.test.anisetest.AniseTestCase.projdescfile [inherited]
```

1.332.3.4 projectdesc

```
anise.test.anisetest.AniseTestCase.projectdesc [inherited]
```

1.332.3.5 rootdir

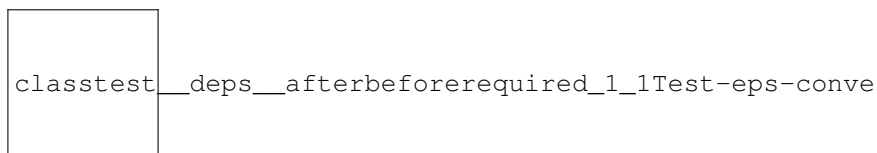
```
anise.test.anisetest.AniseTestCase.rootdir [inherited]
```

The documentation for this class was generated from the following file:

- anise/test/coretests/[t03hooks.py](#)

1.333 test_deps_afterbeforerequired.Test Class Reference

Inheritance diagram for test_deps_afterbeforerequired.Test:



Public Member Functions

- def [test_a](#) (self)

1.333.1 Member Function Documentation

1.333.1.1 test_a()

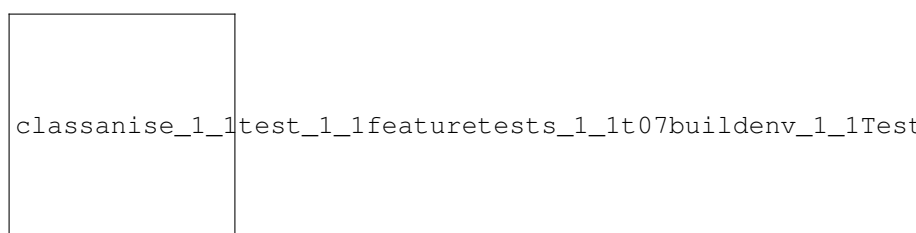
```
def test_deps_afterbeforerequired.Test.test_a (
    self )
```

The documentation for this class was generated from the following file:

- [anise/data/PyDepsEngine/_meta/test_deps_afterbeforerequired.py](#)

1.334 anise.test.featuretests.t07buildenv.Test Class Reference

Inheritance diagram for anise.test.featuretests.t07buildenv.Test:



Public Member Functions

- def `__init__` (self, a, kwa)
- def [test_a](#) (self)
- def [setUp](#) (self)
- def [runanise](#) (self, taskname, values=None, loadfeaturesfrom=None)
- def [tearDown](#) (self)

Public Attributes

- [destination](#)
- [projectdesc](#)
- [rootdir](#)
- [packagestoredir](#)
- [mydir](#)
- [projdescfile](#)

1.334.1 Constructor & Destructor Documentation

1.334.1.1 __init__()

```
def anise.test.featuretests.t07buildenv.Test.__init__ (
    self,
    a,
    kwa )
```

1.334.2 Member Function Documentation

1.334.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.334.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.334.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.334.2.4 `test_a()`

```
def anise.test.featuretests.t07buildenv.Test.test_a (
    self )
```

1.334.3 Member Data Documentation

1.334.3.1 `destination`

```
anise.test.featuretests.t07buildenv.Test.destination
```

1.334.3.2 mydir

`anise.test.anisetest.AniseTestCase.mydir` [inherited]

1.334.3.3 packagestoredir

`anise.test.anisetest.AniseTestCase.packagestoredir` [inherited]

1.334.3.4 projdescfile

`anise.test.anisetest.AniseTestCase.projdescfile` [inherited]

1.334.3.5 projectdesc

`anise.test.anisetest.AniseTestCase.projectdesc` [inherited]

1.334.3.6 rootdir

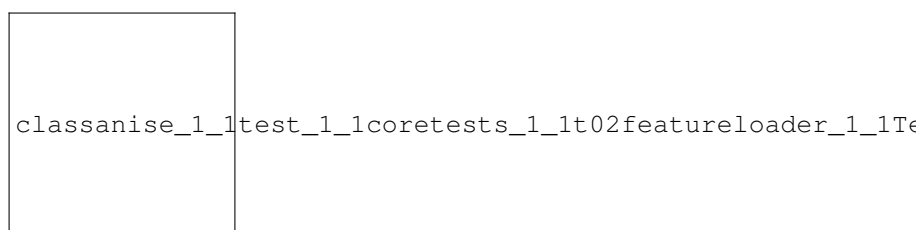
`anise.test.anisetest.AniseTestCase.rootdir` [inherited]

The documentation for this class was generated from the following file:

- `anise/test/featuretests/t07buildenv.py`

1.335 anise.test.coretests.t02featureloader.Test Class Reference

Inheritance diagram for `anise.test.coretests.t02featureloader.Test`:



Public Member Functions

- `def __init__ (self, a, kwa)`
- `def test_a (self)`
- `def setUp (self)`
- `def runanise (self, taskname, values=None, loadfeaturesfrom=None)`
- `def tearDown (self)`

Public Attributes

- `projectdesc`
- `rootdir`
- `packagestoredir`
- `mydir`
- `projdescfile`

1.335.1 Constructor & Destructor Documentation

1.335.1.1 __init__()

```
def anise.test.coretests.t02featureloader.Test.__init__ (  
    self,  
    a,  
    kwa )
```

1.335.2 Member Function Documentation

1.335.2.1 runanise()

```
def anise.test.anisetest.AniseTestCase.runanise (  
    self,  
    taskname,  
    values = None,  
    loadfeaturesfrom = None ) [inherited]
```

1.335.2.2 setUp()

```
def anise.test.anisetest.AniseTestCase.setUp (  
    self ) [inherited]
```

1.335.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.335.2.4 `test_a()`

```
def anise.test.coretests.t02featureloader.Test.test_a (
    self )
```

1.335.3 Member Data Documentation

1.335.3.1 `mydir`

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.335.3.2 `packagestoredir`

```
anise.test.anisetest.AniseTestCase.packagestoredir [inherited]
```

1.335.3.3 `projdescfile`

```
anise.test.anisetest.AniseTestCase.projdescfile [inherited]
```

1.335.3.4 `projectdesc`

```
anise.test.anisetest.AniseTestCase.projectdesc [inherited]
```

1.335.3.5 `rootdir`

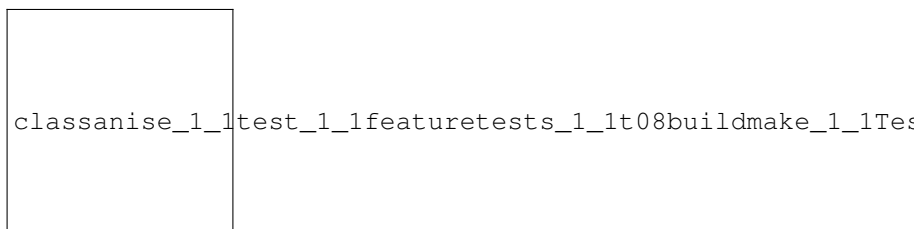
```
anise.test.anisetest.AniseTestCase.rootdir [inherited]
```

The documentation for this class was generated from the following file:

- [anise/test/coretests/t02featureloader.py](#)

1.336 anise.test.featuretests.t08buildmake.Test Class Reference

Inheritance diagram for anise.test.featuretests.t08buildmake.Test:



Public Member Functions

- def `__init__` (self, a, kwa)
- def `test_a` (self)
- def `setUp` (self)
- def `runanise` (self, taskname, values=None, loadfeaturesfrom=None)
- def `tearDown` (self)

Public Attributes

- `destination`
- `projectdesc`
- `rootdir`
- `packagestoredir`
- `mydir`
- `projdescfile`

1.336.1 Constructor & Destructor Documentation

1.336.1.1 `__init__()`

```
def anise.test.featuretests.t08buildmake.Test.__init__ (
    self,
    a,
    kwa )
```

1.336.2 Member Function Documentation

1.336.2.1 runanise()

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.336.2.2 setUp()

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.336.2.3 tearDown()

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.336.2.4 test_a()

```
def anise.test.featuretests.t08buildmake.Test.test_a (
    self )
```

1.336.3 Member Data Documentation

1.336.3.1 destination

```
anise.test.featuretests.t08buildmake.Test.destination
```

1.336.3.2 mydir

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.336.3.3 packagestoredir

`anise.test.anisetest.AniseTestCase.packagestoredir` [inherited]

1.336.3.4 projdescfile

`anise.test.anisetest.AniseTestCase.projdescfile` [inherited]

1.336.3.5 projectdesc

`anise.test.anisetest.AniseTestCase.projectdesc` [inherited]

1.336.3.6 rootdir

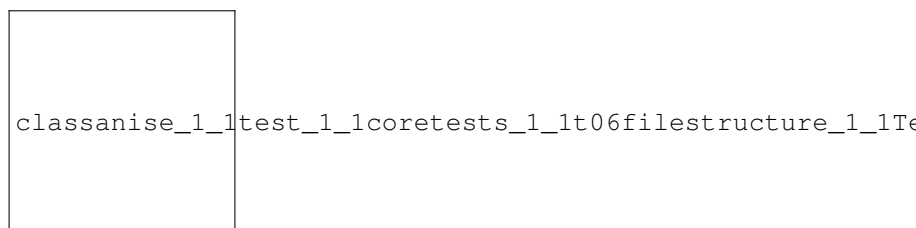
`anise.test.anisetest.AniseTestCase.rootdir` [inherited]

The documentation for this class was generated from the following file:

- [anise/test/featuretests/t08buildmake.py](#)

1.337 anise.test.coretests.t06filestructure.Test Class Reference

Inheritance diagram for `anise.test.coretests.t06filestructure.Test`:



Public Member Functions

- `def __init__ (self, a, kwa)`
- `def test_a (self)`
- `def setUp (self)`
- `def runanise (self, taskname, values=None, loadfeaturesfrom=None)`
- `def tearDown (self)`

Public Attributes

- [projectdesc](#)
- [rootdir](#)
- [packagestoredir](#)
- [mydir](#)
- [projdescfile](#)

1.337.1 Constructor & Destructor Documentation

1.337.1.1 `__init__()`

```
def anise.test.coretests.t06filestructure.Test.__init__ (
    self,
    a,
    kwa )
```

1.337.2 Member Function Documentation

1.337.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.337.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.337.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.337.2.4 test_a()

```
def anise.test.coretests.t06filestructure.Test.test_a (
    self )
```

1.337.3 Member Data Documentation

1.337.3.1 mydir

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.337.3.2 packagestoredir

```
anise.test.anisetest.AniseTestCase.packagestoredir [inherited]
```

1.337.3.3 projdescfile

```
anise.test.anisetest.AniseTestCase.projdescfile [inherited]
```

1.337.3.4 projectdesc

```
anise.test.anisetest.AniseTestCase.projectdesc [inherited]
```

1.337.3.5 rootdir

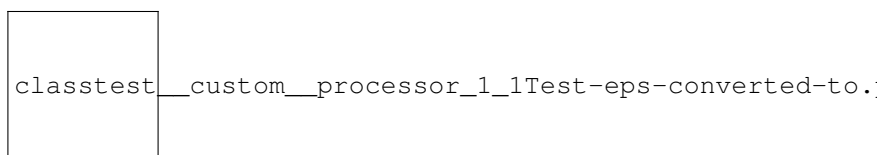
```
anise.test.anisetest.AniseTestCase.rootdir [inherited]
```

The documentation for this class was generated from the following file:

- [anise/test/coretests/t06filestructure.py](#)

1.338 test_custom_processor.Test Class Reference

Inheritance diagram for test_custom_processor.Test:



Public Member Functions

- def [test_a](#) (self)

1.338.1 Member Function Documentation

1.338.1.1 test_a()

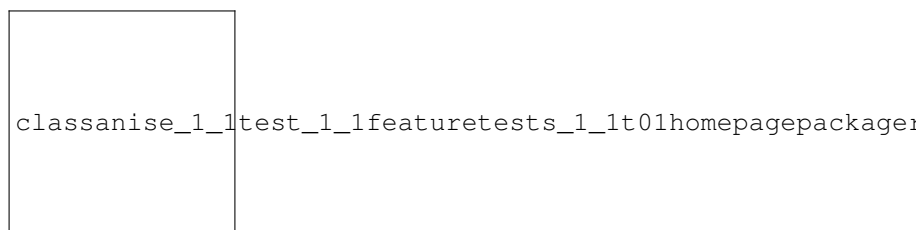
```
def test_custom_processor.Test.test_a (
    self )
```

The documentation for this class was generated from the following file:

- anise/data/PyDepsEngine/_meta/[test_custom_processor.py](#)

1.339 anise.test.featuretests.t01homepagepackagerelease.Test Class Reference

Inheritance diagram for anise.test.featuretests.t01homepagepackagerelease.Test:



Public Member Functions

- def [__init__](#) (self, a, kwa)
- def [test_a](#) (self)
- def [setUp](#) (self)
- def [runanise](#) (self, taskname, values=None, loadfeaturesfrom=None)
- def [tearDown](#) (self)

Public Attributes

- [destination](#)
- [projectdesc](#)
- [rootdir](#)
- [packagestoredir](#)
- [mydir](#)
- [projdescfile](#)

1.339.1 Constructor & Destructor Documentation

1.339.1.1 `__init__()`

```
def anise.test.featuretests.t01homepagepackagerelease.Test.__init__ (
    self,
    a,
    kwa )
```

1.339.2 Member Function Documentation

1.339.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.339.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.339.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.339.2.4 test_a()

