

---

# **avocado-i2n Documentation**

**Intra2net AG**

**Jul 19, 2022**



---

## Contents

---

<b>1</b>	<b>avocado_i2n package</b>	<b>1</b>
1.1	Subpackages	1
1.1.1	avocado_i2n.cartgraph package	1
1.1.1.1	Submodules	1
1.1.1.2	Module contents	3
1.1.2	avocado_i2n.plugins package	3
1.1.2.1	Submodules	3
1.1.2.2	Module contents	3
1.1.3	avocado_i2n.states package	3
1.1.3.1	Submodules	3
1.1.3.2	Module contents	4
1.1.4	avocado_i2n.vmnet package	4
1.1.4.1	Submodules	4
1.1.4.2	Module contents	4
1.2	Submodules	4
1.2.1	avocado_i2n.cmd_parser module	4
1.2.2	avocado_i2n.intertest_setup module	4
1.2.3	avocado_i2n.loader module	4
1.2.4	avocado_i2n.params_parser module	4
1.2.4.1	SUMMARY	4
1.2.4.2	INTERFACE	4
1.2.5	avocado_i2n.runner module	8
1.3	Module contents	8
	<b>Python Module Index</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



## 1.1 Subpackages

### 1.1.1 avocado\_i2n.cartgraph package

#### 1.1.1.1 Submodules

**avocado\_i2n.cartgraph.graph module**

**avocado\_i2n.cartgraph.node module**

**avocado\_i2n.cartgraph.object module**

### SUMMARY

Utility for the main test suite substructures like test objects.

Copyright: Intra2net AG

### INTERFACE

**class** avocado\_i2n.cartgraph.object.**TestObject** (*suffix, config*)

Bases: object

A wrapper for a test object used in one or more test nodes.

#### **params**

Parameters (cache) property.

#### **final\_restr**

Final restriction to make the object parsing variant unique.

**long\_suffix**

Sufficiently unique suffix to identify a variantless test object.

**id**

Unique ID to identify a test object.

**\_\_init\_\_** (*suffix, config*)

Construct a test object (vm) for any test nodes (tests).

**Parameters**

- **name** (*str*) – name of the test object
- **config** (*param.Reparsable*) – variant configuration for the test object

**\_\_repr\_\_** ()

Return repr(self).

**is\_permanent** ()

If the test object is permanent, it can only be created manually (possibly through the use of manual setup steps).

On states on permanent test object are treated differently than on states on normal test object since they are preserved through test runs and even host shutdowns.

**object\_typed\_params** (*params*)

Return object and type filtered parameters using the current object type.

**Parameters** **params** (*param\_utils.Params*) – whether to show generated parameter dictionaries

**regenerate\_params** (*verbose=False*)

Regenerate all parameters from the current reparsable config.

**Parameters** **verbose** (*bool*) – whether to show generated parameter dictionaries

**class** `avocado_i2n.cartgraph.object.NetObject` (*name, config*)

Bases: `avocado_i2n.cartgraph.object.TestObject`

A Net wrapper for a test object used in one or more test nodes.

**\_\_init\_\_** (*name, config*)

Construct a test object (vm) for any test nodes (tests).

All arguments are inherited from the base class.

**class** `avocado_i2n.cartgraph.object.VMObject` (*name, config*)

Bases: `avocado_i2n.cartgraph.object.TestObject`

A VM wrapper for a test object used in one or more test nodes.

**\_\_init\_\_** (*name, config*)

Construct a test object (vm) for any test nodes (tests).

All arguments are inherited from the base class.

**class** `avocado_i2n.cartgraph.object.ImageObject` (*name, config*)

Bases: `avocado_i2n.cartgraph.object.TestObject`

An image wrapper for a test object used in one or more test nodes.

**id**

Sufficiently unique ID to identify a test object.

```
__init__(name, config)
```

Construct a test object (vm) for any test nodes (tests).

All arguments are inherited from the base class.

