OpticalSwitch

Library version:	RENAT 0.1.7
Library scope:	test suite
Named arguments:	supported

Introduction

A library provides control for L1 Optical Switch (currently Calient)

`OpticalSwitch` is a RENAT library that provides control for L1 optical switch. Currently the library only supports Calient.

Table of Contents

Master file

The L1 switch provides a mechanism to remotely connect device interface. Each device interface has been wired to L1 switch already. The connection was described in the master file located specific by *calient-master-path* in the configuration file *renat/config/config.yaml*.

The master file includes several Calients in each tab. The column meaning and order is trivial.

Connection file Format

Keywords *Load From File*, *Clear By File* and *Save To File* use the x-connection file. The connection has following rules:

Connection files are text files and have the following format:

```
# this is the comment
device1,interface1,-,device2,interface2
device1,interface1,>,device2,interface2
```

The separator `-` means a bidirection connection and `>` means a unidirection connection. For a unidirection connection, `device1/interface1` TX will be connected to `device2/interface2` RX.

Note: The separator character must be surrounded by spaces or commas.

The connection file also support jinja2 template format. After the template is evaluated, comment could be used by `comment char`

There is no need to specify which L1 switch for the x-connection. The system will automatically find the appropriate switch.

Shortcuts

Add · **C**lear By File · **C**lose All · **C**onnect All · **D**elete · **G**et Connection Info · **L**oad From File · **R**ead Map · **S**ave To File

Keywords

Keyword	Arguments	Documentation
Add	*dev1*, *intf1*, *dev2*, *intf2*, *direction=bi*, *force=False*	Adds x-connection between `dev1:intf1` and `dev2:intf2` `direction` is `bi` for bi-direction or `uni` for uni-direction. If `direction` is `uni`, the tx of `dev 1:port 1` will be connected to `dev 2:port 2`. With `force` mode, existed connection that use those ports will be deleted. Without `force` mode, an existed connection will make the keyword fails Examples: `OpticalSwitch.Add` `mx2008-31-33` `xe-3/0/0` `mx2008-31-33` `xe-3/0/1` `bi` `${TRUE}` **Note**: when `force` is `False` but the current ports is owned by the same connection endpoints, keyword will succeed. **Note:** For a bidirection connection, 2 single uni-direction connection will be made instead of 1 bi-direction connection. This will make the link could be simulated tx/rx failure later.
Clear By File	*file_name=*, *comment=#*	Clears all x-connections defined in the *connection file* Default *connection file* is defined in `optic/connection` of `config/local.yaml`
Close All		
Connect All		
Delete	*dev1*, *intf1*, *dev2*, *intf2*, *direction=bi*	Deletes the connection between `dev1:intf1 - dev2:intf2` Examples: `OpticalSwitch.Delete` `mx2008-31-33` `xe-3/0/1` `mx2008-31-33` `xe-3/0/1` `uni`
Get Connection	*dev*, *intf*	Returns information of the optic switch port that connected to `dev:intf`. The information is in jason

Info		format. Examples: OpticalSwitch.*Get Connection Info* | mx2008-31-33 | xe-3/0/1 return information looks like below: result = {u'outoc': u'NOHW', u'outopwdh': u'-20.0', u'inos': u'OOS', u'outalias': u'', u'inowner': u'TRANSIT', u'outopwct': u'-23.0', u'inpower': u'-3.4', u'inas': u'IS', u'outpower': u'-4.8', u'outas': u'OOS-NP', u'inopt': u'-17.0', u'inopth': u'13.0', u'incircuit': u'3.3.1>3.3.2', u'inalias': u'', u'inoc': u'NOHW', u'inoptc': u'-20.0', u'outos': u'OOS', u'port': u'3.3.1', u'outowner': u'NONE', u'outcircuit': u''}
Load From File	*file_name=*, *force=True*, *comment=#*	Loads the connection file and set the connections `filename` is the name of the connection file under the current config folder. If `filename` is empty, the value of `optic/connection` from `config/local.yaml` will be used. The `connection file` supports `jinja2` template language. Besides, `#` is the default comment char which could be changed The format of `connection file` follows: ▪ each connection is described by 1 line ▪ `source` and `destination` are separated by `` `- *or* > `, *which mean* `bidirection` or `unidirection` (unidirection connects `source tx` to `dest rx` Connection file sample: device1:port1 - device2:port2 device1:port3 > device2:port Examples: OpticalSwitch.*Load From File* OpticalSwitch.*Load From File* | save1.conn
Read Map		Reads the master port map file Make lower for all informations.
Save To File	*file_name*	Saves the current connection of all devices in this test. By default, all interfaces of the devices are save. If a connection file is given, only interfaces specified in the connection file are saved Examples: OpticalSwitch.*Save To File* | save1.conn

Altogether 9 keywords.
Generated by Libdoc on 2018-03-20 02:58:10.