```
def anise.test.featuretests.t01homepagepackagerelease.Test.test_a (
    self )
```

1.339.3 Member Data Documentation

1.339.3.1 destination

```
anise.test.featuretests.t01homepagepackagerelease.Test.destination
```

1.339.3.2 mydir

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.339.3.3 packagestoredir

```
anise.test.anisetest.AniseTestCase.packagestoredir [inherited]
```

1.339.3.4 projdescfile

```
anise.test.anisetest.AniseTestCase.projdescfile [inherited]
```

1.339.3.5 projectdesc

```
anise.test.anisetest.AniseTestCase.projectdesc [inherited]
```

1.339.3.6 rootdir

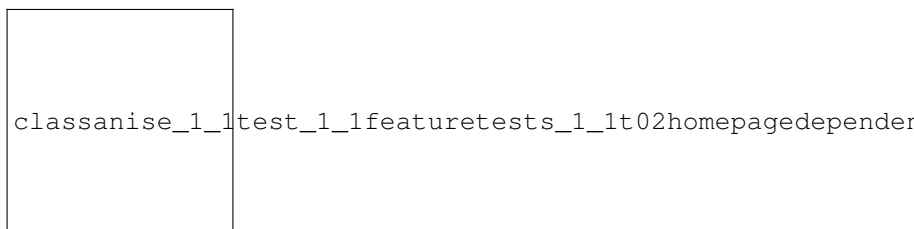
```
anise.test.anisetest.AniseTestCase.rootdir [inherited]
```

The documentation for this class was generated from the following file:

- [anise/test/featuretests/t01homepagepackagerelease.py](#)

1.340 anise.test.featuretests.t02homepagedependencies.Test Class Reference

Inheritance diagram for anise.test.featuretests.t02homepagedependencies.Test:



Public Member Functions

- def `__init__` (self, a, kwa)
- def `test_a` (self)
- def `setUp` (self)
- def `runanise` (self, taskname, values=None, loadfeaturesfrom=None)
- def `tearDown` (self)

Public Attributes

- `destination`
- `projectdesc`
- `rootdir`
- `packagestoredir`
- `mydir`
- `projdescfile`

1.340.1 Constructor & Destructor Documentation

1.340.1.1 `__init__()`

```

def anise.test.featuretests.t02homepagedependencies.Test.__init__ (
    self,
    a,
    kwa )
  
```

1.340.2 Member Function Documentation

1.340.2.1 runanise()

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.340.2.2 setUp()

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.340.2.3 tearDown()

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.340.2.4 test_a()

```
def anise.test.featuretests.t02homepagedependencies.Test.test_a (
    self )
```

1.340.3 Member Data Documentation

1.340.3.1 destination

```
anise.test.featuretests.t02homepagedependencies.Test.destination
```

1.340.3.2 mydir

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.340.3.3 packagestoredir

```
anise.test.anisetest.AniseTestCase.packagestoredir [inherited]
```

1.340.3.4 projdescfile

```
anise.test.anisetest.AniseTestCase.projdescfile [inherited]
```

1.340.3.5 projectdesc

```
anise.test.anisetest.AniseTestCase.projectdesc [inherited]
```

1.340.3.6 rootdir

```
anise.test.anisetest.AniseTestCase.rootdir [inherited]
```

The documentation for this class was generated from the following file:

- [anise/test/featuretests/t02homepagedependencies.py](#)

1.341 test_custom_processor_filterobjects_completelistbyrequirements.Test Class Reference

Inheritance diagram for test_custom_processor_filterobjects_completelistbyrequirements.Test:

**Public Member Functions**

- def [test_a](#) (self)

1.341.1 Member Function Documentation

1.341.1.1 test_a()

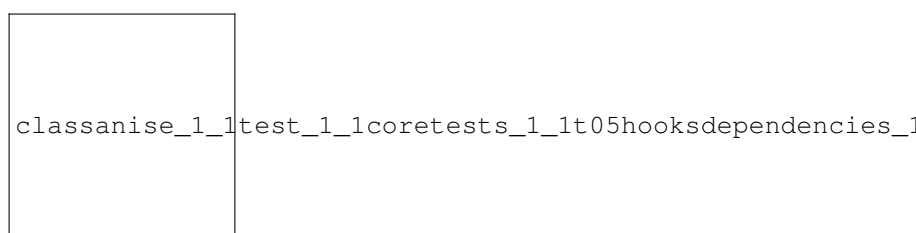
```
def test_custom_processor_filterobjects_completelistbyrequirements.Test.test_a (
    self )
```

The documentation for this class was generated from the following file:

- anise/data/PyDepsEngine/_meta/[test_custom_processor_filterobjects_completelistbyrequirements.py](#)

1.342 anise.test.coretests.t05hooksdependencies.Test Class Reference

Inheritance diagram for anise.test.coretests.t05hooksdependencies.Test:



Public Member Functions

- def [__init__](#) (self, a, kwa)
- def [test_a](#) (self)
- def [setUp](#) (self)
- def [runanise](#) (self, taskname, values=None, loadfeaturesfrom=None)
- def [tearDown](#) (self)

Public Attributes

- [projectdesc](#)
- [rootdir](#)
- [packagestoredir](#)
- [mydir](#)
- [projdescfile](#)

1.342.1 Constructor & Destructor Documentation

1.342.1.1 __init__()

```
def anise.test.coretests.t05hooksdependencies.Test.__init__ (
    self,
    a,
    kwa )
```

1.342.2 Member Function Documentation

1.342.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.342.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.342.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.342.2.4 `test_a()`

```
def anise.test.coretests.t05hooksdependencies.Test.test_a (
    self )
```

1.342.3 Member Data Documentation

1.342.3.1 `mydir`

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.342.3.2 packagestoredir

`anise.test.anisetest.AniseTestCase.packagestoredir` [inherited]

1.342.3.3 projdescfile

`anise.test.anisetest.AniseTestCase.projdescfile` [inherited]

1.342.3.4 projectdesc

`anise.test.anisetest.AniseTestCase.projectdesc` [inherited]

1.342.3.5 rootdir

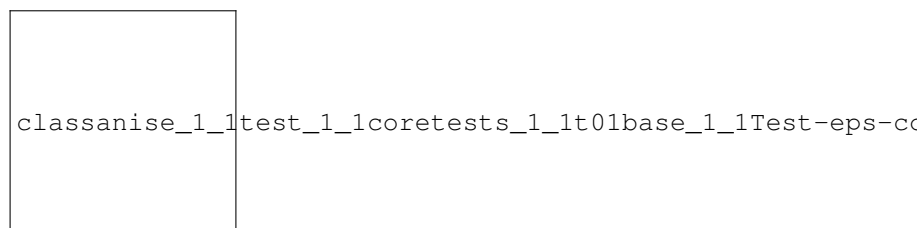
`anise.test.anisetest.AniseTestCase.rootdir` [inherited]

The documentation for this class was generated from the following file:

- [anise/test/coretests/t05hooksdependencies.py](#)

1.343 anise.test.coretests.t01base.Test Class Reference

Inheritance diagram for `anise.test.coretests.t01base.Test`:



Public Member Functions

- `def __init__ (self, a, kwa)`
- `def test_a (self)`
- `def setUp (self)`
- `def runanise (self, taskname, values=None, loadfeaturesfrom=None)`
- `def tearDown (self)`

Public Attributes

- [projectdesc](#)
- [rootdir](#)
- [packagestoredir](#)
- [mydir](#)
- [projdescfile](#)

1.343.1 Constructor & Destructor Documentation

1.343.1.1 `__init__()`

```
def anise.test.coretests.t01base.Test.__init__ (
    self,
    a,
    kwa )
```

1.343.2 Member Function Documentation

1.343.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.343.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.343.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.343.2.4 test_a()

```
def anise.test.coretests.t01base.Test.test_a (
    self )
```

1.343.3 Member Data Documentation

1.343.3.1 mydir

anise.test.anisetest.AniseTestCase.mydir [inherited]

1.343.3.2 packagestoredir

anise.test.anisetest.AniseTestCase.packagestoredir [inherited]

1.343.3.3 projdescfile

anise.test.anisetest.AniseTestCase.projdescfile [inherited]

1.343.3.4 projectdesc

anise.test.anisetest.AniseTestCase.projectdesc [inherited]

1.343.3.5 rootdir

anise.test.anisetest.AniseTestCase.rootdir [inherited]

The documentation for this class was generated from the following file:

- [anise/test/coretests/t01base.py](#)

1.344 test_custom_processor_filterobjects.Test Class Reference

Inheritance diagram for test_custom_processor_filterobjects.Test:



Public Member Functions

- def [test_a](#) (self)

1.344.1 Member Function Documentation

1.344.1.1 test_a()

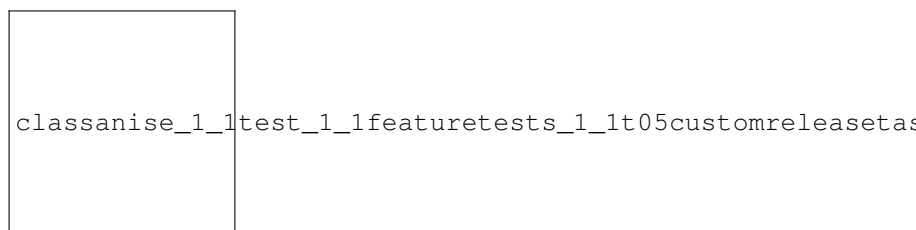
```
def test_custom_processor_filterobjects.Test.test_a (
    self )
```

The documentation for this class was generated from the following file:

- anise/data/PyDepsEngine/_meta/[test_custom_processor_filterobjects.py](#)

1.345 anise.test.featuretests.t05customreleasetask.Test Class Reference

Inheritance diagram for anise.test.featuretests.t05customreleasetask.Test:



Public Member Functions

- def [__init__](#) (self, a, kwa)
- def [test_a](#) (self)
- def [setUp](#) (self)
- def [runanise](#) (self, taskname, values=None, loadfeaturesfrom=None)
- def [tearDown](#) (self)

Public Attributes

- [destination](#)
- [projectdesc](#)
- [rootdir](#)
- [packagestoredir](#)
- [mydir](#)
- [projdescfile](#)

1.345.1 Constructor & Destructor Documentation

1.345.1.1 `__init__()`

```
def anise.test.featuretests.t05customreleasetask.Test.__init__ (
    self,
    a,
    kwa )
```

1.345.2 Member Function Documentation

1.345.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.345.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.345.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.345.2.4 test_a()

```
def anise.test.featuretests.t05customreleasetask.Test.test_a (
    self )
```

1.345.3 Member Data Documentation

1.345.3.1 destination

```
anise.test.featuretests.t05customreleasetask.Test.destination
```

1.345.3.2 mydir

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.345.3.3 packagestoredir

```
anise.test.anisetest.AniseTestCase.packagestoredir [inherited]
```

1.345.3.4 projdescfile

```
anise.test.anisetest.AniseTestCase.projdescfile [inherited]
```

1.345.3.5 projectdesc

```
anise.test.anisetest.AniseTestCase.projectdesc [inherited]
```

1.345.3.6 rootdir

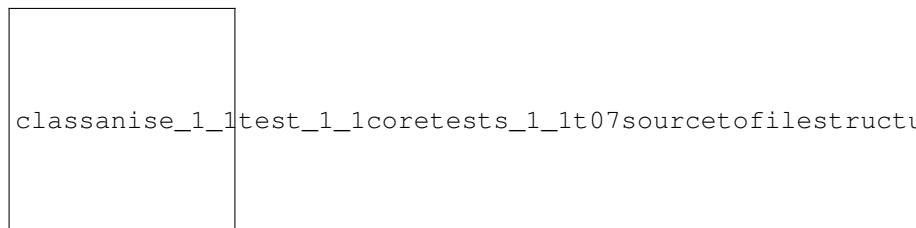
```
anise.test.anisetest.AniseTestCase.rootdir [inherited]
```

The documentation for this class was generated from the following file:

- [anise/test/featuretests/t05customreleasetask.py](#)

1.346 anise.test.coretests.t07sourcetofilestructure.Test Class Reference

Inheritance diagram for anise.test.coretests.t07sourcetofilestructure.Test:



Public Member Functions

- def [__init__](#) (self, a, kwa)
- def [test_a](#) (self)
- def [setUp](#) (self)
- def [runanise](#) (self, taskname, values=None, loadfeaturesfrom=None)
- def [tearDown](#) (self)

Public Attributes

- [projectdesc](#)
- [rootdir](#)
- [packagestoredir](#)
- [mydir](#)
- [projdescfile](#)

1.346.1 Constructor & Destructor Documentation

1.346.1.1 __init__()

```
def anise.test.coretests.t07sourcetofilestructure.Test.__init__ (
    self,
    a,
    kwa )
```

1.346.2 Member Function Documentation

1.346.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.346.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.346.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.346.2.4 `test_a()`

```
def anise.test.coretests.t07sourcetofilestructure.Test.test_a (
    self )
```

1.346.3 Member Data Documentation

1.346.3.1 `mydir`

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.346.3.2 `packagestoredir`

