rpymostat-common Documentation

Release 0.1.0

Jason Antman

Dec 24, 2017

Contents

Master: build failing Develop: build unknown

CHAPTER 1

Contents

1.1 rpymostat_common

1.1.1 rpymostat_common package

1.1.1.1 Submodules

rpymostat_common.discovery module

class `rpymostat_common.discovery.`**`EngineDiscoverer`**

Bases: `object`

Class to discover RPyMostat Engine over the network (mDNS / Avahi / DNS-SD)

`rpymostat_common.discovery.`**`discover_engine`**`()`

Discover the RPyMostat Engine over the network, using mDNS / DNS-SD.

Returns 2-tuple of (engine_addr, engine_port) if discovered before a timeout is reached, otherwise raise a DiscoveryTimeoutException.

rpymostat_common.loader module

`rpymostat_common.loader.`**`_get_varnames`**`(`*klass*`)`

Return a dict of variable names that klass's init method takes, to string descriptions of them (if present).

Parameters **`klass`** (*`abc.ABCMeta`*) – the class to get varnames for (from its __init__ method)

Returns dict

`rpymostat_common.loader.`**`_parse_docstring`**`(`*docstring*`)`

Given a docstring, attempt to parse out all `:param foo:` and `:type foo:` directives and their matching strings, collapsing whitespace. Return a dict of keys 'params' and 'types', each being a dict of name to string.

Parameters **`docstring`** (*`str`*) – docstring to parse

Return type dict

`rpymostat_common.loader.`**`list_classes`**`(`*classes*`)`

Given a list of class objects, print their names, along with their _description attributes (if present) and any arguments they accept. Used in building dynamic CLI help.

`rpymostat_common.loader.`**`load_classes`**`(`*entrypoint_name*, *superclass=None*`)`

Attempt to load all pkg_resources entrypoints matching the given name, and return a list of the objects they load (usually classes). If `superclass` is specified, restrict the returned list to those which are subclasses of `superclass`.

Parameters

- **`entrypoint_name`** (*`str`*) – name of the entrypoint to load
- **`superclass`** (`class or classinfo`) – if specified, restrict the return value to only subclasses of this class / classinfo

Returns list of loaded entrypoints (usually classes)

Return type list

rpymostat_common.unique_ids module

`class` `rpymostat_common.unique_ids.`**`SystemID`**

Bases: `object`

Determine and retrieve a unique system ID for the hardware this is running on.

`id_methods = ['raspberrypi_cpu', 'uuid_getnode']`

`id_string`

Find/calculate and return the unique system ID string for the hardware this is running on.

Internally, this calls all method whose names are listed in *`id_methods`*, in order, and returns the value of the first one that returned something other than None.

Returns unique, never-changing system ID

Return type str

`proc_cpuinfo_hw_re = <_sre.SRE_Pattern object>`

`proc_cpuinfo_rev_re = <_sre.SRE_Pattern object>`

`proc_cpuinfo_serial_re = <_sre.SRE_Pattern object>`

`random_fallback``()`

Generate a host ID using a random UUID via Python's `uuid.uuid4()`. Used as a fallback when the ID can't be determined using any other method.

Returns random UUID

Return type str

`raspberrypi_cpu``()`

If this system is a Raspberry Pi, get its model and (CPU) serial number.

Thanks to: http://elinux.org/RPi_HardwareHistory#Board_Revision_History

Returns RaspberryPi serial number

Return type str

`rpi_hardware = ['BCM2708', 'BCM2709']`

rpi_revisions = {'0015': 'A+ 1.1 256MB (Embest)', '0014': 'Compute Module 1.0 512MB

uuid_getnode()

Determine this system's UUID via Python's `uuid.getnode()` (slow) method.

Returns hardware system ID from Python's `uuid.getnode()`

Return type str

rpymostat_common.version module

1.2 Changelog

1.2.1 x.y.z (YYYY-MM-DD)

- something

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

2.1 License

rpymostat-common is licensed under the GNU Affero General Public License, version 3 or later.

Python Module Index

Index