
ats_utilities Documentation

Release

https://github.com/vroncevic/ats_utilities/releases

Vladimir Roncevic <elektron.ronca@gmail.com>

Apr 18, 2022

1	ats_utilities	3
1.1	ats_utilities package	3
1.1.1	Subpackages	3
1.1.1.1	ats_utilities.abstract package	3
1.1.1.2	ats_utilities.checker package	4
1.1.1.3	ats_utilities.cli package	6
1.1.1.4	ats_utilities.config_io package	11
1.1.1.5	ats_utilities.console_io package	27
1.1.1.6	ats_utilities.cooperative package	31
1.1.1.7	ats_utilities.exceptions package	32
1.1.1.8	ats_utilities.final package	36
1.1.1.9	ats_utilities.info package	37
1.1.1.10	ats_utilities.logging package	42
1.1.1.11	ats_utilities.option package	45
1.1.1.12	ats_utilities.register package	46
1.1.1.13	ats_utilities.singleton package	47
1.1.1.14	ats_utilities.splash package	49
1.1.2	Module contents	55
2	Installation	57
3	Dependencies	59
4	Framework structure	61
5	Copyright and licence	65
6	Indices and tables	67
	Python Module Index	69
	Index	71

ats_utilities is framework for creating Apps/Tools/Scripts.

Developed in `python` code.

The README is used to introduce the modules and provide instructions on how to install the modules, any machine dependencies it may have and any other information that should be provided before the modules are installed.

1.1 ats_utilities package

1.1.1 Subpackages

1.1.1.1 ats_utilities.abstract package

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `AbstractMethod` with attribute(s) and method(s). Created abstract decorator for object methods.

class `ats_utilities.abstract.AbstractMethod` (*method_to_abstract*)
Bases: `object`

Defined class `AbstractMethod` with attribute(s) and method(s). Created abstract decorator for object methods.
It defines:

attributes

- `method_name` - object method name.
- `method_class_name` - class name of method.
- `method_type` - type of object method.
- `method` - method object-reference.

methods

`__init__` - initial constructor.
`__call__` - raise exception `NotImplementedError` on call.
`__str__` - str dunder method for `AbstractMethod`.

1.1.1.2 ats_utilities.checker package

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `ATSChecker` with attribute(s) and method(s). Created API fo checking parameters for object methods and functions.

class `ats_utilities.checker.ATSChecker`

Bases: `object`

Defined class `ATSChecker` with attribute(s) and method(s). Created API fo checking parameters for object methods and functions. It defines:

attributes

`NO_ERROR` - no error, error id (0).
`TYPE_ERROR` - type param error id (1).
`VALUE_ERROR` - value param error id (2).
`FORMAT_ERROR` - wrong format error id (3).
`__start_message` - start segment of usage message.
`__list_of_params` - list of parameters for method/function.
`__error_type` - list of mapped errors.
`__error_type_index` - error type index.
`__error_value_index` - error value index.

methods

`__init__` - initial constructor.
`collect_params` - collect all parameters in one list.
`usage_message` - prepare usage message for method/function.
`check_types` - check parameters (types) for method/function.
`check_values` - check parameters (values) for method/function.
`priority_error` - set priority error id (`TYPE_ERROR`).
`check_params` - check parameters for method/function.
`__str__` - str dunder method for object `ATSChecker`.

`FORMAT_ERROR = 3`

`NO_ERROR = 0`

TYPE_ERROR = 1

VALUE_ERROR = 2

check_params (*params_description*)

Check parameters for method/function.

Parameters **params_description** (<list>) – parameters description.

Returns usage message, status (0 | 1 | 2 | 3).

Return type <str>, <int>

Exceptions None

check_types (*params_description*)

Check parameters (types) for method/function.

Parameters **params_description** (<OrderedDict>) – parameters description.

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

check_values (*params_description*)

Check parameters (values) for method/function.

Parameters **params_description** (<OrderedDict>) – parameters description.

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

collect_params (*params_description*)

Collect all parameters in one list.

Parameters

- **params_description** (<OrderedDict>) – description for parameters.
- **verbose** (<bool>) – enable/disable verbose option.

Returns True (format for params ok) | False.

Returns boolean status, True (ok) | False.

Return type <bool>

Exceptions None

priority_error ()

Set priority error id (TYPE_ERROR).

Returns priority error id (0 | 1 | 2 | 3) | None.

Return type <int> | <NoneType>

Exceptions None

usage_message ()

Prepare usage message for method/function.

Returns usage message for method/function.

Return type <str>

Exceptions None

1.1.1.3 ats_utilities.cli package

Submodules

ats_utilities.cli.cfg_cli module

Module `cfg_cli.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `CfgCLI` with attribute(s) and method(s). Created API for check and load informations, setup argument parser.

class `ats_utilities.cli.cfg_cli.CfgCLI` (*informations_file, verbose=False*)
Bases: `ats_utilities.config_io.cfg.CfgBase`

Defined class `CfgCLI` with attribute(s) and method(s). Created API for check and load informations, setup arguments parser. It defines:

attributes

`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.

`add_new_option` - adding new option for CL interface.

`parse_args` - parse arguments.

`process` - process and run tool operation (Abstract method).

`__str__` - str dunder method for `CfgCLI`.

VERBOSE = `'ats_utilities::cli::cfg_cli'`

add_new_option (**args, **kwargs*)
Adding new option for CL interface.

Parameters

- **args** (<list>) – list of arguments (objects).
- **kwargs** (<dict>) – arguments in shape of dictionary.

Exceptions None

parse_args (*argv*)
Process arguments from start.

Parameters **argv** (<list>) – arguments.

Returns options and arguments.

Return type <Python object(s)>

Exceptions None

```
process = <ats_utilities.abstract.AbstractMethod object>
```

ats_utilities.cli.ini_cli module

Module ini_cli.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class IniCLI with attribute(s) and method(s). Created API for check and load informations, setup argument parser.

class ats_utilities.cli.ini_cli.IniCLI (*informations_file, verbose=False*)
Bases: *ats_utilities.config_io.ini.IniBase*

Defined class IniCLI with attribute(s) and method(s). Created API for check and load informations, setup argument parser. It defines:

attributes

__verbose - enable/disable verbose option.

methods

__init__ - initial constructor.

add_new_option - adding new option for CL interface.

parse_args - parse arguments.

process - process and run tool operation (Abstract method).

__str__ - str dunder method for IniCLI.

```
VERBOSE = 'ats_utilities::cli::ini_cli'
```

```
add_new_option (*args, **kwargs)  
Adding new option for CL interface.
```

Parameters

- **args** (<list>) – list of arguments (objects).
- **kwargs** (<dict>) – arguments in shape of dictionary.

Exceptions None

```
parse_args (argv)  
Process arguments from start.
```

Parameters **argv** (<list>) – arguments.

Returns options and arguments.

Return type <Python object(s)>

Exceptions None

```
process = <ats_utilities.abstract.AbstractMethod object>
```

ats_utilities.cli.json_cli module

Module json_cli.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class JsonCLI with attribute(s) and method(s). Created API for check and load informations, setup argument parser.

class ats_utilities.cli.json_cli.JsonCLI (*informations_file, verbose=False*)
Bases: *ats_utilities.config_io.json.JsonBase*

Defined class JsonCLI with attribute(s) and method(s). Created API for check and load informations, setup argument parser. It defines:

attributes

__verbose - enable/disable verbose option.

methods

__init__ - initial constructor.

add_new_option - adding new option for CL interface.

parse_args - parse arguments.

process - process and run tool operation (Abstract method).

__str__ - str dunder method for JsonCLI.

VERBOSE = 'ats_utilities::cli::json_cli'

add_new_option (**args, **kwargs*)
Adding new option for CL interface.

Parameters

- **args** (<list>) – list of arguments (objects).
- **kwargs** (<dict>) – arguments in shape of dictionary.

Exceptions None

parse_args (*argv*)
Process arguments from start.

Parameters **argv** (<list>) – arguments.

Returns options and arguments.

Return type <Python object(s)>

Exceptions None

process = <ats_utilities.abstract.AbstractMethod object>

ats_utilities.cli.xml_cli module

Module xml_cli.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class XmlCLI with attribute(s) and method(s). Created API for check and load informations, setup argument parser.

class `ats_utilities.cli.xml_cli.XmlCLI` (*informations_file, verbose=False*)
Bases: `ats_utilities.config_io.xml.XmlBase`

Defined class XmlCLI with attribute(s) and method(s). Created API for check and load informations, setup argument parser. It defines:

attributes

`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.

`add_new_option` - adding new option for CL interface.

`parse_args` - parse arguments.

`process` - process and run tool operation (Abstract method).

`__str__` - str dunder method for XmlCLI.

VERBOSE = `'ats_utilities::cli::xml_cli'`

add_new_option (**args, **kwargs*)
Adding new option for CL interface.

Parameters

- **args** (<list>) – list of arguments (objects).
- **kwargs** (<dict>) – arguments in shape of dictionary.

Exceptions None

parse_args (*argv*)
Process arguments from start.

Parameters **argv** (<list>) – arguments.

Returns options and arguments.

Return type <Python object(s)>

Exceptions None

process = <ats_utilities.abstract.AbstractMethod object>

ats_utilities.cli.yaml_cli module

Module `yaml_cli.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities

is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class `YamlCLI` with attribute(s) and method(s). Created API for check and load informations, setup argument parser.

class `ats_utilities.cli.yaml_cli.YamlCLI` (*informations_file, verbose=False*)
Bases: `ats_utilities.config_io.yaml.YamlBase`

Defined class `YamlCLI` with attribute(s) and method(s). Created API for check and load informations, setup argument parser. It defines:

attributes

`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.

`add_new_option` - adding new option for CL interface.

`parse_args` - parse arguments.

`process` - process and run tool operation (Abstract method).

`__str__` - str dunder method for `YamlCLI`.

VERBOSE = `'ats_utilities::cli::yaml_cli'`

add_new_option (**args, **kwargs*)
Adding new option for CL interface.

Parameters

- **args** (*<list>*) – list of arguments (objects).
- **kwargs** (*<dict>*) – arguments in shape of dictionary.

Exceptions None

parse_args (*argv*)
Process arguments from start.

Parameters **argv** (*<list>*) – arguments.

Returns options and arguments.

Return type *<Python object(s)>*

Exceptions None

process = *<ats_utilities.abstract.AbstractMethod object>*

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic elektron.ronca@gmail.com `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General

Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined package `ats_utilities.cli`.

1.1.1.4 `ats_utilities.config_io` package

Subpackages

`ats_utilities.config_io.cfg` package

Submodules

`ats_utilities.config_io.cfg.cfg2object` module

Module `cfg2object.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class `Cfg2Object` with attribute(s) and method(s). Created API for reading configuration/information from a `cfg` file.

class `ats_utilities.config_io.cfg.cfg2object.Cfg2Object` (*configuration_file*, *verbose=False*)

Bases: `ats_utilities.config_io.base_read.BaseReadConfig`

Defined class `Cfg2Object` with attribute(s) and method(s). Created API for read configuration/information from a `cfg` file. It defines:

attributes

`__FORMAT` - format of configuration content.
`__REGEX_MATCH_LINE` - regular expression for matching line.
`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.
`read_configuration` - read configuration from file.
`__str__` - str dunder method for object `Cfg2Object`.

`VERBOSE = 'ats_utilities::config_io::cfg::cfg2object'`

`_Cfg2Object__FORMAT = 'cfg'`

`_Cfg2Object__REGEX_MATCH_LINE = '^\\s*$'`

`read_configuration` (*verbose=False*)

Getting a configuration from `cfg` file.

Parameters `verbose` (<*bool*>) – enable/disable verbose option.

Returns configuration object | None.

Return type <dict> | <NoneType>

Exceptions None

ats_utilities.config_io.cfg.object2cfg module

Module object2cfg.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class Object2Cfg with attribute(s) and method(s). Created API for writing configuration/information to a cfg file.

class `ats_utilities.config_io.cfg.object2cfg.Object2Cfg` (*configuration_file*, *verbose=False*)

Bases: `ats_utilities.config_io.base_write.BaseWriteConfig`

Defined class Object2Cfg with attribute(s) and method(s). Created API for writing configuration/information to a cfg file. It defines:

attributes

`__FORMAT` - format of configuration content.

`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.

`write_configuration` - write config to a cfg file.

`__str__` - str dunder method for object Object2Cfg.