```
anise.test.anisetest.AniseTestCase.packagestoredir [inherited]
```

1.346.3.3 projdescfile

`anise.test.anisetest.AniseTestCase.projdescfile` [inherited]

1.346.3.4 projectdesc

`anise.test.anisetest.AniseTestCase.projectdesc` [inherited]

1.346.3.5 rootdir

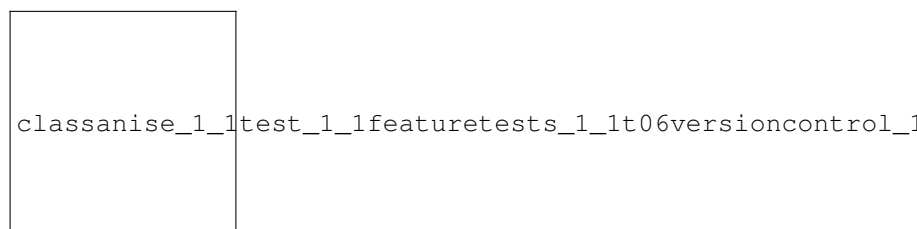
`anise.test.anisetest.AniseTestCase.rootdir` [inherited]

The documentation for this class was generated from the following file:

- [anise/test/coretests/t07sourcetofilestructure.py](#)

1.347 anise.test.featuretests.t06versioncontrol.Test Class Reference

Inheritance diagram for `anise.test.featuretests.t06versioncontrol.Test`:



Public Member Functions

- `def __init__ (self, a, kwa)`
- `def test_a (self)`
- `def setUp (self)`
- `def runanise (self, taskname, values=None, loadfeaturesfrom=None)`
- `def tearDown (self)`

Public Attributes

- [destination](#)
- [projectdesc](#)
- [rootdir](#)
- [packagestoredir](#)
- [mydir](#)
- [projdescfile](#)

1.347.1 Constructor & Destructor Documentation

1.347.1.1 `__init__()`

```
def anise.test.featuretests.t06versioncontrol.Test.__init__ (
    self,
    a,
    kwa )
```

1.347.2 Member Function Documentation

1.347.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.347.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.347.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.347.2.4 `test_a()`

```
def anise.test.featuretests.t06versioncontrol.Test.test_a (
    self )
```

1.347.3 Member Data Documentation

1.347.3.1 destination

`anise.test.featuretests.t06versioncontrol.Test.destination`

1.347.3.2 mydir

`anise.test.anisetest.AniseTestCase.mydir` [inherited]

1.347.3.3 packagestoredir

`anise.test.anisetest.AniseTestCase.packagestoredir` [inherited]

1.347.3.4 projdescfile

`anise.test.anisetest.AniseTestCase.projdescfile` [inherited]

1.347.3.5 projectdesc

`anise.test.anisetest.AniseTestCase.projectdesc` [inherited]

1.347.3.6 rootdir

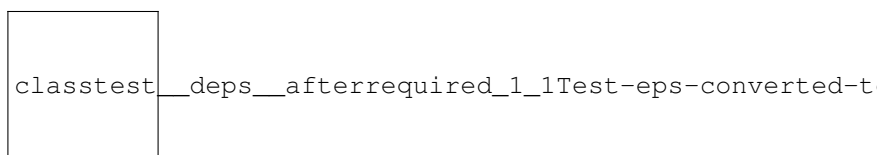
`anise.test.anisetest.AniseTestCase.rootdir` [inherited]

The documentation for this class was generated from the following file:

- [anise/test/featuretests/t06versioncontrol.py](#)

1.348 test_deps_afterrequired.Test Class Reference

Inheritance diagram for test_deps_afterrequired.Test:



Public Member Functions

- def [test_a](#) (self)

1.348.1 Member Function Documentation

1.348.1.1 test_a()

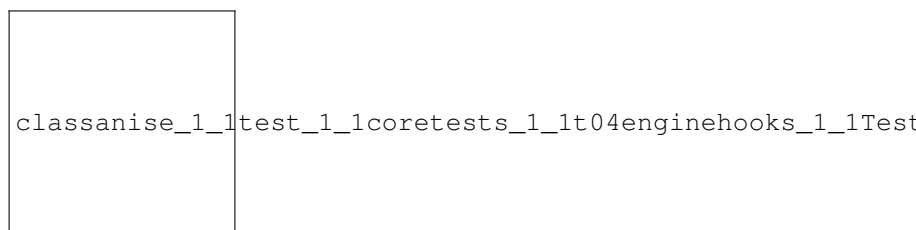
```
def test_deps_afterrequired.Test.test_a (
    self )
```

The documentation for this class was generated from the following file:

- anise/data/PyDepsEngine/_meta/[test_deps_afterrequired.py](#)

1.349 anise.test.coretests.t04enginehooks.Test Class Reference

Inheritance diagram for anise.test.coretests.t04enginehooks.Test:



Public Member Functions

- def [__init__](#) (self, a, kwa)
- def [test_a](#) (self)
- def [setUp](#) (self)
- def [runanise](#) (self, taskname, values=None, loadfeaturesfrom=None)
- def [tearDown](#) (self)

Public Attributes

- [projectdesc](#)
- [rootdir](#)
- [packagestoredir](#)
- [mydir](#)
- [projdescfile](#)

1.349.1 Constructor & Destructor Documentation

1.349.1.1 `__init__()`

```
def anise.test.coretests.t04enginehooks.Test.__init__ (
    self,
    a,
    kwa )
```

1.349.2 Member Function Documentation

1.349.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.349.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.349.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.349.2.4 test_a()

```
def anise.test.coretests.t04enginehooks.Test.test_a (
    self )
```

1.349.3 Member Data Documentation

1.349.3.1 mydir

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.349.3.2 packagestoredir

```
anise.test.anisetest.AniseTestCase.packagestoredir [inherited]
```

1.349.3.3 projdescfile

```
anise.test.anisetest.AniseTestCase.projdescfile [inherited]
```

1.349.3.4 projectdesc

```
anise.test.anisetest.AniseTestCase.projectdesc [inherited]
```

1.349.3.5 rootdir

```
anise.test.anisetest.AniseTestCase.rootdir [inherited]
```

The documentation for this class was generated from the following file:

- [anise/test/coretests/t04enginehooks.py](#)

1.350 anise.test.featuretests.t04packagestore.Test Class Reference

Inheritance diagram for anise.test.featuretests.t04packagestore.Test:



Public Member Functions

- `def __init__ (self, a, kwa)`
- `def test_a (self)`
- `def setUp (self)`
- `def runanise (self, taskname, values=None, loadfeaturesfrom=None)`
- `def tearDown (self)`

Public Attributes

- `destination`
- `projectdesc`
- `rootdir`
- `packagestoredir`
- `mydir`
- `projdescfile`

1.350.1 Constructor & Destructor Documentation

1.350.1.1 __init__()

```
def anise.test.featuretests.t04packagestore.Test.__init__ (
    self,
    a,
    kwa )
```

1.350.2 Member Function Documentation

1.350.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.350.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.350.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.350.2.4 `test_a()`

```
def anise.test.featuretests.t04packagestore.Test.test_a (
    self )
```

1.350.3 Member Data Documentation

1.350.3.1 `destination`

```
anise.test.featuretests.t04packagestore.Test.destination
```

1.350.3.2 `mydir`

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.350.3.3 packagestoredir

`anise.test.anisetest.AniseTestCase.packagestoredir` [inherited]

1.350.3.4 projdescfile

`anise.test.anisetest.AniseTestCase.projdescfile` [inherited]

1.350.3.5 projectdesc

`anise.test.anisetest.AniseTestCase.projectdesc` [inherited]

1.350.3.6 rootdir

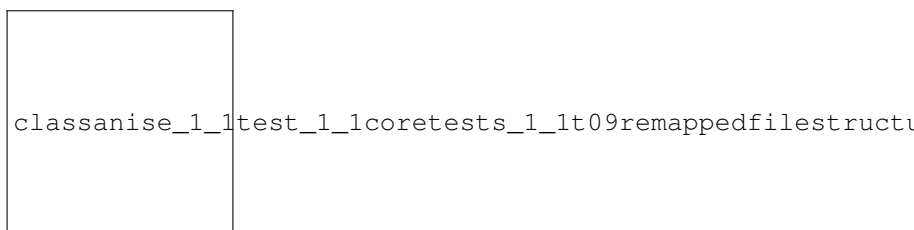
`anise.test.anisetest.AniseTestCase.rootdir` [inherited]

The documentation for this class was generated from the following file:

- [anise/test/featuretests/t04packagestore.py](#)

1.351 anise.test.coretests.t09remappedfilestructure.Test Class Reference

Inheritance diagram for `anise.test.coretests.t09remappedfilestructure.Test`:



Public Member Functions

- `def __init__ (self, a, kwa)`
- `def test_a (self)`
- `def setUp (self)`
- `def runanise (self, taskname, values=None, loadfeaturesfrom=None)`
- `def tearDown (self)`

Public Attributes

- [projectdesc](#)
- [rootdir](#)
- [packagestoredir](#)
- [mydir](#)
- [projdescfile](#)

1.351.1 Constructor & Destructor Documentation

1.351.1.1 `__init__()`

```
def anise.test.coretests.t09remappedfilestructure.Test.__init__ (
    self,
    a,
    kwa )
```

1.351.2 Member Function Documentation

1.351.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.351.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.351.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.351.2.4 test_a()

```
def anise.test.coretests.t09remappedfilestructure.Test.test_a (
    self )
```

1.351.3 Member Data Documentation

1.351.3.1 mydir

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.351.3.2 packagestoredir

```
anise.test.anisetest.AniseTestCase.packagestoredir [inherited]
```

1.351.3.3 projdescfile

```
anise.test.anisetest.AniseTestCase.projdescfile [inherited]
```

1.351.3.4 projectdesc

```
anise.test.anisetest.AniseTestCase.projectdesc [inherited]
```

1.351.3.5 rootdir

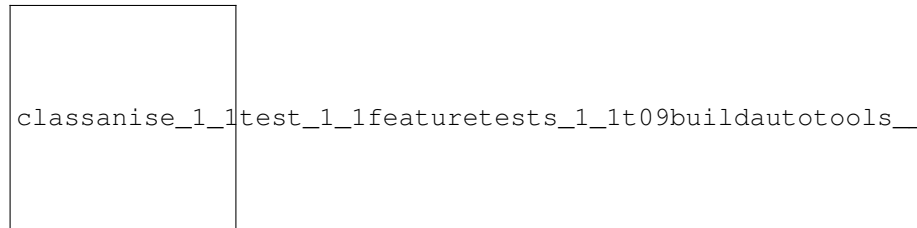
```
anise.test.anisetest.AniseTestCase.rootdir [inherited]
```

The documentation for this class was generated from the following file:

- [anise/test/coretests/t09remappedfilestructure.py](#)

1.352 anise.test.featuretests.t09buildautotools_simple.Test Class Reference

Inheritance diagram for anise.test.featuretests.t09buildautotools_simple.Test:



Public Member Functions

- `def __init__ (self, a, kwa)`
- `def test_a (self)`
- `def setUp (self)`
- `def runanise (self, taskname, values=None, loadfeaturesfrom=None)`
- `def tearDown (self)`

Public Attributes

- `destination`
- `projectdesc`
- `rootdir`
- `packagestoredir`
- `mydir`
- `projdescfile`

1.352.1 Constructor & Destructor Documentation

1.352.1.1 __init__()

```

def anise.test.featuretests.t09buildautotools_simple.Test.__init__ (
    self,
    a,
    kwa )

```

1.352.2 Member Function Documentation

1.352.2.1 runanise()

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.352.2.2 setUp()

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.352.2.3 tearDown()

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.352.2.4 test_a()

```
def anise.test.featuretests.t09buildautotools_simple.Test.test_a (
    self )
```

1.352.3 Member Data Documentation

1.352.3.1 destination

```
anise.test.featuretests.t09buildautotools_simple.Test.destination
```

1.352.3.2 mydir

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.352.3.3 packagestoredir

`anise.test.anisetest.AniseTestCase.packagestoredir` [inherited]

1.352.3.4 projdescfile

`anise.test.anisetest.AniseTestCase.projdescfile` [inherited]

1.352.3.5 projectdesc

`anise.test.anisetest.AniseTestCase.projectdesc` [inherited]

1.352.3.6 rootdir

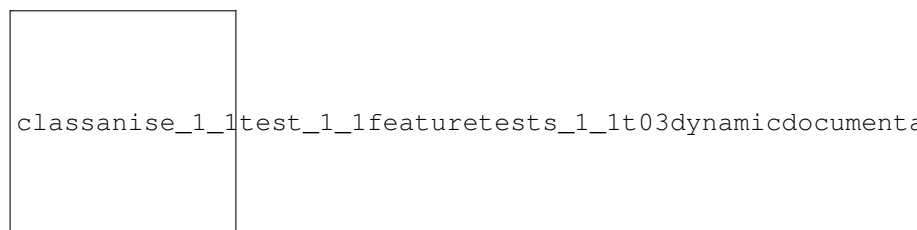
`anise.test.anisetest.AniseTestCase.rootdir` [inherited]

The documentation for this class was generated from the following file:

- [anise/test/featuretests/t09buildautotools_simple.py](#)

1.353 anise.test.featuretests.t03dynamicdocumentation.Test Class Reference

Inheritance diagram for `anise.test.featuretests.t03dynamicdocumentation.Test`:



Public Member Functions

- `def __init__ (self, a, kwa)`
- `def test_a (self)`
- `def setUp (self)`
- `def runanise (self, taskname, values=None, loadfeaturesfrom=None)`
- `def tearDown (self)`

Public Attributes

- [destination](#)
- [projectdesc](#)
- [rootdir](#)
- [packagestoredir](#)
- [mydir](#)
- [projdescfile](#)

1.353.1 Constructor & Destructor Documentation

1.353.1.1 `__init__()`

```
def anise.test.featuretests.t03dynamicdocumentation.Test.__init__ (
    self,
    a,
    kwa )
```

1.353.2 Member Function Documentation

1.353.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.353.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.353.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.353.2.4 test_a()

