Python Configuration Commands

Global Commands

description

Syntax **description** *description-string*
no description

Context config>python>python-policy
config>python>python-script
config>system>persistence>python-policy-cache

Description This command creates a text description stored in the configuration file for a configuration context.

The **description** command associates a text string with a configuration context to help identify the context in the configuration file.

The **no** form of this command removes any description string from the context.

Default No description is associated with the configuration context.

Parameters *description-string* — A text string describing the entity. Allowed values are any string up to 80 characters long composed of printable, 7-bit ASCII characters excluding double quotes. If the string contains special characters (#, $, spaces, etc.), the entire string must be enclosed within double quotes.

shutdown

Syntax [**no**] **shutdown**

Context config>python>python-policy>cache
config>python>python-script

Description Shutting down a Python script triggers the system to load and compile the script from the configured location(s). Since the system supports three locations, the primary, secondary and tertiary, the system will try to load the Python script in that order.

Shutting down a Python script will disable the Python script and cause the corresponding packet to pass through without any modification.

Default no shutdown

python

Syntax **python**

Context config

Description This command enables the context to configure Python parameters.

Python Policy Commands

python-policy

Syntax **python-policy** *name* [**create**]
no python-policy *name*

Context config>python

Description This command configures a Python policy which will select Python scripts to modify specific messages of different protocols.

The **no** form of the command removes the Python policy.

Parameters *name* — Specifies the Python policy name up to 32 characters in length.

create — This keyword is required when first creating the Python policy. Once the context is created, it is possible to navigate into the context without the **create** keyword.

cache

Syntax [**no**] **cache**

Context config>python>python-policy

Description This command enables the context to configure the limits of the caching API inside the Python scripts

entry-size

Syntax **entry-size** *size*
no entry-size

Context config>python>python-policy>cache

Description This command configures the maximum size of the data structure that can be stored in a single Python cache entry which includes both a value and key.

When requesting to store a data structure the size of the serialized object is compared with the value specified. If larger, the object will not be stored and Python will return exception.

The **no** form of the command reverts to the default value.

Default 256

Parameters *size* — Configures the maximum accepted size of a single cache entry.

Values 32 — 2048

max-entries

Syntax **max-entries** *count*
no max-entries

Context config>python>py-pol>cache

Description This command configures the maximum number of Python cache entries that can be stored in the cache of this Python policy.

If the limit has been reached, a Python exception will be thrown when requested to store another data structure.

Default 128000

Parameters *count* — Specifies the maximum number of cache entries allowed.

Values 1 — 1000000

max-entry-lifetime

Syntax **max-entry-lifetime** [**days** *days*] [**hrs** *hours*] [**min** *minutes*] [**sec** *seconds*]
no max-entry-lifetime

Context config>python>py-pol>cache

Description This command configures the maximum allowed lifetime for each entry of the Python cache of this Python policy.

When adding data to the Python cache the lifetime of the given object must always be specified. If the specified lifetime is bigger than the configured value, then the value of the **max-entry-lifetime** will be used instead of the lifetime that was specified.

The **no** form of the command reverts to the default.

Parameters **days** *days* **—** Specifies the maximum lifetime that can be set on a cache entry in days.

Values 0 — 7

Default 1

hrs *hours* **—** Specifies the maximum lifetime that can be set on a cache entry in hours.

Values 0 — 23

min *minutes* **—** Specifies the maximum lifetime that can be set on a cache entry in minutes.

Values 0 — 59

sec *seconds* **—** Specifies the maximum lifetime that can be set on a cache entry in seconds.

Values 0 — 59

mcs-peer

Syntax **mcs-peer** *ip-address* **sync-tag** [32 chars max]
no mcs-peer

Context config>python>py-pol>cache

Description This command specifies the MCS peer's address and sync-tag for syncing the cached entries of the python-policy. The sync-tag must be match on both chassis.

Default no mcs-peer

Parameters *ip-address* — Specifies the IPv4 address of the MCS peer.

sync-tag — Specifies the tag for sync up to 32 characters max.

minimum-lifetimes

Syntax **minimum-lifetimes**

Context config>python>py-pol>cache

Description This command enables the context to configure minimum-lifetime of python cache information.

Default none

high-availability

Syntax **high-availability** *seconds*
no high-availability

Context config>python>py-pol>cache>min-lifetimes

Description This command specifies the minimum lifetime of an entry that it could be synced across CPM.

