
dlipower Documentation

Release 0.0.0

Dwight Hubbard

Jun 16, 2018

Contents

1	The dlipower module	3
1.1	dlipower.dlipower	3
1.1.1	Digital Loggers Web Power Switch management	3
1.2	dlipower.debug	5
1.3	Classes	5
1.3.1	Outlet	5
1.3.2	Powerswitch	6
2	COMMAND LINE USAGE	7
2.1	dlipower script	7
2.2	fence_dli	8
3	Indices and tables	9
	Python Module Index	11

Contents:

The dlipower module

The `dlipower` module provides a python api to manage digital loggers networked power switches.

1.1 `dlipower.dlipower`

The `dlipower.dlipower` module provides the core functionality of the `dlipower` module.

1.1.1 Digital Loggers Web Power Switch management

Description: This is both a module and a script

The module provides a python class named `powerswitch` that allows managing the web power switch from python programs.

When run as a script this acts as a command line utility to manage the DLI Power switch.

This module has been tested against the following Digital Loggers Power network power switches:

WebPowerSwitch II WebPowerSwitch III WebPowerSwitch IV WebPowerSwitch V
Ethernet Power Controller III

Author: Dwight Hubbard d@dhub.me

exception `dlipower.dlipower.DLIPowerException`

Bases: `exceptions.Exception`

An error occurred talking the the DLI Power switch

class `dlipower.dlipower.Outlet` (*switch, outlet_number, description=None, state=None*)

Bases: `object`

A power outlet class

name

Return the name or description of the outlet

off ()
Turn the outlet off

on ()
Turn the outlet on

rename (*new_name*)
Rename the outlet :param new_name: New name for the outlet :return:

state
Return the outlet state

class `dlipower.dlipower.PowerSwitch` (*userid=None, password=None, hostname=None, timeout=None, cycletime=None, retries=None*)

Bases: `object`

Powerswitch class to manage the Digital Loggers Web power switch

command_on_outlets (*command, outlets*)
If a single outlet is passed, handle it as a single outlet and pass back the return code. Otherwise run the operation on multiple outlets in parallel the return code will be failure if any operation fails. Operations that return a string will return a list of strings.

cycle (*outlet=0*)
Cycle power to an outlet False = Power off Success True = Power off Fail Note, does not return any status info about the power on part of the operation by design

determine_outlet (*outlet=None*)
Get the correct outlet number from the outlet passed in, this allows specifying the outlet by the name and making sure the returned outlet is an int

get_outlet_name (*outlet=0*)
Return the name of the outlet

geturl (*url='index.htm'*)
Get a URL from the userid/password protected powerswitch page Return None on failure

load_configuration ()
Return a configuration dictionary

off (*outlet=0*)
Turn off a power to an outlet False = Success True = Fail

on (*outlet=0*)
Turn on power to an outlet False = Success True = Fail

printstatus ()
Print the status off all the outlets as a table to stdout

save_configuration ()
Update the configuration file with the object's settings

set_outlet_name (*outlet=0, name='Unknown'*)
Set the name of an outlet

status (*outlet=1*)
Return the status of an outlet, returned value will be one of: ON, OFF, Unknown

statuslist ()
Return the status of all outlets in a list, each item will contain 3 items plugnumber, hostname and state

verify ()
Verify we can reach the switch, returns true if ok

1.2 dlipower.debug

The `dlipower.debug` module provides debug information for the dlipower module.

DLIPower Debug Utilities

This module contains utility functions useful for troubleshooting dlipower.

This module can be run from the command line using the following command:

```
python -m dlipower.debug
```

This will output information like the following:

```
$ python -m dlipower.debug

dlipower debug information:
  Version: 0.7.120
  Module Path: /home/dwight/github/python-dlipower/dlipower

  Source Code Information
    Git Source URL: https://github.com/dwighthubbard/python-dlipower/tree/
    ↪9c3bb943124d5d9767403960fdf6a622cbea5128
    Git Hash: 9c3bb943124d5d9767403960fdf6a622cbea5128
    Git Version: 0.7.120
    Git Origin: https://github.com/dwighthubbard/python-dlipower.git
    Git Branch: master
```

When run from the command line this will print a dump of information about the module and it's build information.

`dlipower.debug.debug_info()`

Return a multi-line string with the debug information :return:

`dlipower.debug.debug_info_list()`

Return a list with the debug information :return:

`dlipower.debug.print_debug_info()`

Display information about the redislite build, and redis-server on stdout. :return:

1.3 Classes

1.3.1 Outlet

class `dlipower.Outlet` (*switch, outlet_number, description=None, state=None*)

A power outlet class

name

Return the name or description of the outlet

off()

Turn the outlet off

on()

Turn the outlet on

rename (*new_name*)

Rename the outlet :param new_name: New name for the outlet :return:

state

Return the outlet state

1.3.2 Powerswitch

class dlipower.**PowerSwitch** (*userid=None, password=None, hostname=None, timeout=None, cycletime=None, retries=None*)

Powerswitch class to manage the Digital Loggers Web power switch

command_on_outlets (*command, outlets*)