```
def anise.test.featuretests.t03dynamicdocumentation.Test.test_a (
    self )
```

1.353.3 Member Data Documentation

1.353.3.1 destination

```
anise.test.featuretests.t03dynamicdocumentation.Test.destination
```

1.353.3.2 mydir

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.353.3.3 packagestoredir

```
anise.test.anisetest.AniseTestCase.packagestoredir [inherited]
```

1.353.3.4 projdescfile

```
anise.test.anisetest.AniseTestCase.projdescfile [inherited]
```

1.353.3.5 projectdesc

```
anise.test.anisetest.AniseTestCase.projectdesc [inherited]
```

1.353.3.6 rootdir

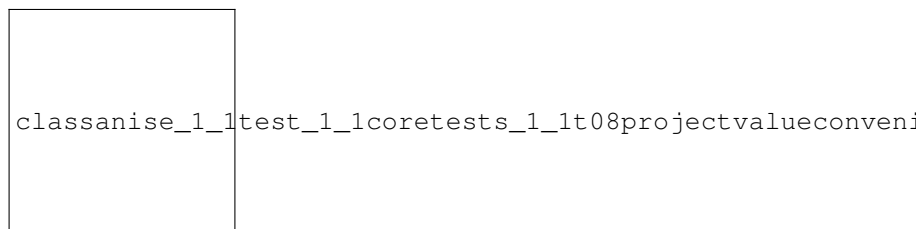
```
anise.test.anisetest.AniseTestCase.rootdir [inherited]
```

The documentation for this class was generated from the following file:

- [anise/test/featuretests/t03dynamicdocumentation.py](#)

1.354 anise.test.coretests.t08projectvalueconvenience.Test Class Reference

Inheritance diagram for anise.test.coretests.t08projectvalueconvenience.Test:



Public Member Functions

- def `__init__` (self, a, kwa)
- def `test_a` (self)
- def `test_b` (self)
- def `setUp` (self)
- def `runanise` (self, taskname, values=None, loadfeaturesfrom=None)
- def `tearDown` (self)

Public Attributes

- `projectdesc`
- `rootdir`
- `packagestoredir`
- `mydir`
- `projdescfile`

1.354.1 Constructor & Destructor Documentation

1.354.1.1 `__init__`()

```
def anise.test.coretests.t08projectvalueconvenience.Test.__init__ (
    self,
    a,
    kwa )
```

1.354.2 Member Function Documentation

1.354.2.1 `runanise()`

```
def anise.test.anisetest.AniseTestCase.runanise (
    self,
    taskname,
    values = None,
    loadfeaturesfrom = None ) [inherited]
```

1.354.2.2 `setUp()`

```
def anise.test.anisetest.AniseTestCase.setUp (
    self ) [inherited]
```

1.354.2.3 `tearDown()`

```
def anise.test.anisetest.AniseTestCase.tearDown (
    self ) [inherited]
```

1.354.2.4 `test_a()`

```
def anise.test.coretests.t08projectvalueconvenience.Test.test_a (
    self )
```

1.354.2.5 `test_b()`

```
def anise.test.coretests.t08projectvalueconvenience.Test.test_b (
    self )
```

1.354.3 Member Data Documentation

1.354.3.1 `mydir`

```
anise.test.anisetest.AniseTestCase.mydir [inherited]
```

1.354.3.2 packagestoredir

`anise.test.anisetest.AniseTestCase.packagestoredir` [inherited]

1.354.3.3 projdescfile

`anise.test.anisetest.AniseTestCase.projdescfile` [inherited]

1.354.3.4 projectdesc

`anise.test.anisetest.AniseTestCase.projectdesc` [inherited]

1.354.3.5 rootdir

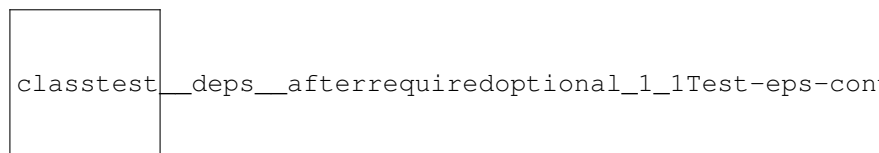
`anise.test.anisetest.AniseTestCase.rootdir` [inherited]

The documentation for this class was generated from the following file:

- `anise/test/coretests/t08projectvalueconvenience.py`

1.355 test_deps_afterrequiredoptional.Test Class Reference

Inheritance diagram for `test_deps_afterrequiredoptional.Test`:



Public Member Functions

- def `test_a` (self)

1.355.1 Member Function Documentation

1.355.1.1 test_a()

```
def test_deps_afterrequiredoptional.Test.test_a (
    self )
```

The documentation for this class was generated from the following file:

- [anise/data/PyDepsEngine/_meta/test_deps_afterrequiredoptional.py](#)

1.356 anise.features.projectdescriptioneditor.testing.TestingImplementation Class Reference

Inheritance diagram for anise.features.projectdescriptioneditor.testing.TestingImplementation:

```
classanise_1_1features_1_1projectdescriptioneditor_
```

Public Member Functions

- `def __init__ (self, label)`
- `def execute (self, params)`

Public Attributes

- [label](#)

1.356.1 Constructor & Destructor Documentation

1.356.1.1 __init__()

```
def anise.features.projectdescriptioneditor.testing.TestingImplementation.__init__ (
    self,
    label )
```

1.356.2 Member Function Documentation

1.356.2.1 execute()

```
def anise.features.projectdescriptioneditor.testing.TestingImplementation.execute (
    self,
    params )
```


1.356.3 Member Data Documentation

1.356.3.1 label

`anise.features.projectdescriptioneditor.testing.TestingImplementation.label`

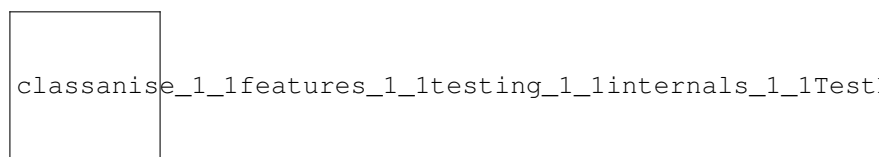
The documentation for this class was generated from the following file:

- `anise/features/projectdescriptioneditor/testing.py`

1.357 anise.features.testing.internals.TestReportWebModule Class Reference

Anise web module for generating the html test report.

Inheritance diagram for `anise.features.testing.internals.TestReportWebModule`:



Public Member Functions

- `def __init__` (self, `testrun`, title)
- `def oninitialize` (self)

Public Attributes

- `testrun`

Static Public Attributes

- `scripts`
- `style`
- `htmlmainpart`
- `headeradditions`

Static Private Member Functions

- `def _successpresentation` (`_success`, failcounter=None)
- `def _gauge` (failed, total)

Static Private Attributes

- [_htmltemplate_testpackage](#)
- [_htmltemplate_test](#)

1.357.1 Detailed Description

Anise web module for generating the html test report.

1.357.2 Constructor & Destructor Documentation

1.357.2.1 __init__()

```
def anise.features.testing.internals.TestReportWebModule.__init__ (
    self,
    testrun,
    title )
```

1.357.3 Member Function Documentation

1.357.3.1 _gauge()

```
def anise.features.testing.internals.TestReportWebModule._gauge (
    failed,
    total ) [static], [private]
```

1.357.3.2 _successpresentation()

```
def anise.features.testing.internals.TestReportWebModule._successpresentation (
    _success,
    failcounter = None ) [static], [private]
```

1.357.3.3 oninitialize()

```
def anise.features.testing.internals.TestReportWebModule.oninitialize (
    self )
```

1.357.4 Member Data Documentation

1.357.4.1 `_htmltemplate_test`

`anise.features.testing.internals.TestReportWebModule._htmltemplate_test` [static], [private]

1.357.4.2 `_htmltemplate_testpackage`

`anise.features.testing.internals.TestReportWebModule._htmltemplate_testpackage` [static], [private]

1.357.4.3 `headeradditions`

`anise.features.testing.internals.TestReportWebModule.headeradditions` [static]

1.357.4.4 `htmlmainpart`

`anise.features.testing.internals.TestReportWebModule.htmlmainpart` [static]

1.357.4.5 `scripts`

`anise.features.testing.internals.TestReportWebModule.scripts` [static]

1.357.4.6 `style`

`anise.features.testing.internals.TestReportWebModule.style` [static]

1.357.4.7 `testrun`

`anise.features.testing.internals.TestReportWebModule.testrun`

The documentation for this class was generated from the following file:

- [anise/features/testing.py](#)

1.358 anise.features.testing.internals.TestRun Class Reference

This instance keeps track of the test results.

Public Member Functions

- def `__init__` (self)
- def `addresult` (self, packagename, testname, output, success, duration_msec)
Adds a test result.
- def `testscout` (self, with_success=None)
Returns the number of tests done.
- def `settest` (self, packagename, testname)
Running setpackage, begintest and endtest together is a convenient alternative to addresult doing mostly the same.
- def `begintest` (self)
Running setpackage, begintest and endtest together is a convenient alternative to addresult doing mostly the same.
- def `endtest` (self, output, success)
Running setpackage, begintest and endtest together is a convenient alternative to addresult doing mostly the same.

Public Attributes

- `testpackages`
- `testpackagelist`
- `currentpackagename`
- `currenttestname`
- `currentbegan`

Static Public Attributes

- `SUCCESS_FAILED`
- `SUCCESS_SUCCESSFUL_WITH_WARNINGS`
- `SUCCESS_SUCCESSFUL`

1.358.1 Detailed Description

This instance keeps track of the test results.

It is used by test runners for return test results.

1.358.2 Constructor & Destructor Documentation

1.358.2.1 `__init__()`

```
def anise.features.testing.internals.TestRun.__init__ (
    self )
```

1.358.3 Member Function Documentation

1.358.3.1 addresult()

```
def anise.features.testing.internals.TestRun.addresult (
    self,
    packagename,
    testname,
    output,
    success,
    duration_msec )
```

Adds a test result.

Parameters

<i>packagename</i>	The test package name.
<i>testname</i>	The test name.
<i>output</i>	The test run output.
<i>success</i>	The test run success state.
<i>duration_msec</i>	The execution duration in milliseconds.

1.358.3.2 begintest()

```
def anise.features.testing.internals.TestRun.begintest (
    self )
```

Running setpackage, begintest and endtest together is a convenient alternative to addresult doing mostly the same.

1.358.3.3 endtest()

```
def anise.features.testing.internals.TestRun.endtest (
    self,
    output,
    success )
```

Running setpackage, begintest and endtest together is a convenient alternative to addresult doing mostly the same.

Parameters

<i>output</i>	The test output.
<i>success</i>	The test run success state.

1.358.3.4 settest()

```
def anise.features.testing.internals.TestRun.settest (
    self,
    packagename,
    testname )
```

Running setpackage, begintest and endtest together is a convenient alternative to addressresult doing mostly the same.

Parameters

<i>packagename</i>	The test package name.
<i>testname</i>	The test name.

1.358.3.5 testscount()

```
def anise.features.testing.internals.TestRun.testscount (
    self,
    with_success = None )
```

Returns the number of tests done.

Parameters

<i>with_success</i>	Only count tests with this success state.
---------------------	---

1.358.4 Member Data Documentation

1.358.4.1 currentbegan

```
anise.features.testing.internals.TestRun.currentbegan
```

1.358.4.2 currentpackagename

```
anise.features.testing.internals.TestRun.currentpackagename
```

1.358.4.3 currenttestname

`anise.features.testing.internals.TestRun.currenttestname`

1.358.4.4 SUCCESS_FAILED

`anise.features.testing.internals.TestRun.SUCCESS_FAILED` [static]

1.358.4.5 SUCCESS_SUCCESSFUL

`anise.features.testing.internals.TestRun.SUCCESS_SUCCESSFUL` [static]

1.358.4.6 SUCCESS_SUCCESSFUL_WITH_WARNINGS

`anise.features.testing.internals.TestRun.SUCCESS_SUCCESSFUL_WITH_WARNINGS` [static]

1.358.4.7 testpackagelist

`anise.features.testing.internals.TestRun.testpackagelist`

1.358.4.8 testpackages

`anise.features.testing.internals.TestRun.testpackages`

The documentation for this class was generated from the following file:

- [anise/features/testing.py](#)

1.359 anise.features.testing.internals.Tests Class Reference

Storage of all tests.

Public Member Functions

- def `__init__` (self)
- def `add` (self, packagename, name, runner, testonrelease=False, kwargs)
Adds a test.
- def `packagelist` (self)
Gets the list of all test package names.
- def `listbypackage` (self, package)
Gets the list of all test names in a package.
- def `test_on_release` (self, package, name)
Adds a test to the releasing, so it is automatically executed each time 'release' is executed.

Public Attributes

- `list`

1.359.1 Detailed Description

Storage of all tests.

1.359.2 Constructor & Destructor Documentation

1.359.2.1 `__init__()`

```
def anise.features.testing.internals.Tests.__init__ (
    self )
```

1.359.3 Member Function Documentation

1.359.3.1 `add()`

```
def anise.features.testing.internals.Tests.add (
    self,
    packagename,
    name,
    runner,
    testonrelease = False,
    kwargs )
```

Adds a test.

Parameters

<i>packagename</i>	The test package name.
<i>name</i>	The test name.
<i>runner</i>	The function to execute for running the test.
<i>testonrelease</i>	If this test should automatically be made a part of the releasing.
<i>kwargs</i>	Additional arguments for the test runner function.

1.359.3.2 listbypackage()