Default no high-availability

Parameters *seconds* — Specifies the minimal lifetime in seconds.

Values 1 — 600

multi-chassis-redundancy

Syntax **multi-chassis-redundancy** *seconds*
no multi-chassis-redundancy

Context config>python>py-pol>cache>min-lifetimes

Description This command specifies the minimum lifetime for a cache entry to be synchronized with the MCS peer.

Default no multi-chassis-redundancy

Parameters *seconds* — Specifies the multi-chassis redundancy time in seconds.

Values 1 — 600

persistence

Syntax **persistence** *seconds*
no persistence

Context config>python>py-pol>cache>min-lifetimes

Description This command configures the minimum lifetime for a cache entry to be made persistent.

Default no persistence

Parameters *persistence* — The minimum lifetime in seconds.

Values 1 — 600

persistence

Syntax [**no**] **persistence**

Context config>python>py-pol>cache

Description This command enables persistency support for the cached entries of the python-policy.

Default no persistence

dhcp

Syntax **dhcp** *type* **direction** {**ingress | egress**} **script** *script*
no dhcp *type* **direction** {**ingress | egress**}

Context config>python>py-policy

Description This command specifies the Python script for the specified DHCPv4 packet type in the specified direction.

Multiple **dhcp** command configurations are allowed in the same Python policy.

The **no** form of the command reverts to the default.

Default none

Parameters *type* — Specifies the message type of the event.

Values discover, offer, request, decline, ack, nak, release, inform, force-renew, lease-query, lease-unassigned, lease-unknown, lease-active

direction {**ingress** | **egress**} **—** specifies whether the packetis being received by the system or or being sent by the system.

script *script* **—** Specifies the name of the Python script up to 32 characters in length, that will be used to handle the specified message.

dhcp6

Syntax **dhcp6** *type* **direction** {**ingress | egress**} **script** *script*
no dhcp6 *type* **direction** {**ingress | egress**}

Context config>python>py-policy

Description This command specifies the Python script for the specified DHCPv6 packet type in the specified direction.

Multiple **dhcps** command configurations are allowed in the same Python policy.

Default none

Parameters *type* — Specifies the message type of the event.

Values solicit, advertise, request, confirm, renew, rebind, reply, release, decline, reconfigure, info-request, relay-forward, relay-reply

direction {**ingress** | **egress**} **—** specifies whether the event is incoming or outgoing.

script *script* **—** Specifies the name of the Python script up to 32 characters in length, that will be used to handle the specified message.

diameter

Syntax **diameter** *type* **direction** {**ingress|egress**} **script** *script*
no diameter *type* **direction** {**ingress|egress**}

Context config>python>py-policy

Description This command specifies the Python script to use for the specified Diameter message type in the specified direction.

Multiple diameter command configurations are allowed in the same Python policy.

Default none

Parameters *type* — Specifies the message type.

Message type	**Application**	**Direction**
aaa – AA Answer	Nasreq	ingress
aar – AA Request	Nasreq	egress
asa – Abort Session Answer	Gx, Gy	egress
asr – Abort Session Request	Gx, Gy	ingress
cca – Credit Control Answer	Gx, Gy	ingress
ccr – Credit Control Request	Gx, Gy	egress
cea – Capabilities Exchange Answer	Base	ingress
cer – Capabilities Exchange Request	Base	egress
dpa – Disconnect Peer Answer	Base	ingress / egress
dpr – Disconnect Peer Request	Base	ingress / egress

dwa – Device Watchdog Answer	Base	ingress / egress
dwr – Device Watchdog Request	Base	ingress / egress
raa – Re-Authentication Answer	Gx, Gy	egress
rar – Re-Authentication Request	Gx, Gy	ingress

direction {**ingress**|**egress**} **—** Specifies if the message is incoming or outgoing.

script *script* **—** Specifies the name of the Python script up to 32 characters in length, that will be used to handle the specified message.

radius

Syntax **radius** *type* **direction** {**ingress | egress**} **script** *script*
no radius *type* **direction** {**ingress | egress**}

Context config>python>py-policy

Description This command specifies the Python script for the specified RADIUS packet type in the specified direction.

Multiple **radius** command configurations are allowed in the same Python policy.