`VERBOSE = 'ats_utilities::config_io::cfg::object2cfg'`

`__Object2Cfg__FORMAT = 'cfg'`

`write_configuration` (*configuration*, *verbose=False*)

Write configuration to a cfg file.

Parameters

- **configuration** (<Python object(s)> | <NoneType>) – configuration object | None.
- **verbose** (<bool>) – enable/disable verbose option.

Returns boolean status, True (success) | False.

Return type <bool>

Exceptions None

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class CfgBase with attribute(s) and method(s). Load ATS configuration/information, setup ATS CL interface.

class ats_utilities.config_io.cfg.CfgBase (*informations_file, verbose=False*)
Bases: object

Defined class CfgBase with attribute(s) and method(s). Load ATS configuration/information, setup ATS CL interface. It defines:

attributes

__verbose - enable/disable verbose option.
tool_operational - control ATS operational functionality.
cfg2obj - in API for informations.
obj2cfg - out API for informations.
option_parser - option parser for ATS.

methods

__init__ - initial constructor.
is_tool_ok - checking is tool operational.
__str__ - str dunder method for object CfgBase.

VERBOSE = 'ats_utilities::config_io::cfg'

is_tool_ok ()

Checking is tool operational.

Returns boolean status, True (yes) | False.

Return type <bool>

Exceptions None

ats_utilities.config_io.ini package

Submodules

ats_utilities.config_io.ini.ini2object module

Module ini2object.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class Ini2Object with attribute(s) and method(s). Created API for reading configuration/information from an ini file.

```
class ats_utilities.config_io.ini.ini2object.Ini2Object (configuration_file, verbose=False)
Bases: ats_utilities.config_io.base_read.BaseReadConfig
```

Defined class Ini2Object with attribute(s) and method(s). Created API for reading configuration/information from an ini file. It defines:

attributes

__FORMAT - format of configuration content.
__verbose - enable/disable verbose option.

methods

__init__ - initial constructor.
read_configuration - read configuration from file.
__str__ - str dunder method for object Ini2Object.

```
VERBOSE = 'ats_utilities::config_io::ini::ini2object'
```

```
_Ini2Object__FORMAT = 'ini'
```

```
read_configuration (verbose=False)
Getting a configuration from an ini file.
```

Parameters *verbose* (<bool>) – enable/disable verbose option.

Returns configuration object | None.

Return type <ConfigParser> | <NoneType>

ats_utilities.config_io.ini.object2ini module

Module object2ini.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class Object2Ini with attribute(s) and method(s). Created API for writing configuration/information to an ini file.

```
class ats_utilities.config_io.ini.object2ini.Object2Ini (configuration_file, verbose=False)
Bases: ats_utilities.config_io.base_write.BaseWriteConfig
```

Defined class Object2Ini with attribute(s) and method(s). Created API for writing configuration/information to an ini file. It defines:

attributes

__FORMAT - format of configuration content.
__verbose - enable/disable verbose option.

methods

`__init__` - initial constructor.
`write_configuration` - write configuration to an ini file.
`__str__` - str dunder method for object `Object2Ini`.

```
VERBOSE = 'ats_utilities::config_io::ini::object2ini'
```

```
_Object2Ini__FORMAT = 'ini'
```

```
write_configuration(configuration, verbose=False)
```

Write configuration to a ini file.

Parameters

- **configuration** (*<Python object(s)> | <NoneType>*) – configuration object | None.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns boolean status, True (success) | False.

Return type *<bool>*

Exception None

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic elektron.ronca@gmail.com ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class `IniBase` with attribute(s) and method(s). Load ATS configuration/information, setup ATS CL interface.

```
class ats_utilities.config_io.ini.IniBase(informations_file, verbose=False)
```

Bases: object

Defined class `IniBase` with attribute(s) and method(s). Load ATS configuration/information, setup ATS CL interface. It defines:

attributes

`__verbose` - enable/disable verbose option.
`tool_operational` - control ATS operational functionality.
`ini2obj` - in API for informations.
`obj2ini` - out API for informations.
`option_parser` - option parser for ATS.

methods

`__init__` - initial constructor.
`is_tool_ok` - checking is tool operational.
`__str__` - str dunder method for object `IniBase`.

```
VERBOSE = 'ats_utilities::config_io::ini'
```

`is_tool_ok()`

Checking is tool operational.

Returns boolean statusm True (yes) | False.

Return type <bool>

Exceptions None

ats_utilities.config_io.json package

Submodules

ats_utilities.config_io.json.json2object module

Module json2object.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class Json2Object with attribute(s) and method(s). Created API for reading a configuration/information from a json file.

class `ats_utilities.config_io.json.json2object.Json2Object` (*configuration_file*,
verbose=False)
Bases: `ats_utilities.config_io.base_read.BaseReadConfig`

Defined class Json2Object with attribute(s) and method(s). Created API for reading a configuration/information from a json file. It defines:

attributes

`__FORMAT` - format of configuration content.

`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.

`read_configuration` - read configuration from file.

`__str__` - str dunder method for object Json2Object.

VERBOSE = `'ats_utilities::config_io::json::json2object'`

__Json2Object__FORMAT = `'json'`

read_configuration (*verbose=False*)

Read configuration from a json file.

Parameters `verbose` (<bool>) – enable/disable verbose option.

Returns configuration object | None.

Return type <Python object(s)> | <NoneType>

Exceptions None

ats_utilities.config_io.json.object2json module

Module object2json.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class Object2Json with attribute(s) and method(s). Created API for writing a configuration/information to a json file.

class ats_utilities.config_io.json.object2json.**Object2Json** (*configuration_file*,
verbose=False)
Bases: *ats_utilities.config_io.base_write.BaseWriteConfig*

Defined class Object2Json with attribute(s) and method(s). Created API for writing a configuration/information to a json file. It defines:

attributes

`__FORMAT` - format of configuration content.
`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.
`write_configuration` - write configuration to a json file.
`__str__` - str dunder method for object Object2Json.

`VERBOSE = 'ats_utilities::config_io::json::object2json'`

`__Object2Json__FORMAT = 'json'`

`write_configuration` (*configuration*, *verbose=False*)
Write configuration to a json file.

Parameters

- **verbose** (<bool>) – enable/disable verbose option.
- **configuration** – configuration object.

Type <Python object(s)>

Returns boolean status, True (success) | False.

Return type <bool>

Exception None

Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied

warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class JsonBase with attribute(s) and method(s). Load ATS configuration/information, setup ATS CL interface.

class `ats_utilities.config_io.json.JsonBase` (*informations_file, verbose=False*)
Bases: `object`

Defined class JsonBase with attribute(s) and method(s). Load ATS configuration/information, setup ATS CL interface. It defines:

attributes

`__verbose` - enable/disable verbose option.
`tool_operational` - control ATS operational functionality.
`json2obj` - in API for informations.
`obj2json` - out API for informations.
`option_parser` - option parser for ATS.

methods

`__init__` - initial constructor.
`is_tool_ok` - checking is tool operational.
`__str__` - str dunder method for object JsonBase.

VERBOSE = `'ats_utilities::config_io::json'`

is_tool_ok ()

Checking is tool operational.

Returns boolean status, True (yes) | False.

Return type <bool>

Exceptions None

ats_utilities.config_io.xml package

Submodules

ats_utilities.config_io.xml.object2xml module

Module `object2xml.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class Object2Xml with attribute(s) and method(s). Created API for writing a configuration/information to a xml file.

```
class ats_utilities.config_io.xml.object2xml.Object2Xml (configuration_file, verbose=False)
```

Bases: *ats_utilities.config_io.base_write.BaseWriteConfig*

Defined class Object2Xml with attribute(s) and method(s). Created API for writing a configuration/information to a xml file. It defines:

attributes

`__FORMAT` - format of configuration content.
`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.
`write_configuration` - write configuration to a xml file.
`__str__` - str dunder method for object Object2Xml.

```
VERBOSE = 'ats_utilities::config_io::xml::object2xml'
```

```
_Object2Xml__FORMAT = 'xml'
```

```
write_configuration (configuration, verbose=False)
```

Write configuration to a xml file.

Parameters

- **configuration** – configuration object.
- **verbose** (<bool>) – enable/disable verbose option.

Type <BeautifulSoup>

Returns boolean status, True (success) | False.

Return type <bool>

Exception None

ats_utilities.config_io.xml.xml2object module

Module xml2object.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class Xml2Object with attribute(s) and method(s). Created API for reading a configuration/information from a xml file.

```
class ats_utilities.config_io.xml.xml2object.Xml2Object (configuration_file, verbose=False)
```

Bases: *ats_utilities.config_io.base_read.BaseReadConfig*

Defined class Xml2Object with attribute(s) and method(s). Created API for reading a configuration/information from a xml file. It defines:

attributes

`__FORMAT` - format of configuration content.

`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.

`read_configuration` - read a configuration from file.

`__str__` - str dunder method for object `Xml2Object`.

```
VERBOSE = 'ats_utilities::config_io::xml::xml2object'
```

```
_Xml2Object__FORMAT = 'xml'
```

```
read_configuration(verbose=False)
```

Read a configuration from a xml file.

Parameters `verbose` (<bool>) – enable/disable verbose option.

Returns configuration object | None.

Return type <BeautifulSoup> | <NoneType>

Exceptions None

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `XmlBase` with attribute(s) and method(s). Load ATS configuration/information, setup ATS CL interface.

```
class ats_utilities.config_io.xml.XmlBase(informations_file, verbose=False)
```

Bases: object

Defined class `XmlBase` with attribute(s) and method(s). Load ATS configuration/information, setup ATS CL interface. It defines:

attributes

`__verbose` - enable/disable verbose option.

`tool_operational` - control ATS operational functionality.

`xml2obj` - in API for informations.

`obj2xml` - out API for informations.

`option_parser` - option parser for ATS.

methods

`__init__` - initial constructor.

`is_tool_ok` - checking is tool operational.

`__str__` - str dunder method for object `XmlBase`.

```
VERBOSE = 'ats_utilities::config_io::xml'
```


`is_tool_ok()`

Checking is tool operational.

Returns boolean status, True (yes) | False.

Return type <bool>

Exceptions None

ats_utilities.config_io.yaml package

Submodules

ats_utilities.config_io.yaml.object2yaml module

Module object2yaml.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class Object2Yaml with attribute(s) and method(s). Created API for writing a configuration/information to a yaml file.

class `ats_utilities.config_io.yaml.object2yaml.Object2Yaml` (*configuration_file*,
verbose=False)
Bases: `ats_utilities.config_io.base_write.BaseWriteConfig`

Defined class Object2Yaml with attribute(s) and method(s). Created API for writing a configuration/information to a yaml file. It defines:

attributes

`__FORMAT` - format of configuration content.

`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.

`write_configuration` - write configuration to a yaml file.

`__str__` - str dunder method for object Object2Yaml.

`VERBOSE = 'ats_utilities::config_io::yaml::object2yaml'`

`_Object2Yaml__FORMAT = 'yaml'`

`write_configuration` (*configuration*, *verbose=False*)

Write configuration to a yaml file.

Parameters

- **configuration** (<Python object(s)> | <NoneType>) – configuration object | None.

- **verbose** (<bool>) – enable/disable verbose option.

Returns boolean status, True (success) | False.

Return type <bool>

Exception None

ats_utilities.config_io.yaml.yaml2object module

Module yaml2object.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class Yaml2Object with attribute(s) and method(s). Created API for reading a configuration/information from a yaml file.

class ats_utilities.config_io.yaml.yaml2object.Yaml2Object (*configuration_file*,
verbose=False)

Bases: *ats_utilities.config_io.base_read.BaseReadConfig*

Defined class Yaml2Object with attribute(s) and method(s). Created API for reading a configuration/information from a yaml file. It defines:

attributes

`__FORMAT` - format of configuration content.

`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.

`read_configuration` - getting a configuration from file.

`__str__` - str dunder method for object Yaml2Object.

VERBOSE = 'ats_utilities::config_io::yaml::yaml2object'

__Yaml2Object__FORMAT = 'yaml'

read_configuration (*verbose=False*)

Getting a configuration from yaml file.

Parameters **verbose** (<bool>) – enable/disable verbose option.

Returns configuration object | None.