#### 1.1.1.2 Module contents

### 1.1.2 avocado\_i2n.plugins package

#### 1.1.2.1 Submodules

**avocado\_i2n.plugins.auto module**

**avocado\_i2n.plugins.manu module**

**avocado\_i2n.plugins.settings module**

Avocado plugin that extends the settings path of our config paths.

```
class avocado_i2n.plugins.settings.I2NSettings
    Bases: avocado.core.plugin_interfaces.Settings

    adjust_settings_paths(paths)
        Entry point where plugin can modify the list of configuration paths.

    __abstractmethods__ = frozenset()
```

#### 1.1.2.2 Module contents

### 1.1.3 avocado\_i2n.states package

#### 1.1.3.1 Submodules

**avocado\_i2n.states.btrfs module**

**avocado\_i2n.states.lvm module**

**avocado\_i2n.states.lxc module**

**avocado\_i2n.states.pool module**

**avocado\_i2n.states.qcow2 module**

**avocado\_i2n.states.ramfile module**

**avocado\_i2n.states.setup module**

avocado\_i2n.states.vmnet module

### 1.1.3.2 Module contents

## 1.1.4 avocado\_i2n.vmnet package

### 1.1.4.1 Submodules

avocado\_i2n.vmnet.interface module

avocado\_i2n.vmnet.netconfig module

avocado\_i2n.vmnet.network module

avocado\_i2n.vmnet.node module

avocado\_i2n.vmnet.tunnel module

### 1.1.4.2 Module contents

## 1.2 Submodules

### 1.2.1 avocado\_i2n.cmd\_parser module

### 1.2.2 avocado\_i2n.intertest\_setup module

### 1.2.3 avocado\_i2n.loader module

### 1.2.4 avocado\_i2n.params\_parser module

#### 1.2.4.1 SUMMARY

Module for handling all Cartesian config parsing and making it reusable and maximally performant.

Copyright: Intra2net AG

#### 1.2.4.2 INTERFACE

**exception** avocado\_i2n.params\_parser.EmptyCartesianProduct(*message*)

Bases: Exception

Empty Cartesian product of variants

**\_\_init\_\_**(*message*)

Initialize an empty Cartesian product exception.

**Parameters** *message* (*str*) – additional message about the exception

avocado\_i2n.params\_parser.custom\_configs\_dir()

Custom directory for all config files.



```

avocado_i2n.params_parser.tests_ovrwr_file()
    Overwrite config file for all tests (nodes).

avocado_i2n.params_parser.vms_ovrwr_file()
    Overwrite config file for all vms (objects).

class avocado_i2n.params_parser.ParsedContent(content)
    Bases: object

    Class for parsed content of a general type.

    __init__(content)
        Initialize the parsed content.

    reportable_form()
        Parsed content representation used in reports of parsing steps.

        Returns resulting report-compatible string

        Return type str

        Raises NotImplementedError as this is an abstract method

    parsable_form()
        Convert parameter content into parsable string.

        Returns resulting parsable string

        Return type str

        Raises NotImplementedError as this is an abstract method

class avocado_i2n.params_parser.ParsedFile(content)
    Bases: avocado_i2n.params_parser.ParsedContent

    Class for parsed content of file type.

    __init__(content)
        Initialize the parsed content.

    reportable_form()
        Parsed file representation used in reports of parsing steps.

        Arguments are identical to the ones of the parent class.

    parsable_form()
        Convert parameter file name into parsable string.

        Returns resulting parsable string

        Return type str

class avocado_i2n.params_parser.ParsedStr(content)
    Bases: avocado_i2n.params_parser.ParsedContent

    Class for parsed content of string type.

    reportable_form()
        Parsed string representation used in reports of parsing steps.

        Arguments are identical to the ones of the parent class.

    parsable_form()
        Convert parameter string into parsable string.

        Returns resulting parsable string

```

**Return type** str

This is equivalent to the string since the string is parsable by definition.

**class** avocado\_i2n.params\_parser.**ParsedDict** (*content*)

Bases: *avocado\_i2n.params\_parser.ParsedContent*

Class for parsed content of dictionary type.

**reportable\_form**()

Parsed dictionary representation used in reports of parsing steps.

Arguments are identical to the ones of the parent class.

**parsable\_form**()

Convert parameter dictionary into parsable string.