Parameters *type* — Specifies the message type of the event.

access-request, access-accept, access-reject, accounting-request, accounting-response, access-challenge, disconnect-request, change-of-authorization-request

direction {**ingress** | **egress**} **—** specifies whether the event is incoming or outgoing.

script *script* **—** Specifies the name of the Python script up to 32 characters in length, that will be used to handle the specified message.

python-script

Syntax **python-script** *name* [**create**]
no python-script *name*

Context config>python

Description This command enables the context to configure Python scripts to modify messages of different protocols.

The **no** form of the command removes the Python script name from the configuration.

Parameters **name —** Specifies the name of this Python script policy.

create — This keyword is required when first creating the Python script. Once the context is created, it is possible to navigate into the context without the **create** keyword.

action-on-fail

Syntax **action-on-fail** {**drop | passthrough**}

no action-on-fail

Context config>python>python-script

Description This command specifies the action taken when Python fails to modify the given message.

The **no** form of the command reverts to the default.

Default drop

Parameters **drop —** Specifies that the packet will be dropped.

passthrough — Specifies that he packet will be sent out without any modifications.

primary-url

Syntax **primary-url** *url*
no primary-url

Context config>python>python-script

Description This command specifies the location of the primary Python script. The system supports three locations for each Python script. Users can store the script file on either a local CF card or an FTP server.

The **no** form of the command removes the URL.

Default no primary-url

Parameters **url —** Specifes the primary URL of the Python script up to 180 characters in length, either a local CF card url or a FTP server URL.

protection

Syntax **protection none**
protection hmac-sha256 key *key* [**hash|hash2**]
no protection

Context config>python>python-script

Description This command specifies the format of the Python script file(s) in this python-script. Unintentional changing of Python script file could be prevented by using protected format.

The **no** form of this command equals to **protection none**.

Default none

Parameters **none —** Indicates the Python script is stored in plain-text, without any mechanism in place to ensure the integrity nor the confidentiality of the content of the Python script.

hmac-sha256 — Indicates the first line of the Python script must consist of the hash value obtained by hashing the rest of the Python script using the hmac-sha256 hashing algorithm given the key specified in tmnxPythonScriptProtectionKey.

key *key* — The specified key along with original Python script file content are used to compute the hash. The computed hash will be compared to the hash in the Python script file. If there is no match, then system will fail to load the script.

hash — Specifies the key is entered in an encrypted form. If the hash parameter is not used, the key is assumed to be in a non-encrypted, clear text form. For security, all keys are stored in encrypted form in the configuration file with the hash parameter specified.

hash2 — Specifies the key is entered in a more complex encrypted form. If the hash2 parameter is not used, the less encrypted hash form is assumed.

secondary-url

Syntax **secondary-url** *url*
no secondary-url

Context config>python>python-script

Description This command specifies the location of secondary Python script. The system supports three locations for each Python-script. Users can store scripts file on either a local CF card or a FTP server.

The **no** form of the command removes the URL.

Default no secondary-url

Parameters **url** — Specifes the secondary URL of the Python script up to 180 characters in length, either a local CF card url or a FTP server URL.

tertiary-url

Syntax **tertiary-url** *url*
no tertiary-url

Context config>python>python-script

Description This command specifies the location of tertiary Python script. The system supports three locations for each Python-script. Users can store scripts file on either a local CF card or a FTP server.

The **no** form of the command removes the URL.

Default no tertiary-url

Parameters **url** — Specifes the tertiary URL of the Python script up to 180 characters in length, either a local CF card url or a FTP server URL.

dhcp-python-policy

Syntax **dhcp-python-policy** *policy-name*
no dhcp-python-policy

Context config>service>vpls>sap

Description This command specified the Python policy for DHCPv4 packets sent/received on the capture SAP.

Default none

Parameters *policy name* — Specifies an existing Python policy name.

dhcp6-python-policy

Syntax **dhcp6-python-policy** *policy-name*
no dhcp6-python-policy

Description This command specified the Python policy for DHCPv6 packets sent/received on the capture SAP.

Default none

Parameters *policy name* — Specifies an existing Python policy name.

python-policy

Syntax **python-policy** *name*
no python-policy

Context config>service>vprn>sub-if>grp-if>ipv4>dhcp4
config>service>ies>sub-if>grp-if>ipv4>dhcp4

Description This command specified the python-policy for DHCPv4 packets sent/received on the group interface.

The **no** form of the command removes the policy name from the configuration.