```
def anise.features.testing.internals.Tests.listbypackage (
    self,
    package )
```

Gets the list of all test names in a package.

Parameters

<i>package</i>	The test package name.
----------------	------------------------

1.359.3.3 packagelist()

```
def anise.features.testing.internals.Tests.packagelist (
    self )
```

Gets the list of all test package names.

1.359.3.4 test_on_release()

```
def anise.features.testing.internals.Tests.test_on_release (
    self,
    package,
    name )
```

Adds a test to the releasing, so it is automatically executed each time 'release' is executed.

Parameters

<i>package</i>	The test package name.
<i>name</i>	The test name.

1.359.4 Member Data Documentation

1.359.4.1 list

```
anise.features.testing.internals.Tests.list
```

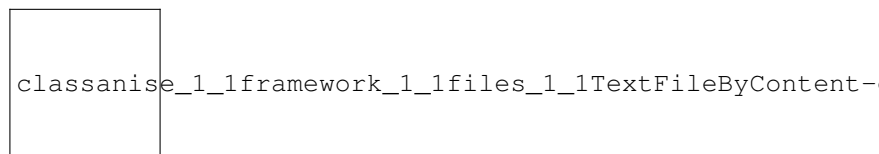
The documentation for this class was generated from the following file:

- [anise/features/testing.py](#)

1.360 anise.framework.files.TextFileByContent Class Reference

Holds a function and some parameters for calling later on (like a delegate; or a more explicit, specialized and 'xml-compatible' version of a lambda) in order to get a [Filestructure](#) as function result.

Inheritance diagram for `anise.framework.files.TextFileByContent`:



Public Member Functions

- `def __init__ (self, content)`
- `def datakeys (self)`
Returns a list of keys for all stored metadata properties.
- `def setdata (self, k, v)`
Sets a metadata property.
- `def getdata (self, k, deflt)`
Gets a metadata property.
- `def initialize (self)`
Initializes this [Filestructure](#).
- `def path (self)`
The path to this structure in the filesystem.
- `def dl (self, subpath="", to=None, progresscallback=None)`
Copies the complete filestructure or a subdirectory to a new destination.
- `def with_modified_rootname (self, newname)`
Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.
- `def mv (self, newname)`
Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.
- `def setautoopen (self, path, interminal=False)`
Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

1.360.1 Detailed Description

Holds a function and some parameters for calling later on (like a delegate; or a more explicit, specialized and 'xml-compatible' version of a lambda) in order to get a [Filestructure](#) as function result.

It acts as a proxy on top of this [Filestructure](#).

1.360.2 Constructor & Destructor Documentation

1.360.2.1 __init__()

```
def anise.framework.files.TextFileByContent.__init__ (
    self,
    content )
```

Parameters

<i>content</i>	The text content for the file.
----------------	--------------------------------

1.360.3 Member Function Documentation

1.360.3.1 datakeys()

```
def anise.framework.files.Filestructure.datakeys (
    self ) [inherited]
```

Returns a list of keys for all stored metadata properties.

1.360.3.2 dl()

```
def anise.framework.files.Filestructure.dl (
    self,
    subpath = "",
    to = None,
    progresscallback = None ) [inherited]
```

Copies the complete filestructure or a subdirectory to a new destination.

Parameters

<i>subpath</i>	If given, only this subdirectory will be copied.
<i>to</i> Generated by Doxygen	If given, it specifies the destination path; if not given, a new directory in the current working directory will be used.
<i>progresscallback</i>	Callback function for retrieving progress information.

1.360.3.3 getdata()

```
def anise.framework.files.Filestructure.getdata (
    self,
    k,
    deflt ) [inherited]
```

Gets a metadata property.

Parameters

<i>k</i>	The key name.
<i>deflt</i>	The default value (returned if the key doesn't exist).

1.360.3.4 initialize()

```
def anise.framework.files.Filestructure.initialize (
    self ) [inherited]
```

Initializes this [Filestructure](#).

When this is called, the dynamic content (if any) is created and written to disc. It is automatically called for typical usage and is not required to be called manually.

1.360.3.5 mv()

```
def anise.framework.files.Filestructure.mv (
    self,
    newname ) [inherited]
```

Returns a new [anise.framework.files.Filestructure](#) resulting in a movement from the old one.

Parameters

<i>newname</i>	The new name (absolute or relative).
----------------	--------------------------------------

1.360.3.6 path()

```
def anise.framework.files.Filestructure.path (
    self ) [inherited]
```

The path to this structure in the filesystem.

1.360.3.7 setautoopen()

```
def anise.framework.files.Filestructure.setautoopen (
    self,
    path,
    interminal = False ) [inherited]
```

Places a hint which requests the anise engine to automatically open a file within this structure when it is returned to the user.

Parameters

<i>path</i>	The relative path to the file which should automatically be opened.
<i>interminal</i>	If this hint is for terminal (non-graphical) usage.

1.360.3.8 setdata()

```
def anise.framework.files.Filestructure.setdata (
    self,
    k,
    v ) [inherited]
```

Sets a metadata property.

Parameters

<i>k</i>	The key name.
<i>v</i>	The value.

1.360.3.9 with_modified_rootname()

```
def anise.framework.files.Filestructure.with_modified_rootname (
    self,
    newname ) [inherited]
```

Returns a new (temporary) [anise.framework.files.Filestructure](#) clone with a new directory name.

Parameters

<i>newname</i>	The name which the new root directory should get.
----------------	---

The documentation for this class was generated from the following file:

- [anise/framework/files.py](#)

1.361 anise.features.dependencies.Type Class Reference

Enumeration of different types of dependencies.

Static Public Attributes

- string `Included` = "included"
- string `Required` = "required"
- string `Required_HasAlternatives` = "required (has alternatives)"
- string `Recommended_HasAlternatives` = "recommended"
- string `Optional` = "optional"

1.361.1 Detailed Description

Enumeration of different types of dependencies.

It describes how strongly required a dependency is.

1.361.2 Member Data Documentation

1.361.2.1 Included

```
string anise.features.dependencies.Type.Included = "included" [static]
```

1.361.2.2 Optional

```
string anise.features.dependencies.Type.Optional = "optional" [static]
```

1.361.2.3 Recommended_HasAlternatives

```
string anise.features.dependencies.Type.Recommended_HasAlternatives = "recommended" [static]
```

1.361.2.4 Required

```
string anise.features.dependencies.Type.Required = "required" [static]
```

1.361.2.5 Required_HasAlternatives

```
string anise.features.dependencies.Type.Required_HasAlternatives = "required (has alternatives)"
[static]
```

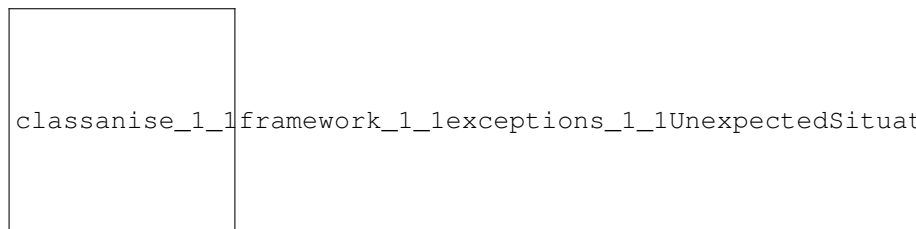
The documentation for this class was generated from the following file:

- [anise/features/dependencies.py](#)

1.362 anise.framework.exceptions.UnexpectedSituationError Class Reference

An unexpected situation led to an error.

Inheritance diagram for anise.framework.exceptions.UnexpectedSituationError:



Public Member Functions

- `def __call__ (self, args)`

1.362.1 Detailed Description

An unexpected situation led to an error.

1.362.2 Member Function Documentation

1.362.2.1 `__call__()`

```
def anise.framework.exceptions.AniseError.__call__ (
    self,
    args ) [inherited]
```

The documentation for this class was generated from the following file:

- [anise/framework/exceptions.py](#)

1.363 anise.framework.projects.Universe Class Reference

Implementation for universe objects.

Classes

- class [RegisteredTask](#)
A task which is registered to the universe object.

Public Member Functions

- def [__init__](#) (self, d, [executionscope](#)=None)
- def [addtask](#) (self, function, name="", label=None, important=True, invisible=False, sortindex=0)
Registers a task and makes it available for execution.
- def [registeredtasks](#) (self)
Returns a list of all registered tasks (as [Universe.RegisteredTask](#)).
- def [append](#) (self, d)
Stores a dictionary of values as properties of this universe object.
- def [addhook](#) (self, hookobj, fct, params=None, provides=None, requires=None, prepares=None)
Adds a handler for a hook to the universe object.
- def [removehook](#) (self, hookhandler)
Removes a hook, which was added before.
- def [gethooks](#) (self, hookobj)
Gets the handlers for a hook in the correct order.
- def [getvalue](#) (self, key, defaultvalue=None)
Returns the value of a certain property for the current universe object.
- def [setvalue](#) (self, key, value)
Sets the value of a certain property for the current universe object.

Public Attributes

- [executionscope](#)
- [hooks](#)

Private Attributes

- [_registeredtasks](#)

1.363.1 Detailed Description

Implementation for universe objects.

1.363.2 Constructor & Destructor Documentation

1.363.2.1 `__init__()`

```
def anise.framework.projects.Universe.__init__ (
    self,
    d,
    executionscope = None )
```

1.363.3 Member Function Documentation

1.363.3.1 `addhook()`

```
def anise.framework.projects.Universe.addhook (
    self,
    hookobj,
    fct,
    params = None,
    provides = None,
    requires = None,
    prepares = None )
```

Adds a handler for a hook to the universe object.

Returns

: An [anise.framework.features.HookHandler](#) instance.

1.363.3.2 `addtask()`

```
def anise.framework.projects.Universe.addtask (
    self,
    function,
    name = "",
    label = None,
    important = True,
    invisible = False,
    sortindex = 0 )
```

Registers a task and makes it available for execution.

Parameters

<i>function</i>	The function which implements the task.
<i>name</i>	The task name.
<i>label</i>	The task label text.
<i>important</i>	If this is an important/interesting task (presented differently in choosers).
<i>invisible</i>	If this task should be hidden in choosers.
<i>sortindex</i>	A number which controls the display order of tasks in choosers.

1.363.3.3 append()

```
def anise.framework.projects.Universe.append (
    self,
    d )
```

Stores a dictionary of values as properties of this universe object.

1.363.3.4 gethooks()

```
def anise.framework.projects.Universe.gethooks (
    self,
    hookobj )
```

Gets the handlers for a hook in the correct order.

Parameters

<i>hookobj</i>	The hook you want to get handlers for.
----------------	--

1.363.3.5 getvalue()

```
def anise.framework.projects.Universe.getvalue (
    self,
    key,
    defaultvalue = None )
```

Returns the value of a certain property for the current universe object.

If this property does not exist, it returns the defaultval. `getvalue("foo.bar")` is similar to `universe.foo.bar`.

Parameters

<i>key</i>	The key name of the property to get.
<i>defaultvalue</i>	This value is returned if the specified property does not exist.

1.363.3.6 registeredtasks()

```
def anise.framework.projects.Universe.registeredtasks (
    self )
```

Returns a list of all registered tasks (as [Universe.RegisteredTask](#)).

1.363.3.7 removehook()

```
def anise.framework.projects.Universe.removehook (
    self,
    hookhandler )
```

Removes a hook, which was added before.

Parameters

<i>hookhandler</i>	The HookHandler you got in addhook.
--------------------	-------------------------------------

1.363.3.8 setvalue()

```
def anise.framework.projects.Universe.setvalue (
    self,
    key,
    value )
```

Sets the value of a certain property for the current universe object.

If a parent property does not exist, it does nothing. Returns True if succeeded. `setvalue("foo.bar", "baz")` is similar to `universe.foo.bar = "baz"`.

Parameters

<i>key</i>	The key name of the property to set.
<i>value</i>	The new value for the specified property.

1.363.4 Member Data Documentation

1.363.4.1 _registeredtasks

```
anise.framework.projects.Universe._registeredtasks [private]
```

1.363.4.2 executionscope

```
anise.framework.projects.Universe.executionscope
```

1.363.4.3 hooks

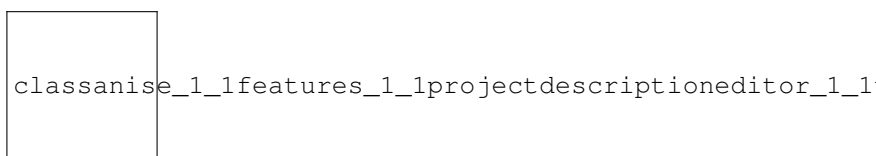
```
anise.framework.projects.Universe.hooks
```

The documentation for this class was generated from the following file:

- [anise/framework/projects.py](#)

1.364 anise.features.projectdescriptioneditor.testing.UnsetReleaseCriticalCustomAction Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.testing.UnsetReleaseCriticalCustomAction`:



Public Member Functions

- `def __init__ (self)`
- `def visible (self, ais, node)`
- `def execute (self, ais, node)`

Public Attributes

- [label](#)
- [function](#)
- [icon](#)

1.364.1 Constructor & Destructor Documentation

1.364.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.testing.UnsetReleaseCriticalCustomAction.__init__←
_ (
    self )
```

1.364.2 Member Function Documentation

1.364.2.1 execute()

```
def anise.features.projectdescriptioneditor.testing.UnsetReleaseCriticalCustomAction.execute (
    self,
    ais,
    node )
```

1.364.2.2 visible()

```
def anise.features.projectdescriptioneditor.testing.UnsetReleaseCriticalCustomAction.visible (
    self,
    ais,
    node )
```

1.364.3 Member Data Documentation

1.364.3.1 function

anise.features.projectdescriptioneditor.CustomAction.function [inherited]

1.364.3.2 icon

anise.features.projectdescriptioneditor.CustomAction.icon [inherited]

1.364.3.3 label

anise.features.projectdescriptioneditor.CustomAction.label [inherited]

The documentation for this class was generated from the following file:

- [anise/features/projectdescriptioneditor/testing.py](#)

1.365 anise.features.ui.internals.UserFeedback Class Reference

This interface provides mechanisms for asking the user for some kinds of interaction.