**Returns** resulting parsable string

**Return type** str

**class** avocado\_i2n.params\_parser.**Reparsable**

Bases: object

Class to represent quickly parsable Cartesian configuration, producing both parser and parameters (parser dicts) on demand.

**\_\_init\_\_**()

Initialize the parsable structure.

**parse\_next\_file** (*pfile*)

Add a file parsing step.

**Parameters** **pfile** (*str*) – file to be parsed next

If the parsable file has a relative form (not an absolute path), it will be searched in the relative test suite config directory.

**parse\_next\_str** (*pstring*)

Add a string parsing step.

**Parameters** **pstring** (*str*) – string to be parsed next

**parse\_next\_dict** (*pdict*)

Add a dictionary parsing step.

**Parameters** **pdict** (*{str, str}*) – dictionary to be parsed next

**parse\_next\_batch** (*base\_file=None, base\_str="", base\_dict=None, ovrwrt\_file=None, ovrwrt\_str="", ovrwrt\_dict=None*)

Parse a batch of base file, string, and dictionary, and possibly an overwrite file (with custom parameters at the user's home location).

**Parameters**

- **base\_file** (*str or None*) – file to be parsed first
- **base\_str** (*str or None*) – string to be parsed first
- **base\_dict** (*{str, str} or None*) – params to be added first
- **ovrwrt\_file** (*str or None*) – file to be parsed last
- **ovrwrt\_str** (*str or None*) – string to be parsed last
- **ovrwrt\_dict** (*{str, str} or None*) – params to be added last

The priority of the setting follows the order of the arguments: Dictionary with some parameters is topmost, string with some parameters is next and the file with parameters is taken as a base. The overwriting version is taken last, the base version first.

**print\_parsed()**

Return printable information about what was parsed so far.

**Returns** structured text of the base/ovrwr file/str/dict parse steps

**Return type** str

`avocado_i2n.params_parser.all_restrictions()`

Return all restrictions that can be passed for any test configuration.

**Returns** all available (from configuration) vms

**Return type** [str]

`avocado_i2n.params_parser.all_objects(key='vms', composites=None)`

Return all test objects that can be passed for any test configuration.

**Param** str key: key to extract parametric objects from

**Parameters** **composites** ([str]) – composite restriction of the returned objects

**Returns** all available (from configuration) objects of a given type

**Return type** [str]

`avocado_i2n.params_parser.main_vm()`

Return the default main vm that can be passed for any test configuration.

**Returns** main available (from configuration) vm

**Return type** str or None

`avocado_i2n.params_parser.re_str(variant_str, base_str="", tag="")`

Add a variant restriction to the base string, optionally adding a custom tag as well.

**Parameters**

- **variant\_str** (str) – variant restriction
- **base\_str** (str) – string where the variant restriction will be added
- **tag** (str) – additional tag to the variant combination

**Returns** restricted parameter string

**Return type** str

`avocado_i2n.params_parser.join_str(variant_strs, base_str="")`

Join all object variant restrictions over the base string.

**Parameters**

- **variant\_strs** ({str, str}) – variant restrictions for each object as key, value pair
- **base\_str** (str) – string where the variant restriction will be added

**Returns** restricted parameter string

**Return type** str

### 1.2.5 avocado\_i2n.runner module

## 1.3 Module contents

### a

- `avocado_i2n`, 8
- `avocado_i2n.cartgraph.object`, 1
- `avocado_i2n.params_parser`, 4
- `avocado_i2n.plugins`, 3
- `avocado_i2n.plugins.settings`, 3
- `avocado_i2n.states`, 4



