
dlipower Documentation

Release 1.0.13

Dwight Hubbard

July 04, 2016

1	The dlipower module	3
1.1	dlipower.dlipower	3
1.1.1	Notes	3
1.1.2	Examples	3
1.2	dlipower.debug	6
1.3	Classes	6
1.3.1	Outlet	6
1.3.2	Powerswitch	7
2	COMMAND LINE USAGE	9
2.1	dlipower script	9
2.2	fence_dli	9
3	Indices and tables	11
	Python Module Index	13

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. Digital Loggers Web Power Switch Management

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.

1.1.1 Notes

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

1.1.2 Examples

Connecting to a Digital Loggers Power switch

```
>>> from dlipower import PowerSwitch
>>> switch = PowerSwitch(hostname='lpc.digital-loggers.com', userid='admin', password='4321')
```

Getting the power state (status) from the switch Printing the switch object will print a table with the Outlet Number, Name and Power State

```
>>> switch
DLIPowerSwitch at lpc.digital-loggers.com
Outlet  Name                State
1       Battery Charger         OFF
2       K3 Power ON              ON
3       Cisco Router             OFF
4       WISP access poi         ON
5       Shack Computer          OFF
6       Router                   OFF
```

```
7      2TB Drive      ON
8      Cable Modem1  ON
```

Getting the name and powerswitch of the first outlet The PowerSwitch has a series of Outlet objects, they will display their name and state if printed.

```
>>> switch[0]
<dlipower_outlet 'Traffic light:OFF'>
```

Renaming the first outlet Changing the “name” attribute of an outlet will rename the outlet on the powerswitch.

```
>>> switch[0].name = 'Battery Charger'
>>> switch[0]
<dlipower_outlet 'Battery Charger:OFF'>
```

Turning the first outlet on Individual outlets can be accessed uses normal list slicing operators.

```
>>> switch[0].on()
False
>>> switch[0]
<dlipower_outlet 'Battery Charger:ON'>
```

Turning all outlets off The PowerSwitch() object supports iterating over the available outlets.

```
>>> for outlet in switch:
...     outlet.off()
...
False
False
False
False
False
False
False
False
False
>>> switch
DLIPowerSwitch at lpc.digital-loggers.com
Outlet  Name          State
1       Battery Charger OFF
2       K3 Power ON    OFF
3       Cisco Router  OFF
4       WISP access poi OFF
5       Shack Computer OFF
6       Router        OFF
7       2TB Drive     OFF
8       Cable Modem1  OFF
```

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, cycle-time=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.

Indices and tables

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

d

`dlipower.debug`, 6

`dlipower.dlipower`, 3

C

command_on_outlets() (dlipower.dlipower.PowerSwitch method), 5

command_on_outlets() (dlipower.PowerSwitch method), 7

cycle() (dlipower.dlipower.PowerSwitch method), 5

cycle() (dlipower.PowerSwitch method), 7

D

debug_info() (in module dlipower.debug), 6

debug_info_list() (in module dlipower.debug), 6

determine_outlet() (dlipower.dlipower.PowerSwitch method), 5

determine_outlet() (dlipower.PowerSwitch method), 7

dlipower.debug (module), 6

dlipower.dlipower (module), 3

DLIPowerException, 4

G

get_outlet_name() (dlipower.dlipower.PowerSwitch method), 5

get_outlet_name() (dlipower.PowerSwitch method), 7

geturl() (dlipower.dlipower.PowerSwitch method), 5

geturl() (dlipower.PowerSwitch method), 7

L

load_configuration() (dlipower.dlipower.PowerSwitch method), 5

load_configuration() (dlipower.PowerSwitch method), 7

N

name (dlipower.dlipower.Outlet attribute), 4

name (dlipower.Outlet attribute), 6

O

off() (dlipower.dlipower.Outlet method), 4

off() (dlipower.dlipower.PowerSwitch method), 5

off() (dlipower.Outlet method), 6

off() (dlipower.PowerSwitch method), 7

on() (dlipower.dlipower.Outlet method), 5

on() (dlipower.dlipower.PowerSwitch method), 5

on() (dlipower.Outlet method), 6

on() (dlipower.PowerSwitch method), 7

Outlet (class in dlipower), 6

Outlet (class in dlipower.dlipower), 4

P

PowerSwitch (class in dlipower), 7

PowerSwitch (class in dlipower.dlipower), 5

print_debug_info() (in module dlipower.debug), 6

printstatus() (dlipower.dlipower.PowerSwitch method), 5

printstatus() (dlipower.PowerSwitch method), 7

R

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

rename() (dlipower.Outlet method), 6

S

save_configuration() (dlipower.dlipower.PowerSwitch method), 5

save_configuration() (dlipower.PowerSwitch method), 7

set_outlet_name() (dlipower.dlipower.PowerSwitch method), 5

set_outlet_name() (dlipower.PowerSwitch method), 7

state (dlipower.dlipower.Outlet attribute), 5

state (dlipower.Outlet attribute), 6

status() (dlipower.dlipower.PowerSwitch method), 5

status() (dlipower.PowerSwitch method), 7

statuslist() (dlipower.dlipower.PowerSwitch method), 5

statuslist() (dlipower.PowerSwitch method), 7

V

verify() (dlipower.dlipower.PowerSwitch method), 5

verify() (dlipower.PowerSwitch method), 7