Inheritance diagram for anise.features.ui.internals.UserFeedback:

```
class anise_1_1features_1_1ui_1_1internals_1_1UserFeedback
```

Public Member Functions

- def `__init__` (self)
- def `message` (self, msg, btns, image)
Shows a message and asks the user to trigger one of the buttons.
- def `input` (self, msg, default, image)
Asks the user to enter text (single line).
- def `multilineinput` (self, msg, default, image)
Asks the user to enter text (multi line).
- def `choice` (self, msg, choices, canceltext, image, choiceimages)
Asks the user to choose from a list of alternatives.
- def `treechoice` (self, msg, choicetree, canceltext, actions, image)
Asks the user to choose from a tree of alternatives.
- def `messageyesno` (self, msg, image=None)
Shows a message and asks the user to trigger either Yes or No.
- def `messageok` (self, msg, image=None)
Shows a message and asks the user to confirm it.

1.365.1 Detailed Description

This interface provides mechanisms for asking the user for some kinds of interaction.

1.365.2 Constructor & Destructor Documentation

1.365.2.1 `__init__()`

```
def anise.features.ui.internals.UserFeedback.__init__ (
    self )
```

1.365.3 Member Function Documentation

1.365.3.1 choice()

```
def anise.features.ui.internals.UserFeedback.choice (
    self,
    msg,
    choices,
    canceltext,
    image,
    choiceimages )
```

Asks the user to choose from a list of alternatives.

Parameters

<i>msg</i>	The message to show.
<i>choices</i>	List of choices.
<i>canceltext</i>	Label of the Cancel button.
<i>image</i>	Icon name.
<i>choiceimages</i>	Icon names for the choices.

Returns

: The index of the choice the user made, or None if user cancelled.

1.365.3.2 input()

```
def anise.features.ui.internals.UserFeedback.input (
    self,
    msg,
    default,
    image )
```

Asks the user to enter text (single line).

Parameters

<i>msg</i>	The message to show.
<i>default</i>	The default text.
<i>image</i>	Icon name.

Returns

: The entered text, or None if user cancelled.

1.365.3.3 message()

```
def anise.features.ui.internals.UserFeedback.message (
    self,
    msg,
    btns,
    image )
```

Shows a message and asks the user to trigger one of the buttons.

Parameters

<i>msg</i>	Message text.
<i>btns</i>	Buttons to show (list of strings).
<i>image</i>	Icon name.

Returns

: The index of the selected button.

1.365.3.4 messageok()

```
def anise.features.ui.internals.UserFeedback.messageok (
    self,
    msg,
    image = None )
```

Shows a message a asks the user to confirm it.

Parameters

<i>msg</i>	The message to show.
<i>image</i>	Icon name.

1.365.3.5 messageyesno()

```
def anise.features.ui.internals.UserFeedback.messageyesno (
    self,
    msg,
    image = None )
```

Shows a message and asks the user to trigger either Yes or No.

Parameters

<i>msg</i>	The message to show.
<i>image</i>	Icon name.

Returns

: True, if the user selected Yes.

1.365.3.6 multilineinput()

```
def anise.features.ui.internals.UserFeedback.multilineinput (
    self,
    msg,
    default,
    image )
```

Asks the user to enter text (multi line).

Parameters

<i>msg</i>	The message to show.
<i>default</i>	The default text.
<i>image</i>	Icon name.

Returns

: The entered text, or None if user cancelled.

1.365.3.7 treechoice()

```
def anise.features.ui.internals.UserFeedback.treechoice (
    self,
    msg,
    choicetree,
    canceltext,
    actions,
    image )
```

Asks the user to choose from a tree of alternatives.

Parameters

<i>msg</i>	The message to show.
<i>choicetree</i>	The tree of available choices. Instance of ChoiceTree .
<i>canceltext</i>	Label of the Cancel button.
<i>actions</i>	Additional actions.
<i>image</i>	Icon name.

Returns

: The chosen tree node ([ChoiceTree.Node](#)), or None if user cancelled.

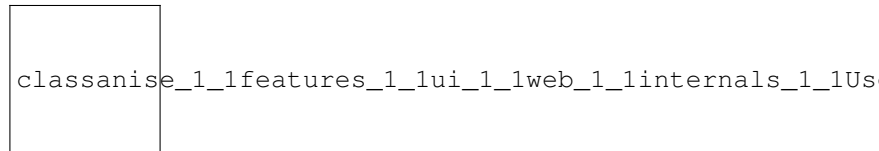
The documentation for this class was generated from the following file:

- [anise/features/ui.py](#)

1.366 anise.features.ui.web.internals.UserFeedback Class Reference

Uses available web browser sessions for user interaction.

Inheritance diagram for `anise.features.ui.web.internals.UserFeedback`:

**Classes**

- class [FeedbackRequest](#)

Public Member Functions

- def [__init__](#) (self)
- def [message](#) (self, msg, btns, image)
- def [input](#) (self, msg, default, image)
- def [multilineinput](#) (self, msg, default, image)
- def [treechoice](#) (self, msg, choicetree, canceltext, actions, image)
- def [choice](#) (self, msg, choices, canceltext, image, choiceimages)
- def [isavailable](#) (self)
- def [messageyesno](#) (self, msg, image=None)
Shows a message and asks the user to trigger either Yes or No.
- def [messageok](#) (self, msg, image=None)
Shows a message a asks the user to confirm it.

Public Attributes

- [nextrequestindex](#)
- [currentrequestindex](#)

Private Member Functions

- def [_webrequest](#) (self, request, requestdata)

1.366.1 Detailed Description

Uses available web browser sessions for user interaction.

1.366.2 Constructor & Destructor Documentation

1.366.2.1 `__init__()`

```
def anise.features.ui.web.internals.UserFeedback.__init__ (
    self )
```

1.366.3 Member Function Documentation

1.366.3.1 `_webrequest()`

```
def anise.features.ui.web.internals.UserFeedback._webrequest (
    self,
    request,
    requestdata ) [private]
```

1.366.3.2 `choice()`

```
def anise.features.ui.web.internals.UserFeedback.choice (
    self,
    msg,
    choices,
    canceltext,
    image,
    choiceimages )
```

1.366.3.3 `input()`

```
def anise.features.ui.web.internals.UserFeedback.input (
    self,
    msg,
    default,
    image )
```

1.366.3.4 isavailable()

```
def anise.features.ui.web.internals.UserFeedback.isavailable (
    self )
```

1.366.3.5 message()

```
def anise.features.ui.web.internals.UserFeedback.message (
    self,
    msg,
    btns,
    image )
```

1.366.3.6 messageok()

```
def anise.features.ui.internals.UserFeedback.messageok (
    self,
    msg,
    image = None ) [inherited]
```

Shows a message a asks the user to confirm it.

Parameters

<i>msg</i>	The message to show.
<i>image</i>	Icon name.

1.366.3.7 messageyesno()

```
def anise.features.ui.internals.UserFeedback.messageyesno (
    self,
    msg,
    image = None ) [inherited]
```

Shows a message and asks the user to trigger either Yes or No.

Parameters

<i>msg</i>	The message to show.
<i>image</i>	Icon name.

Returns

: True, if the user selected Yes.

1.366.3.8 multilineinput()

```
def anise.features.ui.web.internals.UserFeedback.multilineinput (
    self,
    msg,
    default,
    image )
```

1.366.3.9 treechoice()

```
def anise.features.ui.web.internals.UserFeedback.treechoice (
    self,
    msg,
    choicetree,
    canceltext,
    actions,
    image )
```

1.366.4 Member Data Documentation**1.366.4.1 currentrequestindex**

```
anise.features.ui.web.internals.UserFeedback.currentrequestindex
```

1.366.4.2 nextrequestindex

```
anise.features.ui.web.internals.UserFeedback.nextrequestindex
```

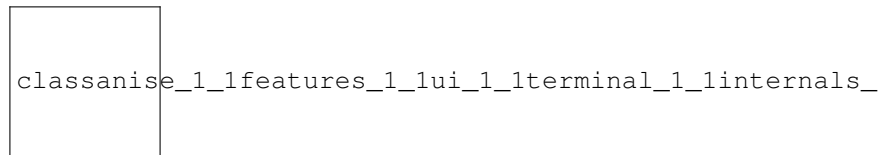
The documentation for this class was generated from the following file:

- [anise/features/ui/web.py](#)

1.367 anise.features.ui.terminal.internals.UserFeedback Class Reference

Uses the standard input and standard output channel of the terminal for user interaction.

Inheritance diagram for anise.features.ui.terminal.internals.UserFeedback:



Public Member Functions

- def `__init__` (self)
- def `message` (self, msg, btns, image)
- def `input` (self, msg, default, image)
- def `multilineinput` (self, msg, default, image)
- def `treechoice` (self, msg, choicetree, canceltext, actions, image)
- def `choice` (self, msg, choices, canceltext, image, choiceimages=None)
- def `messageyesno` (self, msg, image=None)
 - Shows a message and asks the user to trigger either Yes or No.*
- def `messageok` (self, msg, image=None)
 - Shows a message a asks the user to confirm it.*

Public Attributes

- `termhelper`

1.367.1 Detailed Description

Uses the standard input and standard output channel of the terminal for user interaction.

1.367.2 Constructor & Destructor Documentation

1.367.2.1 `__init__()`

```
def anise.features.ui.terminal.internals.UserFeedback.__init__ (
    self )
```

1.367.3 Member Function Documentation

1.367.3.1 choice()

```
def anise.features.ui.terminal.internals.UserFeedback.choice (
    self,
    msg,
    choices,
    canceltext,
    image,
    choiceimages = None )
```

1.367.3.2 input()

```
def anise.features.ui.terminal.internals.UserFeedback.input (
    self,
    msg,
    default,
    image )
```

1.367.3.3 message()

```
def anise.features.ui.terminal.internals.UserFeedback.message (
    self,
    msg,
    btns,
    image )
```

1.367.3.4 messageok()

```
def anise.features.ui.internals.UserFeedback.messageok (
    self,
    msg,
    image = None ) [inherited]
```

Shows a message a asks the user to confirm it.

Parameters

<i>msg</i>	The message to show.
<i>image</i>	Icon name.

1.367.3.5 messageyesno()

```
def anise.features.ui.internals.UserFeedback.messageyesno (
    self,
    msg,
    image = None ) [inherited]
```

Shows a message and asks the user to trigger either Yes or No.

Parameters

<i>msg</i>	The message to show.
<i>image</i>	Icon name.

Returns

: True, if the user selected Yes.

1.367.3.6 multilineinput()

```
def anise.features.ui.terminal.internals.UserFeedback.multilineinput (
    self,
    msg,
    default,
    image )
```

1.367.3.7 treechoice()

```
def anise.features.ui.terminal.internals.UserFeedback.treechoice (
    self,
    msg,
    choicetree,
    canceltext,
    actions,
    image )
```

1.367.4 Member Data Documentation

1.367.4.1 termhelper

`anise.features.ui.terminal.internals.UserFeedback.termhelper`

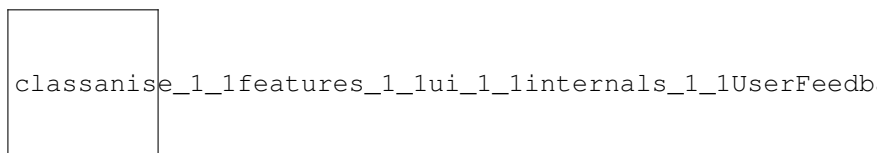
The documentation for this class was generated from the following file:

- [anise/features/ui/terminal.py](#)

1.368 anise.features.ui.internals.UserFeedbackProxy Class Reference

This user feedback implementation forwards requests to another internal [UserFeedback](#), depending on which mode (web, terminal) is selected.

Inheritance diagram for anise.features.ui.internals.UserFeedbackProxy:



Public Member Functions

- def [__init__](#) (self)
- def [message](#) (self, msg, btns, image=None)
- def [input](#) (self, msg, default="", image=None)
- def [multilineinput](#) (self, msg, default="", image=None)
- def [choice](#) (self, msg, choices, canceltext="Cancel", image=None, choiceimages=None)
- def [treechoice](#) (self, msg, choicetree, canceltext="Cancel", actions=None, image=None)
- def [messageyesno](#) (self, msg, image=None)
 - Shows a message and asks the user to trigger either Yes or No.*
- def [messageok](#) (self, msg, image=None)
 - Shows a message a asks the user to confirm it.*

Private Member Functions

- def [_getimpl](#) (self)

1.368.1 Detailed Description

This user feedback implementation forwards requests to another internal [UserFeedback](#), depending on which mode (web, terminal) is selected.