Return type <Python object(s)> | <NoneType>

Exceptions None

Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied

warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class `YamlBase` with attribute(s) and method(s). Load ATS information/configuration, setup ATS CL interface.

class `ats_utilities.config_io.yaml.YamlBase` (*informations_file, verbose=False*)
Bases: `object`

Defined class `YamlBase` with attribute(s) and method(s). Load ATS information/configuration, setup ATS CL interface. It defines:

attributes

`__verbose` - enable/disable verbose option.
`tool_operational` - control ATS operational functionality.
`yaml2obj` - in API for informations.
`obj2yaml` - out API for informations.
`option_parser` - option parser for ATS.

methods

`__init__` - initial constructor.
`is_tool_ok` - checking is tool operational.
`__str__` - str dunder method for object `YamlBase`.

VERBOSE = `'ats_utilities::config_io::yaml'`

is_tool_ok ()

Checking is tool operational.

Returns boolean status, True (yes) | False.

Return type `<bool>`

Exceptions None

Submodules

ats_utilities.config_io.base_check module

Module `base_check.py`

Copyright Copyright (C) 2017 Vladimir Roncevic elektron.ronca@gmail.com `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class `FileChecking` with attribute(s) and method(s). Created API for checking operations with files.

class `ats_utilities.config_io.base_check.FileChecking` (*verbose=False*)
Bases: `object`

Defined class `FileChecking` with attribute(s) and method(s). Created API for checking operations with files. It defines:

attributes

MODES - mode file operations.
__verbose - enable/disable verbose option.
__file_path_ok - file exist, path ok.
__file_mode_ok - supported file mode.
__file_format_ok - file format is (not) expected.

methods

__init__ - initial constructor.
check_path - check file path.
check_mode - check operation mode for file.
check_format - check file format by extension.
is_file_ok - status of file for processing.
__str__ - str dunder method for FileChecking.

MODES = ['r', 'w', 'a', 'b', 'x', 't', '+']

VERBOSE = 'ats_utilities::config_io::base_check'

check_format (*file_path*, *file_format*, *verbose=False*)
Check file format by extension.

Parameters

- **file_path** (<str>) – file path.
- **file_format** (<str>) – file format (file extension).
- **verbose** (<bool>) – enable/disable verbose option.

Exceptions None

check_mode (*file_mode*, *verbose=False*)
Check operation mode for file.

Parameters

- **file_mode** (<str>) – file mode ('r', 'w', 'a', 'b', 'x', 't', '+').
- **verbose** (<bool>) – enable/disable verbose option.

Returns True (regular file mode) | False.

Return type <bool>

Exceptions None

check_path (*file_path*, *verbose=False*)
Check file path.

Parameters

- **file_path** (<str>) – file path.
- **verbose** (<bool>) – enable/disable verbose option.

Exceptions None

is_file_ok ()
Status of file for processing.

Returns boolean status, True (correct file) | False.

Return type <bool>

Exceptions None

ats_utilities.config_io.base_read module

Module base_read.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class BaseReadConfig with attribute(s) and method(s). Created API for read operation for information/configuration files.

class ats_utilities.config_io.base_read.**BaseReadConfig** (*verbose=False*)
Bases: object

Defined class BaseReadConfig with attribute(s) and method(s). Created API for read operation for information/configuration files. It defines:

attributes

`__verbose` - enable/disable verbose option.
`__file_path` - configuration file path.

methods

`__init__` - initial constructor.
`file_path` - property methods for set/get operations.
`is_not_none` - checking is file path None.
`read_configuration` - read configuration (Abstract method).
`__str__` - str dunder method for BaseReadConfig.

VERBOSE = 'ats_utilities::config_io::base_read'

file_path

Property method for getting file path.

Returns configuration file path | None.

Return type <str> | <NoneType>

Exception None

is_not_none()

Checking is file path None.

Returns boolean status, True (file path ok) | False.

Return type <bool>

Exceptions None

read_configuration = <ats_utilities.abstract.AbstractMethod object>

ats_utilities.config_io.base_write module

Module base_write.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class BaseWriteConfig with attribute(s) and method(s). Created API for write operation for configuration files.

class ats_utilities.config_io.base_write.**BaseWriteConfig** (*verbose=False*)
Bases: object

Defined class BaseWriteConfig with attribute(s) and method(s). Created API for write operation for configuration files. It defines:

attributes

`__verbose` - enable/disable verbose option.
`__file_path` - configuration file path.

methods

`__init__` - initial constructor.
`file_path` - property methods for set/get operations.
`is_not_none` - checking is file path None.
`write_configuration` - write configuration (Abstract method).
`__str__` - str dunder method for BaseWriteConfig.

VERBOSE = 'ats_utilities::config_io::base_write'

file_path

Property method for getting file path.

Returns configuration file path | None.

Return type <str> | <NoneType>

Exceptions None

is_not_none()

Checking is file path None.

Returns boolean status, True (file path ok) | False.

Return type <bool>

Exceptions None

write_configuration = <ats_utilities.abstract.AbstractMethod object>

Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ConfigFile with attribute(s) and method(s). Created API for information/configuration context manager.

class `ats_utilities.config_io.ConfigFile` (*file_path, file_mode, file_format, verbose=False*)
Bases: `ats_utilities.config_io.base_check.FileChecking`

Defined class ConfigFile with attribute(s) and method(s). Created API for information/configuration context manager. It defines:

attributes

`__verbose` - enable/disable verbose option.
`__file_path` - configuration file name.
`__file_mode` - file mode.
`__file_format` - file format.
`__file` - file object.

methods

`__init__` - initial constructor.
`__enter__` - open configuration file in mode.
`__exit__` - close configuration file.
`__str__` - str dunder method for ConfigFile.

`VERBOSE = 'ats_utilities::config_io'`

1.1.1.5 ats_utilities.console_io package

Submodules

ats_utilities.console_io.error module

Module error.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ATSError with attribute(s) and method(s). Created error message container for console log mechanism.

class `ats_utilities.console_io.error.ATSError`
Bases: `object`

Defined class ATSError with attribute(s) and method(s). Created error message container for console log mechanism. It defines:

attributes

`__message` - error message container.

methods

`__init__` - initial constructor.

`message` - property methods for set/get operations.

`is_not_none` - checking is message None or not.

`__str__` - str dunder method for ATSError.

`is_not_none()`

Checking is message None or not.

Returns boolean status, True | False.

Return type <bool>

Exceptions None

`message`

Property method for getting message.

Returns formatted errors message.

Return type <str>

Exceptions None

`ats_utilities.console_io.error.error_message(error_path, *message)`
Show error message.

Parameters

- **error_path** (<str>) – error prefix message.
- **message** (<tuple>) – message parts.

Exceptions ATSTypeError | ATSBadCallError

ats_utilities.console_io.success module

Module success.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ATSSuccess with attribute(s) and method(s). Created success message container for console log mechanism.

class `ats_utilities.console_io.success.ATSSuccess`

Bases: object

Defined class ATSSuccess with attribute(s) and method(s). Created success message container for console log mechanism. It defines:

attributes

__message - success message container.

methods

__init__ - initial constructor.

message - property methods for set/get operations.

is_not_none - checking is message None or not.

__str__ - str dunder method for ATSSuccess.

is_not_none ()

Checking is message None or not.

Returns boolean status, True | False.

Return type <bool>

Exceptions None

message

Property method for getting message.

Returns formatted success message.

Return type <str>

Exceptions None

ats_utilities.console_io.success.**success_message** (*success_path*, **message*)

Show success message.

Parameters

- **success_path** (<str>) – success prefix message.
- **message** (<tuple>) – message parts.

Exceptions ATSTypeError | ATSBadCallError

ats_utilities.console_io.verbose module

Module verbose.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ATSVerbose with attribute(s) and method(s). Created verbose message container for console log mechanism.

class ats_utilities.console_io.verbose.**ATSVerbose**

Bases: object

Defined class ATSVerbose with attribute(s) and method(s). Created verbose message container for console log mechanism. It defines:

attributes

__message - verbose message container.

methods

`__init__` - initial constructor.
`message` - property methods for set/get operations.
`is_not_none` - checking is message None or not.
`__str__` - str dunder method for ATSVerbose.

`is_not_none()`

Checking is message None or not.

Returns boolean status, True | False.

Return type <bool>

Exceptions None

`message`

Property method for getting message.

Returns formatted verbose message.

Return type <str>

`ats_utilities.console_io.verbose.verbose_message(verbose_path, verbose=False, *message)`

Show verbose message.

Parameters

- **verbose_path** (<str>) – verbose prefix message.
- **verbose** (<bool>) – enable/disable verbose option.
- **message** (<tuple>) – message parts.

Exceptions ATSTypeError | ATSBadCallError

ats_utilities.console_io.warning module

Module warning.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ATSWarning with attribute(s) and method(s). Created warning message container for console log mechanism.

class `ats_utilities.console_io.warning.ATSWarning`

Bases: object

Defined class ATSWarning with attribute(s) and method(s). Created warning message container for console log mechanism. It defines:

attributes

`__message` - warning message container.

methods

`__init__` - initial constructor.
`message` - property methods for set/get operations.
`is_not_none` - checking is message None or not.
`__str__` - str dunder method for ATSWarning.

is_not_none ()

Checking is message None or not.

Returns boolean status, True | False.

Return type <bool>

Exceptions None

message

Property method for getting message.

Returns formatted warning message.

Return type <str>

Exceptions None

`ats_utilities.console_io.warning.warning_message` (*warning_path*, **message*)

Show warning message.

Parameters

- **warning_path** (<str>) – warning prefix message.
- **message** (<tuple>) – message parts.

Exceptions ATSTypeError | ATSBadCallError

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined package `ats_utilities.console_io`.

1.1.1.6 `ats_utilities.cooperative` package

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General

Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class CooperativeMeta with attribute(s) and method(s). Created cooperative mechanism for multi-inheritance.

class `ats_utilities.cooperative.CooperativeMeta` (*name, bases, members*)

Bases: `type`

Defined class CooperativeMeta with attribute(s) and method(s). Created cooperative mechanism for multi-inheritance. It defines:

attributes

`combined metas` - combined metaclasses in form of dictionary.

methods

`__new__` - new constructor.

`__init__` - initial constructor.

`__str__` - str dunder method for CooperativeMeta.

1.1.1.7 `ats_utilities.exceptions` package

Submodules

`ats_utilities.exceptions.ats_attribute_error` module

Module `ats_attribute_error.py`

Copyright Copyright (C) 2017 Vladimir Roncevic elektron.ronca@gmail.com `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class `ATSAttributeError` with attribute(s) and method(s). Created exception for mechanism checking object attribute.

exception `ats_utilities.exceptions.ats_attribute_error.ATSAttributeError`

Bases: `AttributeError`

Defined class `ATSAttributeError` with attribute(s) and method(s). Created exception for mechanism checking object attribute. It defines:

attributes

None

methods

None

`ats_utilities.exceptions.ats_bad_call_error` module

Module `ats_bad_call_error.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ATSBadCallError with attribute(s) and method(s). Created exception for checking method/function argument(s) mechanism.

exception `ats_utilities.exceptions.ats_bad_call_error.ATSBadCallError`

Bases: Exception

Defined class ATSBadCallError with attribute(s) and method(s). Created exception for checking method/function argument(s) mechanism. It defines:

attributes

None

methods

None

ats_utilities.exceptions.ats_file_error module

Module `ats_file_error.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ATSFileError with attribute(s) and method(s). Created exception for checking file mechanism.

exception `ats_utilities.exceptions.ats_file_error.ATSFileError`

Bases: OSError

Defined class ATSFileError with attribute(s) and method(s). Created exception for checking file mechanism. It defines:

attributes

None

methods

None

ats_utilities.exceptions.ats_key_error module

Module `ats_key_error.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities

is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `ATSKeyError` with attribute(s) and method(s). Created exception for checking key mechanism.

exception `ats_utilities.exceptions.ats_key_error.ATSKeyError`

Bases: `KeyError`

Defined class `ATSKeyError` with attribute(s) and method(s). Created exception for checking key mechanism. It defines:

attributes

None

methods

None

ats_utilities.exceptions.ats_lookup_error module

Module `ats_lookup_error.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `ATSLookupError` with attribute(s) and method(s). Created exception for Lookup errors mechanism.

exception `ats_utilities.exceptions.ats_lookup_error.ATSLookupError`

Bases: `LookupError`

Defined class `ATSLookupError` with attribute(s) and method(s). Created exception for Lookup errors mechanism. It defines:

attributes

None

methods

None

ats_utilities.exceptions.ats_parameter_error module

Module `ats_parameter_error.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General

Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class `ATSPParameterError` with attribute(s) and method(s). Created exception mechanism (raise error for not expected format).

exception `ats_utilities.exceptions.ats_parameter_error.ATSPParameterError`

Bases: `Exception`

Defined class `ATSPParameterError` with attribute(s) and method(s). Created exception mechanism (raise error for not expected format). It defines:

attributes

None

methods

None

`ats_utilities.exceptions.ats_type_error` module

Module `ats_type_error.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class `ATSTypeError` with attribute(s) and method(s). Created exception for check type mechanism.

exception `ats_utilities.exceptions.ats_type_error.ATSTypeError`

Bases: `TypeError`

Defined class `ATSTypeError` with attribute(s) and method(s). Created exception for check type mechanism. It defines:

attributes

None

methods

None

`ats_utilities.exceptions.ats_value_error` module

Module `ats_value_error.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class `ATSError` with attribute(s) and method(s). Created exception for check value mechanism.

exception `ats_utilities.exceptions.ats_value_error.ATSError`

Bases: `ValueError`

Defined class `ATSError` with attribute(s) and method(s). Created exception for check value mechanism. It defines:

attributes

None

methods

None

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `ATSError` with attribute(s) and method(s). Created general exception (raise exception for wrong state/behavior).

exception `ats_utilities.exceptions.ATSError`

Bases: `Exception`

Defined class `ATSError` with attribute(s) and method(s). Created general exception (raise exception for wrong state/behavior). It defines:

attributes

None

methods

None

1.1.1.8 `ats_utilities.final` package

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `ATSFinal` with attribute(s) and method(s). Created API for App/Tool/Script utilities, set final class.

class `ats_utilities.final.ATSFinal` (*name, bases, namespace*)

Bases: `type`

Defined class ATSFinal with attribute(s) and method(s). Created API for App/Tool/Script utilities, set final class. It defines:

attributes

`registry` - set of classes.

methods

`__init__` - initial constructor.