Default none

Parameters *name* — Specifies an existing Python policy name.

python-policy

Syntax **python-policy** *name*
no python-policy

Context config>service>vprn>sub-if>grp-if>ipv6>dhcp6
config>service>ies>sub-if>grp-if>ipv6>dhcp6

Description This command specified the python-policy for DHCPv6 packets sent/received on the group interface.

The **no** form of the command removes the policy name from the configuration.

Default none

Parameters *name* — Specifies an existing Python policy name.

python-policy

Syntax **python-policy** *name*
no python-policy

Context config>service>vprn>sub-if>dhcp
config>service>ies>sub-if>dhcp

Description This command specified the Python policy for DHCPv4 packets sent/received on the retail subscriber interface.

Default none

Parameters *name* — Specifies the name of the Python policy.

python-policy

Syntax **python-policy** *name*
no python-policy

Context config>aaa>radius-srv-plcy

Description This command specified the python-policy for RADIUS packets to/from the RADIUS servers defined in the specified radius-server-policy.

Default none

Parameters *name* — Specifies the name of the Python policy.

python-policy

Syntax **python-policy** *name*
no python-policy

Context config>service>vprn>radius-proxy>server
config>router>radius-proxy>server

Description This command specified the python-policy for RADIUS packets sent/received on the client side of the RADIUS proxy server.

Default none

Parameters *name* — Specifies the name of the Python policy.

persistence

Syntax [**no**] **persistence**

Context config>system

Description This command enables the context to configure persistence parameters on the system.

The persistence feature enables state on information learned through DHCP snooping across reboots to be retained. This information includes data such as the IP address and MAC binding information, lease-length information, and ingress sap information (required for VPLS snooping to identify the ingress interface).

If persistence is enabled when there are no DHCP relay or snooping commands enabled, it will simply create an empty file.

Default no persistence

python-policy-cache

Syntax **python-policy-cache**

Context config>system>persistence

Description This command configures Python policy cache persistency parameters.

python

Syntax **python**

Context config>redundancy>multi-chassis>peer>sync

Description This command enables syncing of python-policy cached entries to the peer.

Use the **mcs-peer** command in the python-policy to enable syncing for a specific python-policy.

Default no python

location

Syntax **location** *cflash-id*
no location

Context config>system>persistence>persistence

Description This command instructs the system where to write the file. The name of the file is: dhcp-persistence.db. On boot the system scans the file systems looking for dhcp-persistence.db, if it finds it starts to load it.

In the subscriber management context, the location specifies the flash device on a CPMCFM card where the data for handling subscriber management persistency is stored.

The **no** form of this command returns the system to the default. If there is a change in file location while persistence is running, a new file will be written on the new flash, and then the old file will be removed.

Default no location

Tools Commands

python-policy

Syntax **python-policy** *name* **cache**
python-policy *name* **cache hex-key** *hex-string*
python-policy *name* **cache** *string-key* [512 chars max]

Context tools>dump

Description This command dumps all cached entries or a specified entry of a specified Python policy.

The DDP key in the output is the python cache persistency record key.

Note that the DDP Key in the output could be used for the tools>**dump>persistence>python** command.

Parameters *name* — Specifies the name of the Python policy.

string-key — Specifies the key of the entry to be updated in ASCII strong format.

hex-key — Specifies the key of the entry to be updated in hex string format.

python-policy

Syntax **python-policy** *name* **cache hex-key** *hex-string* **set-lifetime** [0..2147483647]
python-policy *name* **cache** *string-key* [512 chars max] **set-lifetime** [0..2147483647]

Context tools>perform

Description This command set the lifetime of a specified python cache entry.

Parameters *name* — Specifies the name of the Python policy.

string-key — Specifies the key of the entry to be updated in ASCII strong format.

hex-key *hex-string* **—** Specifies the key of the entry to be updated in hex string format.

set-lifetime — Specifies the new lifetime of the entry.

python-script protect

Syntax **python-script protect input** *file-url* **hmac-sha256 key** *secret-key* **output** *file-url*

Context tools>perform

Description This command converts a normal (unprotected) Python script file into an SRPY format with specified key.

Parameters **input** *file-url* **—** Specifies the URL of the input script file.

key *secret-key* **—** Specifies the key used to compute the hash

output *file-url* **—** Specifies the URL of the output script file

python-script reload

Syntax **python-script reload** *name*

Context tools>perform

Description This command will try to reload/recompile the primary/secondary/tertiary scripts in the specified Python script in order. The system will use the first script that comes up.