1.368.2 Constructor & Destructor Documentation

1.368.2.1 [__init__](#)()

```
def anise.features.ui.internals.UserFeedbackProxy.__init__ (
    self )
```

1.368.3 Member Function Documentation

1.368.3.1 `_getimpl()`

```
def anise.features.ui.internals.UserFeedbackProxy._getimpl (
    self ) [private]
```

1.368.3.2 `choice()`

```
def anise.features.ui.internals.UserFeedbackProxy.choice (
    self,
    msg,
    choices,
    canceltext = "Cancel",
    image = None,
    choiceimages = None )
```

1.368.3.3 `input()`

```
def anise.features.ui.internals.UserFeedbackProxy.input (
    self,
    msg,
    default = "",
    image = None )
```

1.368.3.4 `message()`

```
def anise.features.ui.internals.UserFeedbackProxy.message (
    self,
    msg,
    btns,
    image = None )
```

1.368.3.5 `messageok()`

```
def anise.features.ui.internals.UserFeedback.messageok (
    self,
    msg,
    image = None ) [inherited]
```

Shows a message a asks the user to confirm it.

Parameters

<i>msg</i>	The message to show.
<i>image</i>	Icon name.

1.368.3.6 messageyesno()

```
def anise.features.ui.internals.UserFeedback.messageyesno (
    self,
    msg,
    image = None ) [inherited]
```

Shows a message and asks the user to trigger either Yes or No.

Parameters

<i>msg</i>	The message to show.
<i>image</i>	Icon name.

Returns

: True, if the user selected Yes.

1.368.3.7 multilineinput()

```
def anise.features.ui.internals.UserFeedbackProxy.multilineinput (
    self,
    msg,
    default = "",
    image = None )
```

1.368.3.8 treechoice()

```
def anise.features.ui.internals.UserFeedbackProxy.treechoice (
    self,
    msg,
    choicetree,
    canceltext = "Cancel",
    actions = None,
    image = None )
```

The documentation for this class was generated from the following file:

- [anise/features/ui.py](#)

1.369 anise.Utls Class Reference

Static Public Member Functions

- def [clearcaches](#) ()
- def [isanisesynced](#) (op, eurl)
- def [getaniseprojectname](#) (op, projectdesceurl)
- def [getaniseprojecttasks](#) (rooteurl)

Static Public Attributes

- [lock](#) = threading.Lock()
- dictionary [isanisesyncedcache](#) = {}
- dictionary [aniseprojectdetailscache](#) = {}
- dictionary [aniseprojecttaskscache](#) = {}

Static Private Member Functions

- def [_findaniseroot](#) (op, eurl)

1.369.1 Member Function Documentation

1.369.1.1 _findaniseroot()

```
def anise.Utls._findaniseroot (
    op,
    eurl,
    object ) [static], [private]
```

1.369.1.2 clearcaches()

```
def anise.Utls.clearcaches ( ) [static]
```

1.369.1.3 getaniseprojectname()

```
def anise.Utls.getaniseprojectname (
    op,
    projectdesceurl,
    object ) [static]
```

1.369.1.4 getaniseprojecttasks()

```
def anise.Utls.getaniseprojecttasks (
    rooteurl,
    object ) [static]
```

1.369.1.5 isanisesynced()

```
def anise.Utls.isanisesynced (
    op,
    eurl,
    object ) [static]
```

1.369.2 Member Data Documentation

1.369.2.1 aniseprojectdetailscache

```
dictionary anise.Utls.aniseprojectdetailscache = {} [static]
```

1.369.2.2 aniseprojecttaskscache

```
dictionary anise.Utls.aniseprojecttaskscache = {} [static]
```

1.369.2.3 isanisesyncedcache

```
dictionary anise.Utls.isanisesyncedcache = {} [static]
```

1.369.2.4 lock

```
anise.Utls.lock = threading.Lock() [static]
```

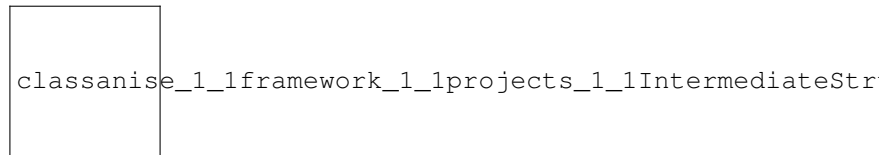
The documentation for this class was generated from the following file:

- `_meta/shallot_plugin/anise.py`

1.370 anise.framework.projects.IntermediateStructure.Value Class Reference

A [Value](#) entry in the project description structure.

Inheritance diagram for anise.framework.projects.IntermediateStructure.Value:



Public Member Functions

- def [__init__](#) (self, t, k=None, v="", [oref](#)="None", args)
- def [appendarg](#) (self, node)
Appends a new [IntermediateStructure.Value](#) to a list node.
- def [value](#) (self)
Returns the value (for primitive values).
- def [setvalue](#) (self, v)
Sets the value (for primitive values).
- def [vtype](#) (self)
Returns the value type name.
- def [setvtype](#) (self, t)
Sets the value type name.
- def [oref](#) (self)
Returns the objectref (for oref values).
- def [setoref](#) (self, v)
Sets the objectref (for oref values).
- def [getprimitiverepr](#) (self)
Returns a string representation of the value, if it is a primitive one (None otherwise).
- def [__str__](#) (self)
- def [key](#) (self)
The key name.
- def [parent](#) (self)
Returns the parent node (either [IntermediateStructure](#) or [IntermediateStructure.Node](#)).
- def [args](#) (self)
Returns the list of stored argument keys.
- def [getarg](#) (self, [key](#))
Returns an argument (as [IntermediateStructure.Value](#)).
- def [setarg](#) (self, [key](#), node)
Sets an argument.
- def [removearg](#) (self, [key](#))
Removes an argument.
- def [clearargs](#) (self)
Clears the list of arguments.
- def [remove](#) (self)
Removes this node.

Static Public Member Functions

- def [istru](#)e (s)
Returns True iff the given string can represent a boolean 'True' value.
- def [fromstring](#) (val, [key](#)=None)
Convenience method which creates a string [Value](#).
- def [frombool](#) (val, [key](#)=None)
Convenience method which creates a boolean [Value](#).

Public Attributes

- [isvalue](#)
- [islet](#)

Private Member Functions

- def [__realign_listvalues](#) (self)
Makes internal list realignment tasks.
- def [__setprimitive](#) (self, v)
Setting a primitive value with typechecking.

Private Attributes

- [_vtype](#)
- [_value](#)
- [_oref](#)

1.370.1 Detailed Description

A [Value](#) entry in the project description structure.

This corresponds to a `val` node in the xml format. It represents an arbitrary value (primitive or composed type).

1.370.2 Constructor & Destructor Documentation

1.370.2.1 `__init__()`

```
def anise.framework.projects.IntermediateStructure.Value.__init__ (
    self,
    t,
    k = None,
    v = "",
    oref = "None",
    args )
```

Parameters

<i>t</i>	The value type
<i>k</i>	The key (i.e. property name in parent object)
<i>v</i>	(optional) The value string for a primitive value.
<i>oref</i>	(optional) The oref string for a objectref value.

1.370.3 Member Function Documentation

1.370.3.1 `__str__()`

```
def anise.framework.projects.IntermediateStructure.Value.__str__ (
    self )
```

1.370.3.2 `_realign_listvalues()`

```
def anise.framework.projects.IntermediateStructure.Value._realign_listvalues (
    self ) [private]
```

Makes internal list realignment tasks.

This is required to call after the list structure changes.

1.370.3.3 `_setprimitive()`

```
def anise.framework.projects.IntermediateStructure.Value._setprimitive (
    self,
    v ) [private]
```

Setting a primitive value with typechecking.

1.370.3.4 `appendarg()`

```
def anise.framework.projects.IntermediateStructure.Value.appendarg (
    self,
    node )
```

Appends a new [IntermediateStructure.Value](#) to a list node.

1.370.3.5 args()

```
def anise.framework.projects.IntermediateStructure.Node.args (
    self ) [inherited]
```

Returns the list of stored argument keys.

1.370.3.6 clearargs()

```
def anise.framework.projects.IntermediateStructure.Node.clearargs (
    self ) [inherited]
```

Clears the list of arguments.

1.370.3.7 frombool()

```
def anise.framework.projects.IntermediateStructure.Value.frombool (
    val,
    key = None ) [static]
```

Convenience method which creates a boolean [Value](#).

1.370.3.8 fromstring()

```
def anise.framework.projects.IntermediateStructure.Value.fromstring (
    val,
    key = None ) [static]
```

Convenience method which creates a string [Value](#).

1.370.3.9 getarg()

```
def anise.framework.projects.IntermediateStructure.Node.getarg (
    self,
    key ) [inherited]
```

Returns an argument (as [IntermediateStructure.Value](#)).

Parameters

key	The argument key.
-----	-------------------

1.370.3.10 `getprimitiverepr()`

```
def anise.framework.projects.IntermediateStructure.Value.getprimitiverepr (
    self )
```

Returns a string representation of the value, if it is a primitive one (None otherwise).

1.370.3.11 `istrue()`

```
def anise.framework.projects.IntermediateStructure.Value.istrue (
    s ) [static]
```

Returns True iff the given string can represent a boolean 'True' value.

1.370.3.12 `key()`

```
def anise.framework.projects.IntermediateStructure.Node.key (
    self ) [inherited]
```

The key name.

1.370.3.13 `oref()`

```
def anise.framework.projects.IntermediateStructure.Value.oref (
    self )
```

Returns the objectref (for oref values).

1.370.3.14 `parent()`

```
def anise.framework.projects.IntermediateStructure.Node.parent (
    self ) [inherited]
```

Returns the parent node (either [IntermediateStructure](#) or [IntermediateStructure.Node](#)).

1.370.3.15 remove()

```
def anise.framework.projects.IntermediateStructure.Node.remove (
    self ) [inherited]
```

Removes this node.

1.370.3.16 removearg()

```
def anise.framework.projects.IntermediateStructure.Node.removearg (
    self,
    key ) [inherited]
```

Removes an argument.

Parameters

<i>key</i>	The argument key.
------------	-------------------

1.370.3.17 setarg()

```
def anise.framework.projects.IntermediateStructure.Node.setarg (
    self,
    key,
    node ) [inherited]
```

Sets an argument.

Parameters

<i>key</i>	The argument key.
<i>node</i>	The argument value (as IntermediateStructure.Value).

1.370.3.18 setoref()

```
def anise.framework.projects.IntermediateStructure.Value.setoref (
    self,
    v )
```

Sets the objectref (for oref values).

1.370.3.19 setvalue()

```
def anise.framework.projects.IntermediateStructure.Value.setvalue (
    self,
    v )
```

Sets the value (for primitive values).

1.370.3.20 setvtype()

```
def anise.framework.projects.IntermediateStructure.Value.setvtype (
    self,
    t )
```

Sets the value type name.

1.370.3.21 value()

```
def anise.framework.projects.IntermediateStructure.Value.value (
    self )
```

Returns the value (for primitive values).

1.370.3.22 vtype()

```
def anise.framework.projects.IntermediateStructure.Value.vtype (
    self )
```

Returns the value type name.

1.370.4 Member Data Documentation**1.370.4.1 _oref**

```
anise.framework.projects.IntermediateStructure.Value._oref [private]
```

1.370.4.2 _value

```
anise.framework.projects.IntermediateStructure.Value._value [private]
```

1.370.4.3 _vtype

```
anise.framework.projects.IntermediateStructure.Value._vtype [private]
```

1.370.4.4 islet

```
anise.framework.projects.IntermediateStructure.Node.islet [inherited]
```

1.370.4.5 isvalue

```
anise.framework.projects.IntermediateStructure.Value.isvalue
```

The documentation for this class was generated from the following file:

- [anise/framework/projects.py](#)

1.371 anise.features.projectdescriptioneditor.versioncontrol.Vcs Class Reference

Public Member Functions

- def `__init__` (self, implclass, featurename, [label](#))
- def `ask` (self, ais)

Public Attributes

- [impl](#)
- [implclassname](#)
- [label](#)

1.371.1 Constructor & Destructor Documentation

1.371.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.versioncontrol.Vcs.__init__ (
    self,
    implclass,
    featurename,
    label )
```

1.371.2 Member Function Documentation

1.371.2.1 `ask()`

```
def anise.features.projectdescriptioneditor.versioncontrol.Vcs.ask (
    self,
    ais )
```

1.371.3 Member Data Documentation

1.371.3.1 impl

`anise.features.projectdescriptioneditor.versioncontrol.Vcs.impl`

1.371.3.2 implclassname

`anise.features.projectdescriptioneditor.versioncontrol.Vcs.implclassname`

1.371.3.3 label

`anise.features.projectdescriptioneditor.versioncontrol.Vcs.label`

The documentation for this class was generated from the following file:

- `anise/features/projectdescriptioneditor/versioncontrol.py`

1.372 anise.features.versioncontrol.internals.VersionControlSystem Class Reference

A version control system.

Inheritance diagram for `anise.features.versioncontrol.internals.VersionControlSystem`:

`class anise_1_1features_1_1versioncontrol_1_1internal`

Public Member Functions

- def `__init__` (self)
- def `getfullversion` (self)
Returns a complete version string.
- def `createrawpackage` (self)
Generates a raw package.
- def `syncversioncontrolsystem` (self, commitmessage="", forceskipped=True, forceunchanged=False)
Synchronizes the local working copy with the version control system.
- def `findvcs` (self)
Checks if the current project is bound to a version control system of this kind.
- def `bindvcs` (self, l)
Binds the current project to this version control system.
- def `fetchrawcommitmessage` (self, revision)
Fetches a raw commit message.
- def `getcommitmessagefromrawcommitmessage` (self, s)
Gets the actual commit message from the raw 'fetchrawcommitmessage' commit message output.
- def `getpreviousrevision` (self, revision)
Gets the predecessor revision for a given revision.
- def `getheadrevision` (self)
Gets the head revision.
- def `getcommitmessage` (self, revision)
Returns the raw commit message (as formatted by the vcs tool) for a revision.
- def `getchangelog` (self, asstring=True)
Computes the project change log from information in specially formatted vcs commit messages.
- def `addchange` (self, changetext=None)
Adds a change to the changelog.
- def `addlabel` (self, label=None)
Adds a version label for the current state.
- def `storechange` (self, changetext)
Stores a new entry to the changelog.
- def `storelabel` (self, label)
Stores a new version label.
- def `branchfromhere` (self, branchname=None)
Creates a new branch from here and switches to it.

Public Attributes

- `label`

1.372.1 Detailed Description

A version control system.

Override this for providing an own version control system.

1.372.2 Constructor & Destructor Documentation

1.372.2.1 `__init__()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.__init__ (
    self )
```

1.372.3 Member Function Documentation

1.372.3.1 `addchange()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.addchange (
    self,
    changetext = None )
```

Adds a change to the changelog.

Parameters

<i>changetext</i>	The changelog entry text.
-------------------	---------------------------

1.372.3.2 `addlabel()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.addlabel (
    self,
    label = None )
```

Adds a version label for the current state.

