Contents

Using Python

Comware 7 provides a built-in Python interpreter that supports the following items:

- Python 2.7 commands.
- Python 2.7 standard API.
- Comware 7 extended API. For more information about the Comware 7 extended API, see "Comware 7 extended Python API."
- Python scripts. You can use a Python script to configure the system.

Entering the Python shell

To use Python commands and APIs, you must enter the Python shell.

To enter the Python shell:

Task	Command
Enter the Python shell from user view.	**python**

Executing a Python script

Execute a Python script in user view.

Task	Command
Execute a Python script.	**python** *filename*

Exiting the Python shell

Execute this command in the Python shell.

Task	Command
Exit the Python shell.	**exit()**

Python usage example

Network requirements

Use a Python script to perform the following tasks:

- Download configuration files **main.cfg** and **backup.cfg** to the device.
- Configure the files as the main and backup configuration files for the next startup.

Figure 1 Network diagram

Usage procedure

Use a text editor on the PC to configure Python script **test.py** as follows:

```
#!usr/bin/python
import comware

comware.Transfer('tftp', '192.168.1.26', 'main.cfg', 'flash:/main.cfg')
comware.Transfer('tftp', '192.168.1.26', 'backup.cfg', 'flash:/backup.cfg')
comware.CLI('startup saved-configuration flash:/main.cfg main ;startup
saved-configuration flash:/backup.cfg backup')
```

Use TFTP to download the script to the device.

```
<Sysname> tftp 192.168.1.26 get test.py
```

Execute the script.

```
<Sysname> python flash:/test.py
<Sysname>startup saved-configuration flash:/main.cfg main
Please wait...... Done.
<Sysname>startup saved-configuration flash:/backup.cfg backup
Please wait...... Done.
```

Verifying the configuration

Display startup configuration files.

```
<Sysname> display startup
 Current startup saved-configuration file: flash:/startup.cfg
 Next main startup saved-configuration file: flash:/main.cfg
 Next backup startup saved-configuration file: flash:/backup.cfg
```

Comware 7 extended Python API

The Comware 7 extended Python API is compatible with the Python syntax.

Importing and using the Comware 7 extended Python API

To use the Comware 7 extended Python API, you must import the API to Python.

Use either of the following methods to import and use the Comware 7 extended Python API:

- Use **import comware** to import the entire API and use **comware.***API* to execute an API.

  For example, to use the extended API **Transfer** to download file **test.cfg** from TFTP server 192.168.1.26:

  ```
  <Sysname> python
  Python 2.7.3 (default)
  [GCC 4.4.1] on linux2
  Type "help", "copyright", "credits" or "license" for more information.
  >>> import comware
  >>> comware.Transfer('tftp', '192.168.1.26', 'test.cfg', 'flash:/test.cfg', user='', password='')
  <comware.Transfer object at 0xb7eab0e0>
  ```

- Use **from comware import** *API* to import an API and use *API* to execute the API.

  For example, to use the extended API **Transfer** to download file **test.cfg** from TFTP server 192.168.1.26:

  ```
  <Sysname> python
  Python 2.7.3 (default)
  [GCC 4.4.1] on linux2
  Type "help", "copyright", "credits" or "license" for more information.
  >>> from comware import Transfer
  >>> Transfer('tftp', '192.168.1.26', 'test.cfg', 'flash:/test.cfg', user='', password='')
  <comware.Transfer object at 0xb7e5e0e0>
  ```

Comware 7 extended Python API functions

CLI class

CLI

Use **CLI** to execute Comware 7 CLI commands and create CLI objects.

Syntax

CLI(*command*='', *do_print*=True)

Parameters

command: Specifies the commands to be executed. To enter multiple commands, use a space and a semicolon (;) as the delimiter. To enter a command in a view other than user view, you must first enter

the commands used to enter the view. For example, you must enter **'system-view ;local-user test class manage'** to execute the **local-user test class manage** command.

do_print: Specifies whether to output the execution result:

- **True**—Outputs the execution result. This value is the default.
- **False**—Does not output the execution result.

Usage guidelines

This API supports only Comware commands. It does not support Linux, Python, or Tcl commands.

Returns

CLI objects

Examples

Add a local user named **test**.

```
<Sysname> python
Python 2.7.3 (default)
[GCC 4.4.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import comware
>>> comware.CLI('system-view ;local-user test class manage')
```

Sample output

```
<Sysname> system-view
System View: return to User View with Ctrl+Z.
[Sysname] local-user test class manage
New local user added.
<comware.CLI object at 0xb7f680a0>
```

get_output

Use **get_output** to get the output from executed commands.