`__str__` - str dunder method for ATSFinal.

1.1.1.9 `ats_utilities.info` package

Submodules

`ats_utilities.info.ats_build_date` module

Module `ats_build_date.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ATSBUILDDate with attribute(s) and method(s). Created API for ATS build date in one property object.

class `ats_utilities.info.ats_build_date.ATSBUILDDate` (*verbose=False*)

Bases: `object`

Defined class ATSBUILDDate with attribute(s) and method(s). Created API for ATS build date in one property object. It defines:

attributes

`__verbose` - enable/disable verbose option.

`__build_date` - ATS build date.

methods

`__init__` - initial constructor.

`build_date` - property methods for set/get operations.

`is_not_none` - checking is ATS build date None.

`__str__` - str dunder method for ATSBUILDDate.

VERBOSE = `'ats_utilities::info::ats_build_date'`

build_date

Property method for getting ATS build date.

Returns ATS build date | None.

Return type <str> | <NoneType>

Exceptions None

is_not_none ()

Checking is ATS build date None.

Returns boolean status, True | False.

Return type <bool>

Exceptions None

ats_utilities.info.ats_info_ok module

Module ats_info_ok.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ATInfoOk with attribute(s) and method(s). Created API for ATS info status in one property object.

class ats_utilities.info.ats_info_ok.**ATInfoOk** (*verbose=False*)

Bases: object

Defined class ATInfoOk with attribute(s) and method(s). Created API for ATS info status in one property object. It defines:

attributes

__verbose - enable/disable verbose option.

__ats_info_ok - ATS information status.

methods

__init__ - initial constructor.

ats_info_ok - property methods for set/get operations.

__str__ - str dunder method for ATInfoOk.

VERBOSE = 'ats_utilities::info::ats_info_ok'

ats_info_ok

Property method for getting ATS information status.

Returns boolean status, ATS information status.

Return type <bool>

Exceptions None

ats_utilities.info.ats_licence module

Module ats_licence.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the

Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class `ATSLicence` with attribute(s) and method(s). Created API for ATS licence in one property object.

class `ats_utilities.info.ats_licence.ATSLicence` (*verbose=False*)

Bases: `object`

Defined class `ATSLicence` with attribute(s) and method(s). Created API for ATS licence in one property object. It defines:

attributes

`__verbose` - enable/disable verbose option.

`__licence` - ATS licence.

methods

`__init__` - initial constructor.

`name` - property methods for set/get operations.

`is_not_none` - checking is ATS licence None.

`__str__` - str dunder method for `ATSLicence`.

VERBOSE = `'ats_utilities::info::ats_licence'`

is_not_none ()

Checking is ATS licence None.

Returns boolean status, True | False.

Return type `<bool>`

Exceptions None

licence

Property method for getting ATS licence.

Returns ATS licence | None.

Return type `<str>` | `<NoneType>`

Exceptions None

ats_utilities.info.ats_name module

Module `ats_name.py`

Copyright Copyright (C) 2017 Vladimir Roncevic elektron.ronca@gmail.com `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class `ATSName` with attribute(s) and method(s). Created API for ATS name in one property object.

class `ats_utilities.info.ats_name.ATSName` (*verbose=False*)

Bases: `object`

Defined class `ATSName` with attribute(s) and method(s). Created API for ATS name in one property object. It defines:

attributes

`__verbose` - enable/disable verbose option.

`__name` - ATS name.

methods

`__init__` - initial constructor.

`name` - property methods for set/get operations.

`is_not_none` - checking is ATS name None.

`__str__` - str dunder method for `ATSName`.

VERBOSE = `'ats_utilities::info::ats_name'`

is_not_none ()

Checking is ATS name None.

Returns boolean status, True | False.

Return type `<bool>`

Exceptions None

name

Property method for getting ATS name.

Returns ATS name | None.

Return type `<str>` | `<NoneType>`

Exceptions None

ats_utilities.info.ats_version module

Module `ats_version.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `ATSVersion` with attribute(s) and method(s). Created API for ATS version in one property object.

class `ats_utilities.info.ats_version.ATSVersion` (*verbose=False*)

Bases: `object`

Defined class `ATSVersion` with attribute(s) and method(s). Created API for ATS version in one property object. It defines:

attributes

`__verbose` - enable/disable verbose option.

`__version` - ATS version.

methods

`__init__` - initial constructor.
`version` - property methods for set/get operations.
`is_not_none` - checking is ATS version None.
`__str__` - str dunder method for ATSVersion.

VERBOSE = 'ats_utilities::info::ats_version'

is_not_none ()

Checking is ATS version None.

Returns boolean status, True | False.

Return type <bool>

Exceptions None

version

Property method for getting ATS version.

Returns ATS version | None.

Return type <str> | <NoneType>

Exceptions None

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `ATSInfo` with attribute(s) and method(s). Created API for ATS informations in one container object.

class `ats_utilities.info.ATSInfo` (*info, verbose=False*)

Bases: `ats_utilities.info.ats_name.ATSName`, `ats_utilities.info.ats_version.ATSVersion`, `ats_utilities.info.ats_licence.ATSLicence`, `ats_utilities.info.ats_build_date.ATSBuildDate`, `ats_utilities.info.ats_info_ok.ATSInfoOk`

Defined class `ATSInfo` with attribute(s) and method(s). Created API for ATS informations in one container object. It defines:

attributes

`__metaclass__` - setting cooperative metaclasses.
`ATS_NAME` - ATS name key.
`ATS_VERSION` - ATS version key.
`ATS_LICENCE` - ATS licence key.
`ATS_BUILD_DATE` - ATS build date key.
`ATS_BASE_INFO` - ATS base information dict.
`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.
`show_base_info` - show ATS informations.
`is_correct` - check information structure.
`__str__` - str dunder method for ATSInfo.

```
ATS_BASE_INFO = {1: 'ats_name', 2: 'ats_version', 3: 'ats_licence', 4: 'ats_build_
```

```
ATS_BUILD_DATE = 'ats_build_date'
```

```
ATS_LICENCE = 'ats_licence'
```

```
ATS_NAME = 'ats_name'
```

```
ATS_VERSION = 'ats_version'
```

```
VERBOSE = 'ats_utilities::info'
```

`is_correct` (*informations*, *verbose=False*)
Check information structure.

Parameters

- **informations** (*<dict>*) – ATS base informations.
- **verbose** (*<bool>*) – enable/disable verbose option.

Exceptions ATSTypeError | ATSBadCallError

`show_base_info` ()
Show ATS informations.

Exceptions None

1.1.1.10 ats_utilities.logging package

Submodules

ats_utilities.logging.ats_logger_file module

Module `ats_logger_file.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `ATSLoggerFile` with attribute(s) and method(s). Created API for ATS logger file path in one property object.

class `ats_utilities.logging.ats_logger_file.ATSLoggerFile` (*verbose=False*)
Bases: `object`

Defined class `ATSLoggerFile` with attribute(s) and method(s). Created API for ATS logger file path in one property object. It defines:

attributes

`__verbose` - enable/disable verbose option.
`__logger_file` - log file path.

methods

`__init__` - initial constructor.
`logger_file` - property methods for set/get operations.
`__str__` - str dunder method for `ATSLoggerFile`.

VERBOSE = `'ats_utilities::logging::ats_logger_file'`

`logger_file`

Property method for getting log file path.

Returns log file path | None.

Return type <str> | <NoneType>

Exceptions None

ats_utilities.logging.ats_logger_name module

Module `ats_logger_name.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `ATSLoggerName` with attribute(s) and method(s). Created API for ATS logger name in one property object.

class `ats_utilities.logging.ats_logger_name.ATSLoggerName` (*verbose=False*)
Bases: `object`

Defined class `ATSLoggerName` with attribute(s) and method(s). Created API for ATS logger name in one property object. It defines:

attributes

`__verbose` - enable/disable verbose option.
`__logger_name` - logger name.

methods

`__init__` - initial constructor.
`logger_name` - property methods for set/get operations.
`__str__` - str dunder method for `ATSLoggerName`.

VERBOSE = `'ats_utilities::logging::ats_logger_name'`

`logger_name`

Property method for getting logger name.

Returns logger name | None.

Return type <str> | <NoneType>

Exceptions None

ats_utilities.logging.ats_logger_status module

Module `ats_logger_status.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `ATSLoggerStatus` with attribute(s) and method(s). Created API for ATS logger status in one property object.

class `ats_utilities.logging.ats_logger_status.ATSLoggerStatus` (*verbose=False*)
Bases: `object`

Defined class `ATSLoggerStatus` with attribute(s) and method(s). Created API for ATS logger status in one property object. It defines:

attributes

`__verbose` - enable/disable verbose option.
`__log_status` - logger status (enabled/disabled).

methods

`__init__` - initial constructor.
`logger_status` - property methods for set/get operations.
`__str__` - str dunder method for `ATSLoggerStatus`.

VERBOSE = `'ats_utilities::logging::ats_logger_status'`

`logger_status`

Property method for getting logger status.

Returns True (logger operative) | False.

Return type <bool>

Exceptions None

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> `ats_utilities` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. `ats_utilities` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `ATSLogger` with attribute(s) and method(s). Created API for ATS logging mechanism.

class `ats_utilities.logging.ATSLogger` (*ats_name, ats_log_file, verbose=False*)

Bases: `ats_utilities.logging.ats_logger_name.ATSLoggerName`, `ats_utilities.logging.ats_logger_file.ATSLoggerFile`, `ats_utilities.logging.ats_logger_status.ATSLoggerStatus`

Defined class `ATSLogger` with attribute(s) and method(s). Created API for ATS logging mechanism. It defines:

attributes

`LOG_MSG_FORMAT` - log message format.
`LOG_DATE_FORMAT` - log date format.
`ATS_DEBUG` - debug log level.
`ATS_WARNING` - warning log level.
`ATS_CRITICAL` - critical log level.
`ATS_ERROR` - error log level.
`ATS_INFO` - info log level.
`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.
`write_log` - write message to log file.
`__str__` - str dunder method for `ATSLogger`.

`ATS_CRITICAL = 50`

`ATS_DEBUG = 10`

`ATS_ERROR = 40`

`ATS_INFO = 20`

`ATS_WARNING = 30`

`LOG_DATE_FORMAT = '%m/%d/%Y %I:%M:%S %p'`

`LOG_MSG_FORMAT = '%(asctime)s - %(levelname)s - %(message)s'`

`VERBOSE = 'ats_utilities::logging'`

`write_log` (*message, ctrl, verbose=False*)

Write message to log file.

Parameters

- **message** (*<str>*) – log message.
- **ctrl** (*<int>*) – control flag (debug, warning, critical, errors, info).
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns boolean status, True (success) | False.