If a single outlet is passed, handle it as a single outlet and pass back the return code. Otherwise run the operation on multiple outlets in parallel the return code will be failure if any operation fails. Operations that return a string will return a list of strings.

cycle (*outlet=0*)

Cycle power to an outlet False = Power off Success True = Power off Fail Note, does not return any status info about the power on part of the operation by design

determine_outlet (*outlet=None*)

Get the correct outlet number from the outlet passed in, this allows specifying the outlet by the name and making sure the returned outlet is an int

get_outlet_name (*outlet=0*)

Return the name of the outlet

geturl (*url='index.htm'*)

Get a URL from the userid/password protected powerswitch page Return None on failure

load_configuration ()

Return a configuration dictionary

off (*outlet=0*)

Turn off a power to an outlet False = Success True = Fail

on (*outlet=0*)

Turn on power to an outlet False = Success True = Fail

printstatus ()

Print the status off all the outlets as a table to stdout

save_configuration ()

Update the configuration file with the object's settings

set_outlet_name (*outlet=0, name='Unknown'*)

Set the name of an outlet

status (*outlet=1*)

Return the status of an outlet, returned value will be one of: ON, OFF, Unknown

statuslist ()

Return the status of all outlets in a list, each item will contain 3 items plugnumber, hostname and state

verify ()

Verify we can reach the switch, returns true if ok

COMMAND LINE USAGE

The dlipower package provides two scripts.

2.1 dlipower script

This script provides a command line interface to the dli power switches.

```
Usage: dlipower [options] [status|on|off|cycle|get_outlet_name|set_outlet_name]
↪ [range] [newname]
```

Options:

-h, --help	show this help message and exit
--hostname=HOSTNAME	hostname/ip of the power switch (default none)
--timeout=TIMEOUT	Timeout for value for power switch communication (default none)
--cycletime=CYCLETIME	Delay between off/on states for power cycle operations (default none)
--user=USER	userid to connect with (default none)
--password=PASSWORD	password (default none)
--save_settings	Save the settings to the configuration file
--quiet	Suppress error output

Arguments:

range - One or more ports delimited by commas

Example:

1,3,5-9 (Refers to outlets 1,3,5,6,7,8,9)

newname - The name to rename the outlet to``

2.2 fence_dli

The fence_dli script is a linux cluster compatible stonith fencing script for dlipower switches.

CHAPTER 3

Indices and tables

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

d

`dlipower.debug`, 5

`dlipower.dlipower`, 3

C

command_on_outlets() (dlipower.dlipower.PowerSwitch method), 4
command_on_outlets() (dlipower.PowerSwitch method), 6
cycle() (dlipower.dlipower.PowerSwitch method), 4
cycle() (dlipower.PowerSwitch method), 6

D

debug_info() (in module dlipower.debug), 5
debug_info_list() (in module dlipower.debug), 5
determine_outlet() (dlipower.dlipower.PowerSwitch method), 4
determine_outlet() (dlipower.PowerSwitch method), 6
dlipower.debug (module), 5
dlipower.dlipower (module), 3
DLIPowerException, 3

G

get_outlet_name() (dlipower.dlipower.PowerSwitch method), 4
get_outlet_name() (dlipower.PowerSwitch method), 6
geturl() (dlipower.dlipower.PowerSwitch method), 4
geturl() (dlipower.PowerSwitch method), 6

L

load_configuration() (dlipower.dlipower.PowerSwitch method), 4
load_configuration() (dlipower.PowerSwitch method), 6

N

name (dlipower.dlipower.Outlet attribute), 3
name (dlipower.Outlet attribute), 5

O

off() (dlipower.dlipower.Outlet method), 4
off() (dlipower.dlipower.PowerSwitch method), 4
off() (dlipower.Outlet method), 5
off() (dlipower.PowerSwitch method), 6

on() (dlipower.dlipower.Outlet method), 4
on() (dlipower.dlipower.PowerSwitch method), 4
on() (dlipower.Outlet method), 5
on() (dlipower.PowerSwitch method), 6
Outlet (class in dlipower), 5
Outlet (class in dlipower.dlipower), 3

P

PowerSwitch (class in dlipower), 6
PowerSwitch (class in dlipower.dlipower), 4
print_debug_info() (in module dlipower.debug), 5
printstatus() (dlipower.dlipower.PowerSwitch method), 4
printstatus() (dlipower.PowerSwitch method), 6

R

rename() (dlipower.dlipower.Outlet method), 4
rename() (dlipower.Outlet method), 5

S

save_configuration() (dlipower.dlipower.PowerSwitch method), 4
save_configuration() (dlipower.PowerSwitch method), 6
set_outlet_name() (dlipower.dlipower.PowerSwitch method), 4
set_outlet_name() (dlipower.PowerSwitch method), 6
state (dlipower.dlipower.Outlet attribute), 4
state (dlipower.Outlet attribute), 5
status() (dlipower.dlipower.PowerSwitch method), 4
status() (dlipower.PowerSwitch method), 6
statuslist() (dlipower.dlipower.PowerSwitch method), 4
statuslist() (dlipower.PowerSwitch method), 6

V

verify() (dlipower.dlipower.PowerSwitch method), 4
verify() (dlipower.PowerSwitch method), 6