## Symbols

- `__abstractmethods__` (avocado\_i2n.plugins.settings.I2NSettings attribute), 3
  - `__init__()` (avocado\_i2n.cartgraph.object.ImageObject method), 2
  - `__init__()` (avocado\_i2n.cartgraph.object.NetObject method), 2
  - `__init__()` (avocado\_i2n.cartgraph.object.TestObject method), 2
  - `__init__()` (avocado\_i2n.cartgraph.object.VMObject method), 2
  - `__init__()` (avocado\_i2n.params\_parser.EmptyCartesianProduct method), 4
  - `__init__()` (avocado\_i2n.params\_parser.ParsedContent method), 5
  - `__init__()` (avocado\_i2n.params\_parser.ParsedFile method), 5
  - `__init__()` (avocado\_i2n.params\_parser.Reparsable method), 6
  - `__repr__()` (avocado\_i2n.cartgraph.object.TestObject method), 2
- ## A
- `adjust_settings_paths()` (avocado\_i2n.plugins.settings.I2NSettings method), 3
  - `all_objects()` (in module avocado\_i2n.params\_parser), 7
  - `all_restrictions()` (in module avocado\_i2n.params\_parser), 7
  - `avocado_i2n` (module), 8
  - `avocado_i2n.cartgraph.object` (module), 1
  - `avocado_i2n.params_parser` (module), 4
  - `avocado_i2n.plugins` (module), 3
  - `avocado_i2n.plugins.settings` (module), 3
  - `avocado_i2n.states` (module), 4
- ## C
- `custom_configs_dir()` (in module avocado\_i2n.params\_parser), 4
- ## E
- `EmptyCartesianProduct`, 4
- ## F
- `final_restr` (avocado\_i2n.cartgraph.object.TestObject attribute), 1
- ## I
- `I2NSettings` (class in avocado\_i2n.plugins.settings), 3
  - `id` (avocado\_i2n.cartgraph.object.ImageObject attribute), 2
  - `id` (avocado\_i2n.cartgraph.object.TestObject attribute), 2
  - `ImageObject` (class in avocado\_i2n.cartgraph.object), 2
  - `is_permanent()` (avocado\_i2n.cartgraph.object.TestObject method), 2
- ## J
- `join_str()` (in module avocado\_i2n.params\_parser), 7
- ## L
- `long_suffix` (avocado\_i2n.cartgraph.object.TestObject attribute), 1
- ## M
- `main_vm()` (in module avocado\_i2n.params\_parser), 7
- ## N
- `NetObject` (class in avocado\_i2n.cartgraph.object), 2

**O**

`object_typed_params()` (avocado\_i2n.cartgraph.object.TestObject method), 2

**P**

`params` (avocado\_i2n.cartgraph.object.TestObject attribute), 1

`parsable_form()` (avocado\_i2n.params\_parser.ParsedContent method), 5

`parsable_form()` (avocado\_i2n.params\_parser.ParsedDict method), 6

`parsable_form()` (avocado\_i2n.params\_parser.ParsedFile method), 5

`parsable_form()` (avocado\_i2n.params\_parser.ParsedStr method), 5

`parse_next_batch()` (avocado\_i2n.params\_parser.Reparsable method), 6

`parse_next_dict()` (avocado\_i2n.params\_parser.Reparsable method), 6

`parse_next_file()` (avocado\_i2n.params\_parser.Reparsable method), 6

`parse_next_str()` (avocado\_i2n.params\_parser.Reparsable method), 6

`ParsedContent` (class in avocado\_i2n.params\_parser), 5

`ParsedDict` (class in avocado\_i2n.params\_parser), 6

`ParsedFile` (class in avocado\_i2n.params\_parser), 5

`ParsedStr` (class in avocado\_i2n.params\_parser), 5

`print_parsed()` (avocado\_i2n.params\_parser.Reparsable method), 7

**R**

`re_str()` (in module avocado\_i2n.params\_parser), 7

`regenerate_params()` (avocado\_i2n.cartgraph.object.TestObject method), 2

`Reparsable` (class in avocado\_i2n.params\_parser), 6

`reportable_form()` (avocado\_i2n.params\_parser.ParsedContent method), 5

`reportable_form()` (avocado\_i2n.params\_parser.ParsedDict method), 6

`reportable_form()` (avocado\_i2n.params\_parser.ParsedFile method), 5

`reportable_form()` (avocado\_i2n.params\_parser.ParsedStr method), 5

**T**

`TestObject` (class in avocado\_i2n.cartgraph.object), 1

`tests_ovrwr_file()` (in module avocado\_i2n.params\_parser), 4

**V**

`VMObject` (class in avocado\_i2n.cartgraph.object), 2

`vms_ovrwr_file()` (in module avocado\_i2n.params\_parser), 5