Return type `<bool>`

Exceptions `ATSTypeError` | `ATSBadCallError`

1.1.1.11 `ats_utilities.option` package

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ATSOptionParser with attribute(s) and method(s). Created option parser and argument processor.

class `ats_utilities.option.ATSOptionParser` (*version, epilog, description, verbose=False*)

Bases: `object`

Defined class ATSOptionParser with attribute(s) and method(s). Created option parser and argument processor.

It defines:

attributes

`__verbose` - enable/disable verbose option.

`__opt_parser` - options parser.

methods

`__init__` - initial constructor.

`add_operation` - add option to ATS.

`parse_args` - process arguments from start.

`__str__` - str dunder method for ATSOptionParser.

VERBOSE = `'ats_utilities::option'`

add_operation (**args, **kwargs*)

Add option to ATS.

Parameters

- **args** (*<list>*) – list of arguments (objects).
- **kwargs** (*<dict>*) – arguments in shape of dictionary.

Exceptions None

parse_args (*arguments, verbose=False*)

Process arguments from start.

Parameters

- **arguments** (*<list>*) – arguments.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns namespace object.

Return type `<argparse.Namespace>`

Exceptions None

1.1.1.12 `ats_utilities.register` package

Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ATSRegister with attribute(s) and method(s). Created API for App/Tool/Script utilities, auto-register class.

class `ats_utilities.register.ATSRegister` (*name, bases, namespace*)
Bases: `type`

Defined class ATSRegister with attribute(s) and method(s). Created API for App/Tool/Script utilities, auto-register class. It defines:

attributes

registry - set of classes.

methods

`__init__` - initial constructor.

`__iter__` - getting iterator of set.

`__str__` - str dunder method for ATSRegister.

1.1.1.13 ats_utilities.singleton package

Submodules

ats_utilities.singleton.base module

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class Singleton with attribute(s) and method(s). Created API for auto-register singleton object.

class `ats_utilities.singleton.base.Singleton`
Bases: `object`

Defined class Singleton with attribute(s) and method(s). Created API for auto-register singleton object. It defines:

attributes

`__INSTANCES` - class dictionary for collecting instances.

methods

`__new__` - set class instance.

`__str__` - str dunder method for Singleton.

```
__Singleton__INSTANCE = None
```

ats_utilities.singleton.functional module

Module functional.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined decorator FunctionalSingleton without attribute(s). Created API for auto-register singleton object.

```
ats_utilities.singleton.functional.FunctionalSingleton (class_)
    Checking instance and setup instance.
```

Parameters `class (<Python Object>)` – singleton class.

Returns python object instance.

Return type <Python Object>

Exceptions None

ats_utilities.singleton.meta module

Module meta.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class SingletonMeta with attribute(s) and method(s). Created API for auto-register singleton object.

```
class ats_utilities.singleton.meta.SingletonMeta
    Bases: type
```

Defined class SingletonMeta with attribute(s) and method(s). Created API for auto-register singleton object. It defines:

attributes

`__INSTANCES` - class dictionary for collecting instances.

methods

`__call__` - check instance and setup instance.

`__str__` - str dunder method for SingletonMeta.

```
__SingletonMeta__INSTANCES = {}
```

ats_utilities.singleton.meta_thread module

Module meta_thread.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ThreadSafeSingletonMeta with attribute(s) and method(s). Created API for auto-register singleton object.

class ats_utilities.singleton.meta_thread.ThreadSafeSingletonMeta

Bases: type

Defined class ThreadSafeSingletonMeta with attribute(s) and method(s). Created API for auto-register singleton object. It defines:

attributes

__INSTANCES - class dictionary for collecting instances.

__LOCK - thread lock object.

methods

__call__ - check instance and setup instance.

__str__ - str dunder method for ThreadSafeSingletonMeta.

```
_ThreadSafeSingletonMeta__INSTANCES = {}
```

```
_ThreadSafeSingletonMeta__LOCK = <unlocked _thread.lock object>
```

Module contents

Module __init__.py

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined package ats_utilities.singleton.

1.1.1.14 ats_utilities.splash package

Submodules

ats_utilities.splash.ext_infrastructure module

Module ext_infrastructure.py

Copyright Copyright (C) 2021 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ExtInfrastructure with attribute(s) and method(s). Created API for processing hyperlinks for splash.

class `ats_utilities.splash.ext_infrastructure.ExtInfrastructure` (*property*, *verbose=False*)

Bases: `object`

Defined class ExtInfrastructure with attribute(s) and method(s). Created API for processing hyperlinks for splash. It defines:

attributes

`__verbose` - enable/disable verbose option.
`__property` - splash property in dict format.

methods

`__init__` - initial constructor.
`get_info_text` - pre-processed info text.
`get_issue_text` - pre-processed issue text.
`get_author_text` - pre-processed author text.
`__str__` - dunder method for ExtInfrastructure.

VERBOSE = `'ats_utilities::splash::ext_infrastructure'`

get_author_text ()

Pre-processed author text for splash.

Returns hyperlink with author info.

Return type <str>

Exceptions None

get_info_text ()

Pre-processed info text for splash.

Returns hyperlink with info text.

Return type <str>

Exceptions None

get_issue_text ()

Pre-processed issue text for splash.

Returns hyperlink with issue info.

Return type <str>

Exceptions None

ats_utilities.splash.github_infrastructure module

Module `github_infrastructure.py`

Copyright Copyright (C) 2021 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class GitHubInfrastructure with attribute(s) and method(s). Created API for processing hyperlinks for splash.

class ats_utilities.splash.github_infrastructure.**GitHubInfrastructure** (*property*,
ver-
bose=False)

Bases: object

Defined class GitHubInfrastructure with attribute(s) and method(s). Created API for processing hyperlinks for splash. It defines:

attributes

__verbose - enable/disable verbose option.

__property - splash property in dict format.

methods

__init__ - initial constructor.

get_info_text - pre-processed info text.

get_issue_text - pre-processed issue text.

get_author_text - pre-processed author text.

__str__ - dunder method for GitHubInfrastructure.

VERBOSE = 'ats_utilities::splash::github_infrastructure'

get_author_text ()

Pre-processed author text for splash.

Returns hyperlink with author info.

Return type <str>

Exceptions None

get_info_text ()

Pre-processed info text for splash.

Returns hyperlink with info text.

Return type <str>

Exceptions None

get_issue_text ()

Pre-processed issue text for splash.

Returns hyperlink with issue info.

Return type <str>

Exceptions None

ats_utilities.splash.progress_bar module

Module progress_bar.py

Copyright Copyright (C) 2021 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class ProgressBar with attribute(s) and method(s). Load a progressbar as part of splashscreen.

class ats_utilities.splash.progress_bar.ProgressBar (end, start=0)
Bases: object

Defined class ProgressBar with attribute(s) and method(s). Load a progressbar as part of splashscreen. It defines:

attributes

DEFAULT_BAR_LENGTH - length of progressbar.
DEFAULT_CHAR_ON - loaded progress element.
DEFAULT_CHAR_OFF - unloaded progress element.
start - start of level.
end - end of level.
_bar_length - progress length.
_level - progress level.
_plotted - plotted progress.
_level_chars - level progress chars.

methods

__init__ - initial constructor.
set_level - set level of progress.
plot_progress - plot progress.
set_and_plot - set and plot progress.
__del__ - dunder method for ProgressBar.

DEFAULT_BAR_LENGTH = 60

DEFAULT_CHAR_OFF = ' '

DEFAULT_CHAR_ON = ''

plot_progress (columns)
Plot progress.

Parameters **columns** (<int>) – columns for open console session.

Exceptions None

set_and_plot (level, columns)
Set and plot progress.

Parameters

- **level** (<int>) – level of progress.
- **columns** (<int>) – columns for open console session.

Exceptions None

set_level (*level*)

Set level.

Parameters **level** (<*int*>) – level of progress.

Exceptions None

ats_utilities.splash.splash_property module

Module splash_property.py

Copyright Copyright (C) 2021 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class SplashProperty with attribute(s) and method(s). Created API for checking splash property.

class ats_utilities.splash.splash_property.**SplashProperty** (*property*, *verbose=False*)

Bases: object

Defined class SplashProperty with attribute(s) and method(s). Created API for checking splash property. It defines:

attributes

`__EXPECTED_PROPERTY_KEYS` - expected property names.

`__verbose` - enable/disable verbose option.

`__property` - splash property in dict format.

methods

`__init__` - initial constructor.

`validation` - validation of splash property.

`__str__` - dunder method for SplashProperty.

`VERBOSE = 'ats_utilities::splash::splash_property'`

`__SplashProperty__EXPECTED_PROPERTY_KEYS = ['ats_organization', 'ats_repository', 'ats_n`

validate (*verbose=False*)

Validation of splash property.

Parameters **verbose** (<*bool*>) – enable/disable verbose option.

Returns Boolean status True - property ok else False.

Return type <bool>

Exceptions None

ats_utilities.splash.terminal_properties module

Module terminal_properties.py

Copyright Copyright (C) 2021 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class TerminalProperties with attribute(s) and method(s). Defined API for getting terminal properties.

class `ats_utilities.splash.terminal_properties.TerminalProperties` (*verbose=False*)
Bases: `object`

Defined class TerminalProperties with attribute(s) and method(s). Defined API for getting terminal properties. It defines:

attributes

`__verbose` - enable/disable verbose option.
`__window_size` - terminal window size.

methods

`__init__` - initial constructor.
`__ioctl_get_window_size` - size for descriptor.
`__ioctl_for_all_descriptors` - size for all descriptors.
`size` - getting size of terminal window.
`__str__` - dunder method for TerminalProperties.

VERBOSE = `'ats_utilities::splash::terminal_properties'`

`__TerminalProperties__ioctl_for_all_descriptors` (*verbose=False*)
Check window size for all descriptors.

Parameters `verbose` (<bool>) – enable/disable verbose option.

Returns status true window size checked else false.

Return type <bool>

Exceptions None

`__TerminalProperties__ioctl_get_window_size` (*file_descriptor, verbose=False*)
Check window size for descriptor.

Parameters

- `file_descriptor` (<int>) – file descriptor.
- `verbose` (<bool>) – enable/disable verbose option.

Returns window size of terminal.

Return type <tuple>

Exceptions `ATSTypeError` | `ATSBadCallError`

size (*verbose=False*)
Center console line.

Parameters `verbose` (<bool>) – enable/disable verbose option.

Returns window size | None.

Return type <tuple> | <NoneType>

Exceptions None

Module contents

Module `__init__.py`

Copyright Copyright (C) 2021 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class `Splash` with attribute(s) and method(s). Load a splash screen info and add hyperlinks.

class `ats_utilities.splash.Splash` (*ats_splash_property*, *verbose=False*)
Bases: `ats_utilities.splash.splash_property.SplashProperty`

Defined class `Splash` with attribute(s) and method(s). Load a splash screen info and add hyperlinks. It defines:

attributes

`__verbose` - enable/disable verbose option.

methods

`__init__` - initial constructor.

`center` - center console line.

`__str__` - dunder method for `Splash`.

VERBOSE = `'ats_utilities::splash'`

center (*columns*, *additional_shifter*, *text*, *verbose=False*)
Center console line.

Parameters

- **columns** (<int>) – columns for open console session.
- **additional_shifter** (<int>) – additional shifters.
- **text** (<str>) – text for console session.
- **verbose** (<bool>) – enable/disable verbose option.

Exceptions `ATSTypeError` | `ATSBadCallError`

1.1.2 Module contents

Module `__init__.py`

Copyright Copyright (C) 2017 Vladimir Roncevic <elektron.ronca@gmail.com> ats_utilities is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. ats_utilities is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Info Defined class VerboseRoot with attribute(s) and method(s). Created API for setup verbose class path from metaclass.

class `ats_utilities.VerboseRoot`

Bases: `type`

Defined class VerboseRoot with attribute(s) and method(s). Created API for setup verbose class path from metaclass. It defines:

attributes

`ROOT_PACKAGE_NAME` - root package name of framework.

methods

`__new__` - new constructor.

`__str__` - str dunder method for VerboseRoot.