Parameters

<i>label</i>	The label string.
--------------	-------------------

1.372.3.3 `bindvcs()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.bindvcs (
    self,
    l )
```

Binds the current project to this version control system.

Uses user interaction for getting details like urls.

Override this method in custom subclasses or leave the default implementation.

Parameters

/	The let configuration node for storing additional parameters.
---	---

1.372.3.4 branchfromhere()

```
def anise.features.versioncontrol.internals.VersionControlSystem.branchfromhere (
    self,
    branchname = None )
```

Creates a new branch from here and switches to it.

Unpushed changes will become part of that new branch.

Override this method in custom subclasses.

Parameters

<i>branchname</i>	The branch name (None: input dialog).
-------------------	---------------------------------------

1.372.3.5 createrawpackage()

```
def anise.features.versioncontrol.internals.VersionControlSystem.createrawpackage (
    self )
```

Generates a raw package.

This package is the basis for most (or all) of the other package types.

Override this method in custom subclasses.

1.372.3.6 fetchrawcommitmessage()

```
def anise.features.versioncontrol.internals.VersionControlSystem.fetchrawcommitmessage (
    self,
    revision )
```

Fetches a raw commit message.

Do not use it directly, but `getcommitmessage` instead.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>revision</i>	The revision.
-----------------	---------------

Returns

: The raw commit message as string in a custom format.

1.372.3.7 findvcs()

```
def anise.features.versioncontrol.internals.VersionControlSystem.findvcs (
    self )
```

Checks if the current project is bound to a version control system of this kind.

Override this method in custom subclasses or leave the default implementation.

1.372.3.8 getchangelog()

```
def anise.features.versioncontrol.internals.VersionControlSystem.getchangelog (
    self,
    asstring = True )
```

Computes the project change log from information in specially formatted vcs commit messages.

Parameters

<i>asstring</i>	If to output the changelog as string (instead of an internal data structure).
-----------------	---

1.372.3.9 getcommitmessage()

```
def anise.features.versioncontrol.internals.VersionControlSystem.getcommitmessage (
    self,
    revision )
```

Returns the raw commit message (as formatted by the vcs tool) for a revision.

Parameters

<i>revision</i>	The revision.
-----------------	---------------

1.372.3.10 `getcommitmessagefromrawcommitmessage()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.getcommitmessagefromrawcommitmessage  
(  
    self,  
    s )
```

Gets the actual commit message from the raw 'fetchrawcommitmessage' commit message output.

Do not use it directly, but `getcommitmessage` instead.

Override this method in custom subclasses or leave the default implementation.

Parameters

s	The raw commit message string.
---	--------------------------------

Returns

: An instance of [CommitMessage](#).

1.372.3.11 `getfullversion()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.getfullversion (  
    self )
```

Returns a complete version string.

Override this method in custom subclasses.

1.372.3.12 `getheadrevision()`

```
def anise.features.versioncontrol.internals.VersionControlSystem.getheadrevision (  
    self )
```

Gets the head revision.

Override this method in custom subclasses or leave the default implementation.

1.372.3.13 getpreviousrevision()

```
def anise.features.versioncontrol.internals.VersionControlSystem.getpreviousrevision (
    self,
    revision )
```

Gets the predecessor revision for a given revision.

Override this method in custom subclasses or leave the default implementation.

Parameters

<i>revision</i>	A revision.
-----------------	-------------

Returns

: The predeccessing revision.

1.372.3.14 storechange()

```
def anise.features.versioncontrol.internals.VersionControlSystem.storechange (
    self,
    changetext )
```

Stores a new entry to the changelog.

Do not use this directly, but addchange instead.

Parameters

<i>changetext</i>	The changelog entry text.
-------------------	---------------------------

1.372.3.15 storelabel()

```
def anise.features.versioncontrol.internals.VersionControlSystem.storelabel (
    self,
    label )
```

Stores a new version label.

Do not use this directly, but addlabel instead.

Parameters

<i>label</i>	The label string.
--------------	-------------------

1.372.3.16 syncversioncontrolsystem()

```
def anise.features.versioncontrol.internals.VersionControlSystem.syncversioncontrolsystem (
    self,
    commitmessage = "",
    forceskipped = True,
    forceunchanged = False )
```

Synchronizes the local working copy with the version control system.

Override this method in custom subclasses.

Parameters

<i>commitmessage</i>	Commit message text.
<i>forceskipped</i>	Force synchronizing even if the user decided to skip that.
<i>forceunchanged</i>	Force synchronizing even if there are no changes to commit.

1.372.4 Member Data Documentation

1.372.4.1 label

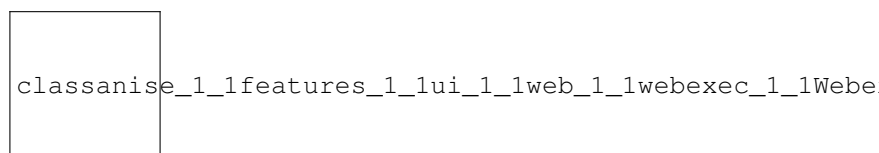
```
anise.features.versioncontrol.internals.VersionControlSystem.label
```

The documentation for this class was generated from the following file:

- [anise/features/versioncontrol.py](#)

1.373 anise.features.ui.web.webexec.WebexecApplication Class Reference

Inheritance diagram for anise.features.ui.web.webexec.WebexecApplication:



Classes

- class [AbortedInterrupt](#)

Public Member Functions

- def [__init__](#) (self, parentid=None, returntoparent=[True](#))
- def [oninitialize](#) (self)
- def [stop](#) (self)
- def [do_application_shutdown](#) (self, request)

Public Attributes

- [startmessageonlyterminal](#)
- [mainthread](#)
- [mainexecscope](#)
- [universe](#)

Private Attributes

- [_shutdownByUser](#)

1.373.1 Constructor & Destructor Documentation

1.373.1.1 [__init__\(\)](#)

```
def anise.features.ui.web.webexec.WebexecApplication.__init__ (
    self,
    parentid = None,
    returntoparent = True )
```

1.373.2 Member Function Documentation

1.373.2.1 [do_application_shutdown\(\)](#)

```
def anise.features.ui.web.webexec.WebexecApplication.do_application_shutdown (
    self,
    request )
```

1.373.2.2 oninitialize()

```
def anise.features.ui.web.webexec.WebexecApplication.oninitialize (
    self )
```

1.373.2.3 stop()

```
def anise.features.ui.web.webexec.WebexecApplication.stop (
    self )
```

1.373.3 Member Data Documentation

1.373.3.1 _shutdownByUser

```
anise.features.ui.web.webexec.WebexecApplication._shutdownByUser [private]
```

1.373.3.2 mainexecscope

```
anise.features.ui.web.webexec.WebexecApplication.mainexecscope
```

1.373.3.3 mainthread

```
anise.features.ui.web.webexec.WebexecApplication.mainthread
```

1.373.3.4 startmessageonlyterminal

```
anise.features.ui.web.webexec.WebexecApplication.startmessageonlyterminal
```

1.373.3.5 universe

```
anise.features.ui.web.webexec.WebexecApplication.universe
```

The documentation for this class was generated from the following file:

- [anise/features/ui/web/webexec.py](#)

1.374 anise.features.packages.python.WheelApplicationLink Class Reference

Specification for one program symlink added by a Python wheel package.

Public Member Functions

- `def __init__(self, linkname, module, method, gui)`

Public Attributes

- `linkname`
- `module`
- `method`
- `gui`

1.374.1 Detailed Description

Specification for one program symlink added by a Python wheel package.

1.374.2 Constructor & Destructor Documentation

1.374.2.1 __init__()

```
def anise.features.packages.python.WheelApplicationLink.__init__(  
    self,  
    linkname,  
    module,  
    method,  
    gui )
```

Parameters

<i>linkname</i>	The name of the link as the user will be able to call it.
<i>module</i>	The name of the module which contains the entry method.
<i>method</i>	The name of the entry method.
<i>gui</i>	If the command opens a gui (instead of a terminal application).

1.374.3 Member Data Documentation

1.374.3.1 gui

```
anise.features.packages.python.WheelApplicationLink.gui
```

1.374.3.2 linkname

```
anise.features.packages.python.WheelApplicationLink.linkname
```

1.374.3.3 method

```
anise.features.packages.python.WheelApplicationLink.method
```

1.374.3.4 module

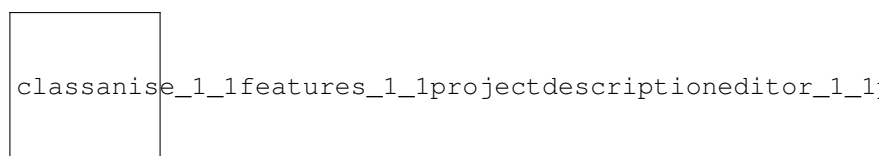
```
anise.features.packages.python.WheelApplicationLink.module
```

The documentation for this class was generated from the following file:

- [anise/features/packages/python.py](#)

1.375 anise.features.projectdescriptioneditor.packages.Win32Package Class Reference

Inheritance diagram for `anise.features.projectdescriptioneditor.packages.Win32Package`:

**Public Member Functions**

- `def __init__ (self)`
- `def ask (self)`
- `def gettaskref (self)`

Public Attributes

- [label](#)
- [taskref](#)

1.375.1 Constructor & Destructor Documentation

1.375.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.packages.Win32Package.__init__ (
    self )
```

1.375.2 Member Function Documentation

1.375.2.1 `ask()`

```
def anise.features.projectdescriptioneditor.packages.Win32Package.ask (
    self )
```

1.375.2.2 `gettaskref()`

```
def anise.features.projectdescriptioneditor.packages.AbstractPackage.gettaskref (
    self ) [inherited]
```

1.375.3 Member Data Documentation

1.375.3.1 `label`

```
anise.features.projectdescriptioneditor.packages.Win32Package.label
```

1.375.3.2 `taskref`

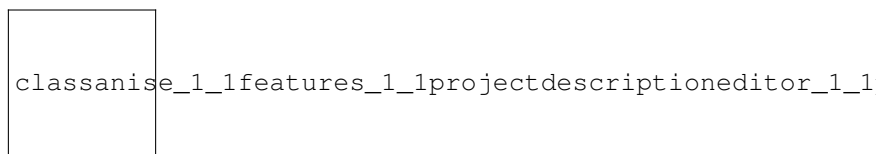
```
anise.features.projectdescriptioneditor.packages.Win32Package.taskref
```

The documentation for this class was generated from the following file:

- `anise/features/projectdescriptioneditor/packages.py`

1.376 anise.features.projectdescriptioneditor.packages.Win32PackageAddStartMenuEntryCustomAction Class Reference

Inheritance diagram for anise.features.projectdescriptioneditor.packages.Win32PackageAddStartMenuEntryCustomAction:



Public Member Functions

- def `__init__` (self)
- def `visible` (self, ais, node)
- def `execute` (self, ais, node)

Public Attributes

- `label`
- `function`
- `icon`

1.376.1 Constructor & Destructor Documentation

1.376.1.1 `__init__()`

```
def anise.features.projectdescriptioneditor.packages.Win32PackageAddStartMenuEntryCustomAction.__init__(
    self )
```

1.376.2 Member Function Documentation

1.376.2.1 `execute()`

```
def anise.features.projectdescriptioneditor.packages.Win32PackageAddStartMenuEntryCustomAction.execute (
    self,
    ais,
    node )
```

1.376.2.2 visible()

```
def anise.features.projectdescriptioneditor.packages.Win32PackageAddStartMenuEntryCustomAction.visible (
    self,
    ais,
    node )
```

1.376.3 Member Data Documentation

1.376.3.1 function

anise.features.projectdescriptioneditor.CustomAction.function [inherited]

1.376.3.2 icon

anise.features.projectdescriptioneditor.CustomAction.icon [inherited]

1.376.3.3 label

anise.features.projectdescriptioneditor.CustomAction.label [inherited]

The documentation for this class was generated from the following file:

- anise/features/projectdescriptioneditor/[packages.py](#)