Syntax

CLI.get_output()

Returns

Output from executed commands

Examples

Add a local user and get the output from the command.

```
<Sysname> python
Python 2.7.3 (default)
[GCC 4.4.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import comware
>>> c = comware.CLI('system-view ;local-user test class manage', False)
>>> c.get_output()
```

Sample output

```
['<Sysname>system-view', 'System View: return to User View with Ctrl+Z.',
'[Sysname]local-user test class manage', 'New local user added.']
```

Transfer class

Transfer

Use **Transfer** to download a file from a server.

Syntax

Transfer(*protocol*='', *host*='', *source*='', *dest*='', *vrf*='',*login_timeout*=10, *user*='', *password*='')

Parameters

protocol: Specifies the protocol used to download a file:

- **ftp**—Uses FTP.
- **tftp**—Uses TFTP.
- **http**—Uses HTTP.

host: Specifies the IP address of the remote server.

source: Specifies the name of the file to be downloaded from the remote server.

dest: Specifies a name for the downloaded file.

vrf: Specifies the MPLS L3VPN instance to which the remote server belongs. This argument represents the VPN instance name, a case-sensitive string of 1 to 31 characters. If the server belongs to the public network, do not specify this argument.

login_timeout: Specifies the timeout for the operation, in seconds. The default is 10.

user: Specifies the username for logging in to the server.

password: Specifies the login password.

Returns

Transfer object

Examples

Download file **test.cfg** from TFTP server 192.168.1.26.

```
<Sysname> python
Python 2.7.3 (default)
[GCC 4.4.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import comware
>>> comware.Transfer('tftp', '192.168.1.26', 'test.cfg', 'flash:/test.cfg', user='',
password='')
```

Sample output

```
<comware.Transfer object at 0xb7f700e0>
```

get_error

Use **get_error** to get the error information from the download operation.

Syntax

Transfer.get_error()

Returns

Error information (if there is no error information, **None** is returned)

Examples

Download file **test.cfg** from TFTP server 1.1.1.1 and get the error information from the operation.

```
<Sysname> python
Python 2.7.3 (default)
[GCC 4.4.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import comware
>>> c = comware.Transfer('tftp', '1.1.1.1', 'test.cfg', 'flash:/test.cfg', user='',
password='')
>>> c.get_error()
```

Sample output

```
“Timeout was reached”
```

API get_self_slot

get_self_slot

Use **get_self_slot** to get the member ID of the master device.

Syntax

get_self_slot()

Returns

A list object in the format of [-1,*slot-number*]. The *slot-number* indicates the member ID of the master device.

Examples

Get the member ID of the master device.

```
<Sysname> python
Python 2.7.3 (default)
[GCC 4.4.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import comware
>>> comware.get_self_slot()
```

Sample output

```
[-1,1]
```

API get_standby_slot

get_standby_slot

Use **get_standby_slot** to get the member IDs of the subordinate devices.

Syntax

get_standby_slot()

Returns

A list object in one of the following formats:

- [ ]—The IRF fabric does not have a subordinate device.
- [[-1,*slot-number*]]—The IRF fabric has only one subordinate device.
- [[-1,*slot-number1*],[-1,*slot-number2*],...]—The IRF fabric has multiple subordinate devices.

The *slot-number* arguments indicate the member IDs of the subordinate devices.

Examples

Get the member IDs of the subordinate devices.

```
<Sysname> python
Python 2.7.3 (default)
[GCC 4.4.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import comware
>>> comware.get_standby_slot()
```

Sample output

```
[[-1, 1], [-1, 2]]
```

API get_slot_range

get_slot_range

Use **get_slot_range** to get the supported IRF member ID range.

Syntax

get_slot_range()

Returns

A dictionary object in the format of {'MaxSlot': *max-slot-number*, 'MinSlot': *min-slot-number* }. The *max-slot-number* argument indicates the maximum member ID. The *min-slot-number* argument indicates the minimum member ID.

Examples

Get the supported IRF member ID range.

```
<Sysname> python
Python 2.7.3 (default)
[GCC 4.4.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import comware
>>> comware. get_slot_range()
```

Sample output

```
{'MaxSlot': 20, 'MinSlot': 1}
```

API get_slot_info

get_slot_info

Use **get_slot_info** to get information about a member device.

Syntax

get_slot_info()

Returns

A dictionary object in the format of {'Slot': *slot-number*, 'Status': '*status*', 'Chassis': *chassis-number*, 'Role': '*role*', 'Cpu': *CPU-number* }. The *slot-number* argument indicates the member ID of the device. The *status* argument indicates the status of the member device. The *chassis-number* and *CPU-number* arguments are fixed at 0. The *role* argument indicates the role of the member device.

Examples

Get information about a member device.

```
<Sysname> python
Python 2.7.3 (default)
[GCC 4.4.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import comware
>>> comware.get_slot_info(1)
```

Sample output

```
{'Slot': 1, 'Status': 'Normal', 'Chassis': 0, 'Role': 'Master', 'Cpu': 0}
```