juniper

Library scope: global
Named arguments: supported

Introduction

Provides keywords for Juniper platform

Notes: Ignore the *self* parameters when using those keywords.

Shortcuts

Copy Config · **C**opy File · **C**reate Best Path Select Data · **D**isable Interface · **E**nable Interface · **F**lap Interface · **G**et Chassis Serial · **G**et Cli Mode · **G**et Config · **G**et Current Datetime · **G**et File · **G**et Intf Addr · **G**et Route Number · **G**et Version · **L**ink Status · **L**oad Config · **N**umber Of Bgp Neighbor · **N**umber Of Ospf Neighbor · **N**umber Of Ospf3 Neighbor · **P**ush Config

Keywords

Keyword	Arguments	Documentation
Copy Config	*self*, *dst_name=None*	Gets the configuration file of the router This keyword directly copy the configuration file from the router
Copy File	*self*, *src_path*, *filename=None*, *pre_config=None*, *pos_config=None*	Copies a file from router Parameters: ▪ *src_path*: a absolute path insides the router ▪ *filename*: a file name under `result` folder
Create Best Path Select Data	*self*, *route_content*, *output_excel=best.xlsx*	Creates the matrix of best path selection Provides the test described in *smb://10.128.3.91/SharePoint01/31_VerificationRoom/31_13_検証環境セット/BGP-Best-Path-SelectionのAll-in-One設定_20161118改良/* The test uses predefined Ixia config and follows predefined steps
Disable Interface	*self*, *intf*	Disables an interface `intf`
Enable Interface	*self*, *intf*	Enables an interface `intf`
Flap Interface	*self*, *intf*, *time_str=10s*	Simulates an interface flap for interface `intf` Disables the interface and wait for a while before turning it up again
Get Chassis Serial	*self*	Returns the serial number of the chassis
Get Cli Mode	*self*	Returns current mode of the CLI. Return value is `config` for configuration mode or `command` for command mode
Get Config	*self*, *dst_name=None*	Gets the current configuration file of the router to current `result` folder. Default `dst_name` is `juniper.conf.gz` This keyword push the configuration from **FROM** the router to the RENAT server

Get Current Datetime	*self*, *time_format=%H:%M:%S*, *delta_time=0s*, *dir=+*, ***kwargs*	Returns the current date time with vendor format `delta_time` will be added or subtracted to current time, default is `0s`
		`time_format` decides the time part of the output. Example result are :
		`May 24 04:14:25`
		`May  4 04:14:25`
		Note: The date part is padded by space, and the result is allways 15 characters

Get File	*self*, *src_file*, *dst_file=None*	Gets a file from router Parameters: ▪ *src_file*: a absolute path insides the router ▪ *dst_file*: a file name under `result` folder if *dst_file* is not defined, it will be the filename of the *src_file*. The keyword copy the specific file **FROM** the router to the RENAT server
Get Intf Addr	*self*, *intf_name*, *family=inet*	Returns the tuple of address and netmask of an interface `family` should be `inet` or `inet6` If the address is not set, (",") will be returned.
Get Route Number	*self*, *table=inet.0*	Returns number of active route in the `table` `table` could be `inet.0` or `inet.6`
Get Version	*self*	Returns router version information

Link Status	*self*, *if_name*	Returns link physical status as string (aka: "up down", "up up")
Load Config	*self*, *mode=set*, *config_file=*, *confirm=0s*, *vars=*, *err_match=(error:|unknown command:)*	Loads configuration to a router. Usable `mode` is `set`, `override`, `merge` and `replace` `set` mode uses configuration that contains `set` command. Mode `override`, `merge` and `replace` use ordinary JunOS configuration file with appropriate mode. `config_file` is a configuration file inside the `config` folder of the current test case. Config file could includes jinja2 template. The template will be evalued with *LOCAL*, *GLOBAL* and varibles defined by *vars*. The *vars* has the format: var1=value1,var2=value2 ... If the loading has no error that match the `error_match`, the configuration will be commited. The keywordl waits for `confirm` seconds before rollback the commited configuration. A zero value indicates an immediatly commit
Number Of Bgp Neighbor	*self*, *state=Established*, *cmd=show bgp neighbor | match "Type"*	Returns number of BGP neighbor in `state` state
Number Of Ospf Neighbor	*self*, *state=Full*, *cmd=show ospf neighbor*	Returns number of OPSF neighbors with status `state`
Number Of Ospf3 Neighbor	*self*, *state=Full*, *cmd=show ospf3 neighbor*	Returns number of OPSFv3 neighbors with status `state`
Push Config	*self*, *mode=set*, *config_file=*, *pre_config=None*, *pos_config=None*, *confirm=0s*, *vars=*, *err_match=(error:|unknown command:)*	Pushes configuration directly to the router Usable `mode` is `set`, `override`, `merge` and `replace` `set` mode uses configuration that contains `set` command. Mode `override`, `merge` and `replace` use ordinary JunOS configuration file with appropriate mode. `config_file` is a configuration file inside the `config` folder of the current test case. Config file could includes jinja2 template. The template will be evalued with *LOCAL*, *GLOBAL* and varibles defined by *vars*. The *vars* has the format: var1=value1,var2=value2 ... If the loading has no error that match the `error_match`, the configuration will be commited. The keywordl waits for `confirm` seconds before rollback the commited configuration. A zero value indicates an immediatly commit *pre_config* and *pos_config* is extra configuration commands separated by *;* that would be excuted before and after the configuration is loaded. It could be used to add extra firewall rules to the router. **Note**: by default the keyword will activate the SSH service on the router if the service is not activated and disable the service after loading the configuration.