`ROOT_PACKAGE_NAME = 'ATS_UTILITIES'`

CHAPTER 2

Installation

Used next development environment



```
vroncevic@debtux
-----
OS: Debian GNU/Linux 11 (bullseye) x86_64
Host: 20F5S80B00 ThinkPad X260
Kernel: 5.10.0-8-amd64
Uptime: 34 mins
Packages: 3372 (dpkg)
Shell: bash 5.1.8
Resolution: 1366x768
DE: GNOME 3.38.4
WM: Mutter
WM Theme: Adwaita
Theme: Adwaita-dark [GTK2/3]
Icons: Adwaita [GTK2/3]
Terminal: terminator
CPU: Intel i5-6200U (4) @ 2.800GHz
GPU: Intel Skylake GT2 [HD Graphics 520]
Memory: 2907MiB / 7390MiB
```

The terminal window displays system information on the right and a large ASCII art logo on the left. The logo is a stylized 'G' shape composed of characters like '\$', 'P', 'Y', 'b', and 'g'. At the bottom of the terminal window, there is a color calibration bar with several colored squares.

Navigate to release [page](#) download and extract release archive .

To install **ats_utilities** run

```
tar xvzf ats_utilities-x.y.z.tar.gz
cd ats_utilities-x.y.z
# python2
wget https://bootstrap.pypa.io/pip/2.7/get-pip.py
python2 get-pip.py
python2 -m pip install --upgrade setuptools
python2 -m pip install --upgrade pip
python2 -m pip install --upgrade build
pip2 install -r requirements.txt
python2 -m build --no-isolation --wheel
pip2 install dist/ats_utilities-x.y.z-py2-none-any.whl
rm -f get-pip.py
# python3
wget https://bootstrap.pypa.io/get-pip.py
python3 get-pip.py
python3 -m pip install --upgrade setuptools
python3 -m pip install --upgrade pip
python3 -m pip install --upgrade build
pip3 install -r requirements.txt
python3 -m build --no-isolation --wheel
pip3 install dist/ats_utilities-x.y.z-py3-none-any.whl
rm -f get-pip.py
```

Or type the following

```
tar xvzf ats_utilities-x.y.z.tar.gz
cd ats_utilities-x.y.z/
# python2
wget https://bootstrap.pypa.io/pip/2.7/get-pip.py
python2 get-pip.py
python2 -m pip install --upgrade setuptools
python2 -m pip install --upgrade pip
python2 -m pip install --upgrade build
pip2 install -r requirements.txt
python2 setup.py install_lib
python2 setup.py install_egg_info
# python3
wget https://bootstrap.pypa.io/get-pip.py
python3 get-pip.py
python3 -m pip install --upgrade setuptools
python3 -m pip install --upgrade pip
python3 -m pip install --upgrade build
pip3 install -r requirements.txt
python3 setup.py install_lib
python3 setup.py install_egg_info
```

You can use Docker to create image/container, or You can use pip to install

```
# python2
pip2 install ats_utilities
# python3
pip3 install ats_utilities
```

CHAPTER 3

Dependencies

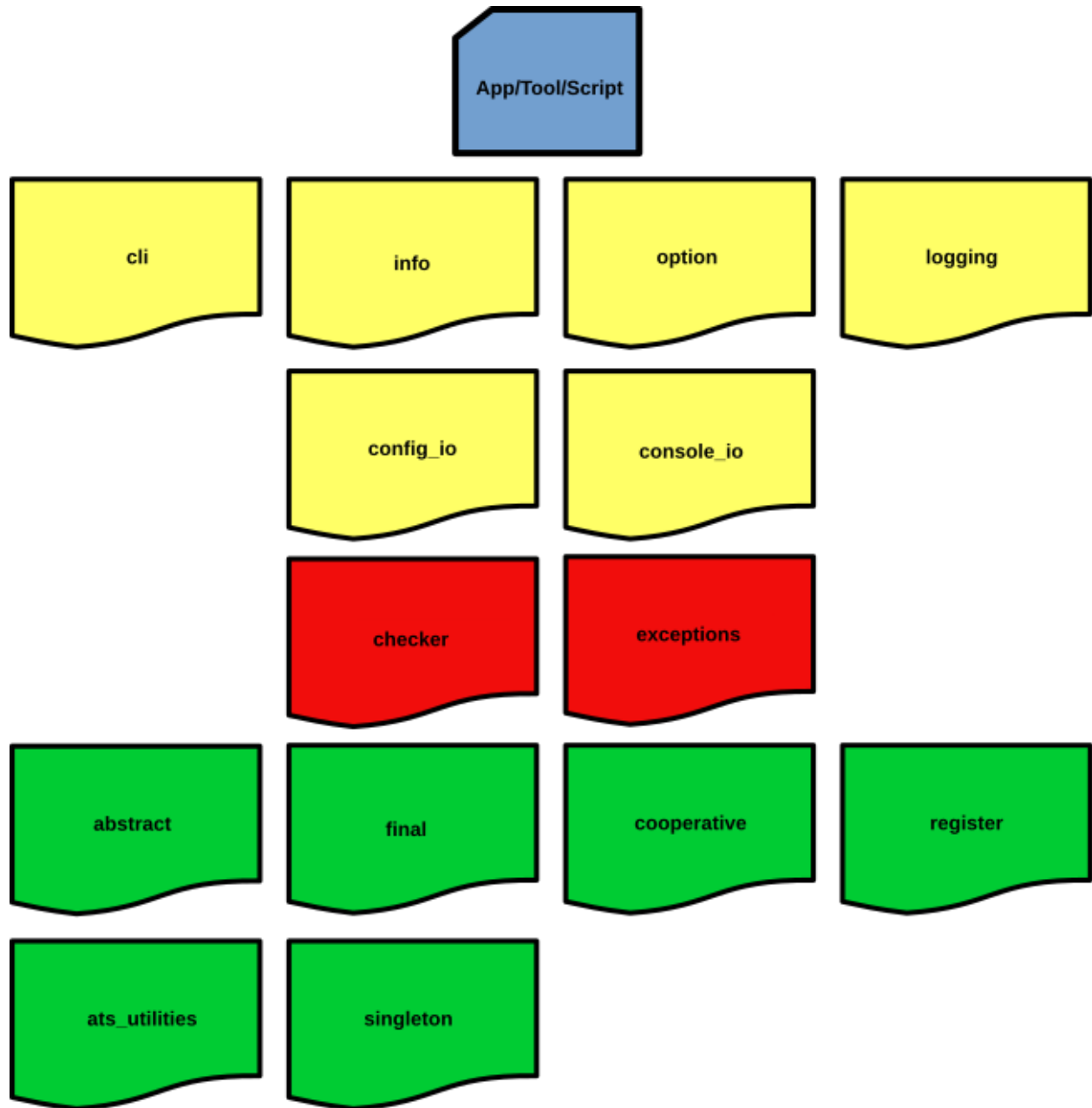
ats_utilities requires next modules and libraries

- six - Python 2 and 3 compatibility library
- yaml - YAML parser and emitter for Python
- bs4 - Screen-scraping library
- configparser - Configuration parser library
- colorama - Cross-platform colored terminal text

CHAPTER 4

Framework structure

ats_utilities is based on OOP.



Framework structure

```
ats_utilities/  
├── abstract/  
│   └── __init__.py  
├── checker/  
│   └── __init__.py  
├── cli/  
│   ├── cfg_cli.py  
│   ├── ini_cli.py  
│   ├── __init__.py  
│   ├── json_cli.py  
│   ├── xml_cli.py  
│   └── yaml_cli.py
```

(continues on next page)

(continued from previous page)

```
— config_io/
  — base_check.py
  — base_read.py
  — base_write.py
  — cfg/
    — cfg2object.py
    — __init__.py
    — object2cfg.py
  — ini/
    — ini2object.py
    — __init__.py
    — object2ini.py
  — __init__.py
  — json/
    — __init__.py
    — json2object.py
    — object2json.py
  — xml/
    — __init__.py
    — object2xml.py
    — xml2object.py
  — yaml/
    — __init__.py
    — object2yaml.py
    — yaml2object.py
— console_io/
  — error.py
  — __init__.py
  — success.py
  — verbose.py
  — warning.py
— cooperative/
  — __init__.py
— exceptions/
  — ats_attribute_error.py
  — ats_bad_call_error.py
  — ats_file_error.py
  — ats_key_error.py
  — ats_lookup_error.py
  — ats_parameter_error.py
  — ats_type_error.py
  — ats_value_error.py
  — __init__.py
— final/
  — __init__.py
— info/
  — ats_build_date.py
  — ats_info_ok.py
  — ats_licence.py
  — ats_name.py
  — ats_version.py
  — __init__.py
— __init__.py
— logging/
  — ats_logger_file.py
  — ats_logger_name.py
  — ats_logger_status.py
```

(continues on next page)

(continued from previous page)



Copyright and licence

Copyright (C) 2017 by vrontcevic.github.io/ats_utilities

ats_utilities is free software; you can redistribute it and/or modify it under the same terms as Python itself, either Python version 2.x/3.x or, at your option, any later version of Python 3 you may have available.

Lets help and support PSF.



CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

a

ats_utilities, 55
ats_utilities.abstract, 3
ats_utilities.checker, 4
ats_utilities.cli, 10
ats_utilities.cli.cfg_cli, 6
ats_utilities.cli.ini_cli, 7
ats_utilities.cli.json_cli, 8
ats_utilities.cli.xml_cli, 8
ats_utilities.cli.yaml_cli, 9
ats_utilities.config_io, 26
ats_utilities.config_io.base_check, 23
ats_utilities.config_io.base_read, 25
ats_utilities.config_io.base_write, 26
ats_utilities.config_io.cfg, 12
ats_utilities.config_io.cfg.cfg2object, 11
ats_utilities.config_io.cfg.object2cfg, 12
ats_utilities.config_io.ini, 15
ats_utilities.config_io.ini.ini2object, 13
ats_utilities.config_io.ini.object2ini, 14
ats_utilities.config_io.json, 17
ats_utilities.config_io.json.json2object, 16
ats_utilities.config_io.json.object2json, 17
ats_utilities.config_io.xml, 20
ats_utilities.config_io.xml.object2xml, 18
ats_utilities.config_io.xml.xml2object, 19
ats_utilities.config_io.yaml, 22
ats_utilities.config_io.yaml.object2yaml, 21
ats_utilities.config_io.yaml.yaml2object, 22
ats_utilities.console_io, 31
ats_utilities.console_io.error, 27
ats_utilities.console_io.success, 28
ats_utilities.console_io.verbose, 29
ats_utilities.console_io.warning, 30
ats_utilities.cooperative, 31
ats_utilities.exceptions, 36
ats_utilities.exceptions.ats_attribute_error, 32
ats_utilities.exceptions.ats_bad_call_error, 32
ats_utilities.exceptions.ats_file_error, 33
ats_utilities.exceptions.ats_key_error, 33
ats_utilities.exceptions.ats_lookup_error, 34
ats_utilities.exceptions.ats_parameter_error, 34
ats_utilities.exceptions.ats_type_error, 35
ats_utilities.exceptions.ats_value_error, 35
ats_utilities.final, 36
ats_utilities.info, 41
ats_utilities.info.ats_build_date, 37
ats_utilities.info.ats_info_ok, 38
ats_utilities.info.ats_licence, 38
ats_utilities.info.ats_name, 39
ats_utilities.info.ats_version, 40
ats_utilities.logging, 44
ats_utilities.logging.ats_logger_file, 42
ats_utilities.logging.ats_logger_name, 43
ats_utilities.logging.ats_logger_status, 44
ats_utilities.option, 45
ats_utilities.register, 46
ats_utilities.singleton, 49

ats_utilities.singleton.base, [47](#)
ats_utilities.singleton.functional, [48](#)
ats_utilities.singleton.meta, [48](#)
ats_utilities.singleton.meta_thread, [49](#)
ats_utilities.splash, [55](#)
ats_utilities.splash.ext_infrastructure,
 [49](#)
ats_utilities.splash.github_infrastructure,
 [50](#)
ats_utilities.splash.progress_bar, [52](#)
ats_utilities.splash.splash_property,
 [53](#)
ats_utilities.splash.terminal_properties,
 [53](#)