Parameters *name* — Specifies the name of the python-script to be reloaded.

Show Commands

python

Syntax **python**

Context show

Description This command enables the context to display Python information.

python-policy

Syntax **python-policy**
python-policy *policy-name* [**association**]]|

Context show>python

Description This command displays information about the currently configured Python policy.

The system will display a list of currently configured Python policy names if no parameter is specified.

Parameters *policy-name* — Specifies the Python policy name to display.

association — Displays the associations of the specified Python policy.

Sample Output

```
show python python-policy "dhcp"
===============================================================================
Python policy "dhcp"
===============================================================================
Description   : (Not Specified)
-------------------------------------------------------------------------------
Messages
-------------------------------------------------------------------------------
Type                                   Dir     Script
-------------------------------------------------------------------------------
dhcpDiscover                           egress  dhcpv4
dhcpRequest                            egress  dhcpv4
dhcpAck                                ingress dtc
-------------------------------------------------------------------------------
No. of Messages: 3
-------------------------------------------------------------------------------
===============================================================================
```

```
show python python-policy "dhcp" association
===============================================================================
Python Policy Association
===============================================================================
Location
-------------------------------------------------------------------------------
Service: 500, GrpIf g1, dhcp
```

```
-------------------------------------------------------------------------------
No. of Python policy association: 1
===============================================================================
```

python-script

Syntax **python-script**
python-script *script-name* [**association | source-in-use**

Context show>python

Description This command displays information about the currently configured Python script.

The system will display a list of currently configured Python script names if no parameter is specified.

Parameters *script-name* — Specifies the Python script name to display information.

association — Displays the associations of the specified Python script.

source-in-use — Displays the Python source code in use.

Sample Output

```
show python python-script "dhcpv4"
===============================================================================
Python script "dhcpv4"
===============================================================================
Description   : (Not Specified)
Admin state   : inService
Oper state    : inService
Action on fail: drop
Protection    : none
Primary URL   : cf1:/dhcpv4.py
Secondary URL : (Not Specified)
Tertiary URL  : (Not Specified)
Active URL    : primary
Last changed  : 01/26/2014 05:02:10
===============================================================================


show python python-script "dhcpv4" association
===============================================================================
Python Script Association
===============================================================================
Policy                          Type                                    Dir
-------------------------------------------------------------------------------
dhcp                            dhcpDiscover                            egress
dhcp                            dhcpRequest                             egress
-------------------------------------------------------------------------------
No. of Python script association: 2
===============================================================================
```

Debug Commands

python-script

Syntax **python-script** *script-name*

Context debug>python

Description This command enters the debug context the specified Python script

Parameters *policy-name* — Specifies the Python script name.

script-all-info

Syntax **script-all-info**

Context debug>python>py-script

Description This command enables the script-compile-error, script-export-variables, script-output, script-outputon-error, and script-runtime-error functionalities.

script-compile-error

Syntax [**no**] **script-compile-error**

Context debug>python>py-script

Description This command sends the traceback of the compile error to the logger. The traceback contains detailed information about where and why the compilation fails. The compilation takes place when the CLI user changes the admin state of the Python script from shutdown to no-shutdown.

script-export-variables

Syntax [**no**] **script-export-variables**

Context debug>python>py-script

Description This command sends the output variables of the Python script to the logger when the script ran successfully.

script-output

Syntax [**no**] **script-output**

Context debug>python>py-script

Description This command sends the output (such as from 'print' statements) of the Python script to the logger.

script-output-on-error

Syntax [**no**] **script-output-on-error**

Context debug>python>py-script

Description This command sends the output (such as traceback data) of the Python script to the logger, but only when the script fails.

script-output

Syntax [**no**] **script-output**

Context debug>python>py-script

Description This command sends the traceback of the Python script failure to the logger.

Clear Commands

python-policy

Syntax
python-policy *name* **cache**
python-policy *name* **cache hex-key** *hex-string*
python-policy *name* **cache** *string-key* [512 chars max]

Context clear>python

Description This command clears Python policy data.

Parameters *name* — Specifies the name of the Python policy.

string-key — Specifies the key of the entry to be updated in ASCII strong format.

hex-key — Specifies the key of the entry to be updated in hex string format.