Symbols

- `_Cfg2Object__FORMAT`
(*ats_utilities.config_io.cfg.cfg2object.Cfg2Object* attribute), 11
 - `_Cfg2Object__REGEX_MATCH_LINE`
(*ats_utilities.config_io.cfg.cfg2object.Cfg2Object* attribute), 11
 - `_Ini2Object__FORMAT`
(*ats_utilities.config_io.ini.ini2object.Ini2Object* attribute), 14
 - `_Json2Object__FORMAT`
(*ats_utilities.config_io.json.json2object.Json2Object* attribute), 16
 - `_Object2Cfg__FORMAT`
(*ats_utilities.config_io.cfg.object2cfg.Object2Cfg* attribute), 12
 - `_Object2Ini__FORMAT`
(*ats_utilities.config_io.ini.object2ini.Object2Ini* attribute), 15
 - `_Object2Json__FORMAT`
(*ats_utilities.config_io.json.object2json.Object2Json* attribute), 17
 - `_Object2Xml__FORMAT`
(*ats_utilities.config_io.xml.object2xml.Object2Xml* attribute), 19
 - `_Object2Yaml__FORMAT`
(*ats_utilities.config_io.yaml.object2yaml.Object2Yaml* attribute), 21
 - `_SingletonMeta__INSTANCES`
(*ats_utilities.singleton.meta.SingletonMeta* attribute), 48
 - `_Singleton__INSTANCE`
(*ats_utilities.singleton.base.Singleton* attribute), 47
 - `_SplashProperty__EXPECTED_PROPERTY_KEYS`
(*ats_utilities.splash.splash_property.SplashProperty* attribute), 53
 - `_TerminalProperties__ioctl_for_all_descriptors`
(*ats_utilities.splash.terminal_properties.TerminalProperties* method), 54
 - `_TerminalProperties__ioctl_get_window_size()`
(*ats_utilities.splash.terminal_properties.TerminalProperties* method), 54
 - `_ThreadSafeSingletonMeta__INSTANCES`
(*ats_utilities.singleton.meta_thread.ThreadSafeSingletonMeta* attribute), 49
 - `_ThreadSafeSingletonMeta__LOCK`
(*ats_utilities.singleton.meta_thread.ThreadSafeSingletonMeta* attribute), 49
 - `_Xml2Object__FORMAT`
(*ats_utilities.config_io.xml.xml2object.Xml2Object* attribute), 20
 - `_Yaml2Object__FORMAT`
(*ats_utilities.config_io.yaml.yaml2object.Yaml2Object* attribute), 22
- ## A
- `AbstractMethod` (class in *ats_utilities.abstract*), 3
 - `add_new_option()` (*ats_utilities.cli.cfg_cli.CfgCLI* method), 6
 - `add_new_option()` (*ats_utilities.cli.ini_cli.IniCLI* method), 7
 - `add_new_option()` (*ats_utilities.cli.json_cli.JsonCLI* method), 8
 - `add_new_option()` (*ats_utilities.cli.xml_cli.XmlCLI* method), 9
 - `add_new_option()` (*ats_utilities.cli.yaml_cli.YamlCLI* method), 10
 - `add_operation()` (*ats_utilities.option.ATSOptionParser* method), 46
 - `ATS_BASE_INFO` (*ats_utilities.info.ATSInfo* attribute), 42
 - `ATS_BUILD_DATE` (*ats_utilities.info.ATSInfo* attribute), 42
 - `ATS_CRITICAL` (*ats_utilities.logging.ATSLogger* attribute), 45
 - `ATS_DEBUG` (*ats_utilities.logging.ATSLogger* attribute), 45

ATS_ERROR (*ats_utilities.logging.ATSLogger attribute*), 45
 ATS_INFO (*ats_utilities.logging.ATSLogger attribute*), 45
 ats_info_ok (*ats_utilities.info.ats_info_ok.ATSInfoOk attribute*), 38
 ATS_LICENCE (*ats_utilities.info.ATSInfo attribute*), 42
 ATS_NAME (*ats_utilities.info.ATSInfo attribute*), 42
 ats_utilities (*module*), 55
 ats_utilities.abstract (*module*), 3
 ats_utilities.checker (*module*), 4
 ats_utilities.cli (*module*), 10
 ats_utilities.cli.cfg_cli (*module*), 6
 ats_utilities.cli.ini_cli (*module*), 7
 ats_utilities.cli.json_cli (*module*), 8
 ats_utilities.cli.xml_cli (*module*), 8
 ats_utilities.cli.yaml_cli (*module*), 9
 ats_utilities.config_io (*module*), 26
 ats_utilities.config_io.base_check (*module*), 23
 ats_utilities.config_io.base_read (*module*), 25
 ats_utilities.config_io.base_write (*module*), 26
 ats_utilities.config_io.cfg (*module*), 12
 ats_utilities.config_io.cfg.cfg2object (*module*), 11
 ats_utilities.config_io.cfg.object2cfg (*module*), 12
 ats_utilities.config_io.ini (*module*), 15
 ats_utilities.config_io.ini.ini2object (*module*), 13
 ats_utilities.config_io.ini.object2ini (*module*), 14
 ats_utilities.config_io.json (*module*), 17
 ats_utilities.config_io.json.json2object (*module*), 16
 ats_utilities.config_io.json.object2json (*module*), 17
 ats_utilities.config_io.xml (*module*), 20
 ats_utilities.config_io.xml.object2xml (*module*), 18
 ats_utilities.config_io.xml.xml2object (*module*), 19
 ats_utilities.config_io.yaml (*module*), 22
 ats_utilities.config_io.yaml.object2yaml (*module*), 21
 ats_utilities.config_io.yaml.yaml2object (*module*), 22
 ats_utilities.console_io (*module*), 31
 ats_utilities.console_io.error (*module*), 27
 ats_utilities.console_io.success (*module*), 28
 ats_utilities.console_io.verbose (*module*), 29
 ats_utilities.console_io.warning (*module*), 30
 ats_utilities.cooperative (*module*), 31
 ats_utilities.exceptions (*module*), 36
 ats_utilities.exceptions.ats_attribute_error (*module*), 32
 ats_utilities.exceptions.ats_bad_call_error (*module*), 32
 ats_utilities.exceptions.ats_file_error (*module*), 33
 ats_utilities.exceptions.ats_key_error (*module*), 33
 ats_utilities.exceptions.ats_lookup_error (*module*), 34
 ats_utilities.exceptions.ats_parameter_error (*module*), 34
 ats_utilities.exceptions.ats_type_error (*module*), 35
 ats_utilities.exceptions.ats_value_error (*module*), 35
 ats_utilities.final (*module*), 36
 ats_utilities.info (*module*), 41
 ats_utilities.info.ats_build_date (*module*), 37
 ats_utilities.info.ats_info_ok (*module*), 38
 ats_utilities.info.ats_licence (*module*), 38
 ats_utilities.info.ats_name (*module*), 39
 ats_utilities.info.ats_version (*module*), 40
 ats_utilities.logging (*module*), 44
 ats_utilities.logging.ats_logger_file (*module*), 42
 ats_utilities.logging.ats_logger_name (*module*), 43
 ats_utilities.logging.ats_logger_status (*module*), 44
 ats_utilities.option (*module*), 45
 ats_utilities.register (*module*), 46
 ats_utilities.singleton (*module*), 49
 ats_utilities.singleton.base (*module*), 47
 ats_utilities.singleton.functional (*module*), 48
 ats_utilities.singleton.meta (*module*), 48
 ats_utilities.singleton.meta_thread (*module*), 49
 ats_utilities.splash (*module*), 55
 ats_utilities.splash.ext_infrastructure (*module*), 49
 ats_utilities.splash.github_infrastructure (*module*), 50

ats_utilities.splash.progress_bar (module), 52

ats_utilities.splash.splash_property (module), 53

ats_utilities.splash.terminal_properties (module), 53

ATS_VERSION (ats_utilities.info.ATSInfo attribute), 42

ATS_WARNING (ats_utilities.logging.ATSLogger attribute), 45

ATSAttributeError, 32

ATSBadCallError, 33

ATSBuildDate (class in ats_utilities.info.ats_build_date), 37

ATSChecker (class in ats_utilities.checker), 4

ATSError, 36

ATSError (class in ats_utilities.console_io.error), 27

ATSFileError, 33

ATSFinal (class in ats_utilities.final), 36

ATSInfo (class in ats_utilities.info), 41

ATSInfoOk (class in ats_utilities.info.ats_info_ok), 38

ATSKeyError, 34

ATSLicence (class in ats_utilities.info.ats_licence), 39

ATSLogger (class in ats_utilities.logging), 44

ATSLoggerFile (class in ats_utilities.logging.ats_logger_file), 42

ATSLoggerName (class in ats_utilities.logging.ats_logger_name), 43

ATSLoggerStatus (class in ats_utilities.logging.ats_logger_status), 44

ATSLookupError, 34

ATSName (class in ats_utilities.info.ats_name), 39

ATSOptionParser (class in ats_utilities.option), 46

ATSParameterError, 35

ATSRegister (class in ats_utilities.register), 47

ATSSuccess (class in ats_utilities.console_io.success), 28

ATSTypeError, 35

ATSValueError, 36

ATSVerbose (class in ats_utilities.console_io.verbose), 29

ATSVersion (class in ats_utilities.info.ats_version), 40

ATSWarning (class in ats_utilities.console_io.warning), 30

B

BaseReadConfig (class in ats_utilities.config_io.base_read), 25

BaseWriteConfig (class in ats_utilities.config_io.base_write), 26

build_date (ats_utilities.info.ats_build_date.ATSBuildDate attribute), 37

C

center () (ats_utilities.splash.Splash method), 55

Cfg2Object (class in ats_utilities.config_io.cfg.cfg2object), 11

CfgBase (class in ats_utilities.config_io.cfg), 13

CfgCLI (class in ats_utilities.cli.cfg_cli), 6

check_format () (ats_utilities.config_io.base_check.FileChecking method), 24

check_mode () (ats_utilities.config_io.base_check.FileChecking method), 24

check_params () (ats_utilities.checker.ATSChecker method), 5

check_path () (ats_utilities.config_io.base_check.FileChecking method), 24

check_types () (ats_utilities.checker.ATSChecker method), 5

check_values () (ats_utilities.checker.ATSChecker method), 5

collect_params () (ats_utilities.checker.ATSChecker method), 5

ConfigFile (class in ats_utilities.config_io), 27

CooperativeMeta (class in ats_utilities.cooperative), 32

D

DEFAULT_BAR_LENGTH (ats_utilities.splash.progress_bar.ProgressBar attribute), 52

DEFAULT_CHAR_OFF (ats_utilities.splash.progress_bar.ProgressBar attribute), 52

DEFAULT_CHAR_ON (ats_utilities.splash.progress_bar.ProgressBar attribute), 52

E

error_message () (in module ats_utilities.console_io.error), 28

ExtInfrastructure (class in ats_utilities.splash.ext_infrastructure), 50

F

file_path (ats_utilities.config_io.base_read.BaseReadConfig attribute), 25

file_path (ats_utilities.config_io.base_write.BaseWriteConfig attribute), 26

FileChecking (class in ats_utilities.config_io.base_check), 23

FORMAT_ERROR (ats_utilities.checker.ATSChecker attribute), 4

FunctionalSingleton () (in module ats_utilities.singleton.functional), 48

get_author_text () (ats_utilities.splash.ext_infrastructure.ExtInfrastructure method), 50

get_author_text ()

(*ats_utilities.splash.github_infrastructure.GitHubInfrastructure* *JSON2Object* (class in *ats_utilities.config_io.json.json2object*), 16
method), 51

get_info_text () (*ats_utilities.splash.ext_infrastructure.ExtInfrastructure* *JSONBase* (class in *ats_utilities.config_io.json*), 18
method), 50

get_info_text () (*ats_utilities.splash.github_infrastructure.GitHubInfrastructure* *JsonCLI* (class in *ats_utilities.cli.json_cli*), 8
method), 51

get_issue_text () (*ats_utilities.splash.ext_infrastructure.ExtInfrastructure* *License* (*ats_utilities.info.ats_licence.ATSLicence* at-
tribute), 39
method), 50

get_issue_text () (*ats_utilities.splash.github_infrastructure.GitHubInfrastructure* *LOG_DATE_FORMAT* (*ats_utilities.logging.ATSLogger*
attribute), 45
method), 51

GitHubInfrastructure (class in *ats_utilities.splash.github_infrastructure*),
51

I

Ini2Object (class in *ats_utilities.config_io.ini.ini2object*), 14

IniBase (class in *ats_utilities.config_io.ini*), 15

IniCLI (class in *ats_utilities.cli.ini_cli*), 7

is_correct () (*ats_utilities.info.ATSInfo* method), 42

is_file_ok () (*ats_utilities.config_io.base_check.FileChecking* *message* (*ats_utilities.console_io.error.ATSError* at-
tribute), 24
method), 24

is_not_none () (*ats_utilities.config_io.base_read.BaseReadConfig* *message* (*ats_utilities.console_io.success.ATSSuccess*
attribute), 25
method), 25

is_not_none () (*ats_utilities.config_io.base_write.BaseWriteConfig* *message* (*ats_utilities.console_io.verbose.ATSVerbose*
attribute), 26
method), 26

is_not_none () (*ats_utilities.console_io.error.ATSError* *message* (*ats_utilities.console_io.warning.ATSWarning*
attribute), 28
method), 28

is_not_none () (*ats_utilities.console_io.success.ATSSuccess* *MODES* (*ats_utilities.config_io.base_check.FileChecking*
attribute), 29
method), 29

is_not_none () (*ats_utilities.console_io.verbose.ATSVerbose*
method), 30

is_not_none () (*ats_utilities.console_io.warning.ATSWarning* *name* (*ats_utilities.info.ats_name.ATSName* attribute),
method), 31

is_not_none () (*ats_utilities.info.ats_build_date.ATSBuildDate* *NO_ERROR* (*ats_utilities.checker.ATSChecker* attribute),
method), 38

is_not_none () (*ats_utilities.info.ats_licence.ATSLicence*
method), 39

is_not_none () (*ats_utilities.info.ats_name.ATSName* *Object2Cfg* (class in *ats_utilities.config_io.cfg.object2cfg*), 12
method), 40

is_not_none () (*ats_utilities.info.ats_version.ATSVersion* *Object2Ini* (class in *ats_utilities.config_io.ini.object2ini*), 14
method), 41

is_tool_ok () (*ats_utilities.config_io.cfg.CfgBase* *Object2Json* (class in *ats_utilities.config_io.json.object2json*), 17
method), 13

is_tool_ok () (*ats_utilities.config_io.ini.IniBase* *Object2Xml* (class in *ats_utilities.config_io.xml.object2xml*), 18
method), 15

is_tool_ok () (*ats_utilities.config_io.json.JsonBase* *Object2Yaml* (class in *ats_utilities.config_io.yaml.object2yaml*),
method), 18

is_tool_ok () (*ats_utilities.config_io.xml.XmlBase*
method), 20

is_tool_ok () (*ats_utilities.config_io.yaml.YamlBase*
method), 23

J

JSON2Object (class in *ats_utilities.config_io.json.json2object*), 16

JSONBase (class in *ats_utilities.config_io.json*), 18

JsonCLI (class in *ats_utilities.cli.json_cli*), 8

L

License (*ats_utilities.info.ats_licence.ATSLicence* at-
tribute), 39

LOG_DATE_FORMAT (*ats_utilities.logging.ATSLogger*
attribute), 45

LOG_MSG_FORMAT (*ats_utilities.logging.ATSLogger* at-
tribute), 45

logger_file (*ats_utilities.logging.ats_logger_file.ATSLoggerFile*
attribute), 43

logger_name (*ats_utilities.logging.ats_logger_name.ATSLoggerName*
attribute), 43

logger_status (*ats_utilities.logging.ats_logger_status.ATSLoggerStatu*
attribute), 44

M

message (*ats_utilities.console_io.error.ATSError* at-
tribute), 28

message (*ats_utilities.console_io.success.ATSSuccess*
attribute), 29

message (*ats_utilities.console_io.verbose.ATSVerbose*
attribute), 30

message (*ats_utilities.console_io.warning.ATSWarning*
attribute), 31

MODES (*ats_utilities.config_io.base_check.FileChecking*
attribute), 24

N

name (*ats_utilities.info.ats_name.ATSName* attribute),
40

NO_ERROR (*ats_utilities.checker.ATSChecker* attribute),
4

O

Object2Cfg (class in *ats_utilities.config_io.cfg.object2cfg*), 12

Object2Ini (class in *ats_utilities.config_io.ini.object2ini*), 14

Object2Json (class in *ats_utilities.config_io.json.object2json*), 17

Object2Xml (class in *ats_utilities.config_io.xml.object2xml*), 18

Object2Yaml (class in *ats_utilities.config_io.yaml.object2yaml*),
21

P

parse_args () (*ats_utilities.cli.cfg_cli.CfgCLI*
method), 6

parse_args () (ats_utilities.cli.ini_cli.IniCLI SingletonMeta (class in ats_utilities.singleton.meta),
 method), 7 48
 parse_args () (ats_utilities.cli.json_cli.JsonCLI size () (ats_utilities.splash.terminal_properties.TerminalProperties
 method), 8 method), 54
 parse_args () (ats_utilities.cli.xml_cli.XmlCLI Splash (class in ats_utilities.splash), 55
 method), 9 SplashProperty (class in
 parse_args () (ats_utilities.cli.yaml_cli.YamlCLI ats_utilities.splash.splash_property), 53
 method), 10 success_message () (in module
 parse_args () (ats_utilities.option.ATSOptionParser ats_utilities.console_io.success), 29
 method), 46
 plot_progress () (ats_utilities.splash.progress_bar.ProgressBar T
 method), 52 TerminalProperties (class in
 priority_error () (ats_utilities.checker.ATSChecker ats_utilities.splash.terminal_properties),
 method), 5 54
 process (ats_utilities.cli.cfg_cli.CfgCLI attribute), 7 ThreadSafeSingletonMeta (class in
 process (ats_utilities.cli.ini_cli.IniCLI attribute), 7 ats_utilities.singleton.meta_thread), 49
 process (ats_utilities.cli.json_cli.JsonCLI attribute), 8 TYPE_ERROR (ats_utilities.checker.ATSChecker at-
 process (ats_utilities.cli.xml_cli.XmlCLI attribute), 9 tribute), 4
 process (ats_utilities.cli.yaml_cli.YamlCLI attribute),
 10
 ProgressBar (class in usage_message () (ats_utilities.checker.ATSChecker
 ats_utilities.splash.progress_bar), 52 method), 5

R

read_configuration (ats_utilities.config_io.base_read.BaseReadConfig validate () (ats_utilities.splash.splash_property.SplashProperty
 attribute), 25 method), 53
 read_configuration () (ats_utilities.config_io.cfg_cfg2object.Cfg2Object VALUE_ERROR (ats_utilities.checker.ATSChecker
 method), 11 attribute), 5
 read_configuration () (ats_utilities.config_io.ini_ini2object.Ini2Object VERBOSE (ats_utilities.cli.cfg_cli.CfgCLI attribute), 6
 method), 14 VERBOSE (ats_utilities.cli.ini_cli.IniCLI attribute), 7
 read_configuration () (ats_utilities.config_io.json_json2object.Json2Object VERBOSE (ats_utilities.cli.json_cli.JsonCLI attribute), 8
 method), 16 VERBOSE (ats_utilities.cli.xml_cli.XmlCLI attribute), 9
 read_configuration () (ats_utilities.config_io.xml_xml2object.Xml2Object VERBOSE (ats_utilities.cli.yaml_cli.YamlCLI attribute),
 method), 20 10
 read_configuration () (ats_utilities.config_io.json_json2object.Json2Object VERBOSE (ats_utilities.config_io.base_check.FileChecking
 method), 16 attribute), 24
 read_configuration () (ats_utilities.config_io.xml_xml2object.Xml2Object VERBOSE (ats_utilities.config_io.base_read.BaseReadConfig
 method), 20 attribute), 25
 read_configuration () (ats_utilities.config_io.yaml_yaml2object.Yaml2Object VERBOSE (ats_utilities.config_io.base_write.BaseWriteConfig
 method), 22 attribute), 26
 ROOT_PACKAGE_NAME (ats_utilities.VerboseRoot at- VERBOSE (ats_utilities.config_io.cfg_cfg2object.Cfg2Object
 tribute), 56 attribute), 11

S

set_and_plot () (ats_utilities.splash.progress_bar.ProgressBar VERBOSE (ats_utilities.config_io.base_check.FileChecking
 method), 52 attribute), 24
 set_level () (ats_utilities.splash.progress_bar.ProgressBar VERBOSE (ats_utilities.config_io.base_read.BaseReadConfig
 method), 53 attribute), 25
 show_base_info () (ats_utilities.info.ATSInfo VERBOSE (ats_utilities.config_io.base_write.BaseWriteConfig
 method), 42 attribute), 26
 Singleton (class in ats_utilities.singleton.base), 47 VERBOSE (ats_utilities.config_io.cfg_cfg2object.Cfg2Object
 attribute), 11
 VERBOSE (ats_utilities.config_io.cfg_cfg2object.Cfg2Object
 attribute), 12
 VERBOSE (ats_utilities.config_io.ConfigFile attribute),
 27
 VERBOSE (ats_utilities.config_io.ini_ini2object.Ini2Object
 attribute), 14
 VERBOSE (ats_utilities.config_io.ini.IniBase attribute),
 15
 VERBOSE (ats_utilities.config_io.ini.object2ini.Object2Ini
 attribute), 15

VERBOSE (*ats_utilities.config_io.json.json2object.Json2Object* attribute), 16
 VERBOSE (*ats_utilities.config_io.json.JsonBase* attribute), 18
 VERBOSE (*ats_utilities.config_io.json.object2json.Object2Json* attribute), 17
 VERBOSE (*ats_utilities.config_io.xml.object2xml.Object2Xml* attribute), 19
 VERBOSE (*ats_utilities.config_io.xml.xml2object.Xml2Object* attribute), 20
 VERBOSE (*ats_utilities.config_io.xml.XmlBase* attribute), 20
 VERBOSE (*ats_utilities.config_io.yaml.object2yaml.Object2Yaml* attribute), 21
 VERBOSE (*ats_utilities.config_io.yaml.yaml2object.Yaml2Object* attribute), 22
 VERBOSE (*ats_utilities.config_io.yaml.YamlBase* attribute), 23
 VERBOSE (*ats_utilities.info.ats_build_date.ATSBuildDate* attribute), 37
 VERBOSE (*ats_utilities.info.ats_info_ok.ATSInfoOk* attribute), 38
 VERBOSE (*ats_utilities.info.ats_licence.ATSLicence* attribute), 39
 VERBOSE (*ats_utilities.info.ats_name.ATSName* attribute), 40
 VERBOSE (*ats_utilities.info.ats_version.ATSVersion* attribute), 41
 VERBOSE (*ats_utilities.info.ATSInfo* attribute), 42
 VERBOSE (*ats_utilities.logging.ats_logger_file.ATSLoggerFile* attribute), 43
 VERBOSE (*ats_utilities.logging.ats_logger_name.ATSLoggerName* attribute), 43
 VERBOSE (*ats_utilities.logging.ats_logger_status.ATSLoggerStatus* attribute), 44
 VERBOSE (*ats_utilities.logging.ATSLogger* attribute), 45
 VERBOSE (*ats_utilities.option.ATSOptionParser* attribute), 46
 VERBOSE (*ats_utilities.splash.ext_infrastructure.ExtInfrastructure* attribute), 50
 VERBOSE (*ats_utilities.splash.github_infrastructure.GitHubInfrastructure* attribute), 51
 VERBOSE (*ats_utilities.splash.Splash* attribute), 55
 VERBOSE (*ats_utilities.splash.splash_property.SplashProperty* attribute), 53
 VERBOSE (*ats_utilities.splash.terminal_properties.TerminalProperties* attribute), 54
 verbose_message () (in module *ats_utilities.console_io.verbose*), 30
 VerboseRoot (class in *ats_utilities*), 56
 version (*ats_utilities.info.ats_version.ATSVersion* attribute), 41

W
 warning_message () (in module *ats_utilities.console_io.warning*), 31
 write_configuration (*ats_utilities.config_io.base_write.BaseWriteConfig* attribute), 26
 write_configuration () (*ats_utilities.config_io.cfg.object2cfg.Object2Cfg* method), 12
 write_configuration () (*ats_utilities.config_io.ini.object2ini.Object2Ini* method), 15
 write_configuration () (*ats_utilities.config_io.json.object2json.Object2Json* method), 17
 write_configuration () (*ats_utilities.config_io.xml.object2xml.Object2Xml* method), 19
 write_configuration () (*ats_utilities.config_io.yaml.object2yaml.Object2Yaml* method), 21
 write_log () (*ats_utilities.logging.ATSLogger* method), 45

X
 Xml2Object (class in *ats_utilities.config_io.xml.xml2object*), 19
 XmlBase (class in *ats_utilities.config_io.xml*), 20
 XmlCLI (class in *ats_utilities.cli.xml_cli*), 9

Y
 Yaml2Object (class in *ats_utilities.config_io.yaml.yaml2object*), 22
 YamlBase (class in *ats_utilities.config_io.yaml*), 23
 YamlCLI (class in *ats_utilities.cli.yaml_cli*), 10