



## Mobile Policy PCF Commands

- [advance-tuning](#), on page 1
- [advance-tuning async-threading](#), on page 2
- [advance-tuning async-threading threading-config](#), on page 3
- [advance-tuning brute-force-recovery](#), on page 4
- [advance-tuning http2-threading](#), on page 5
- [advance-tuning overload-control](#), on page 6
- [advance-tuning overload-control diameter global](#), on page 6
- [advance-tuning overload-control diameter global action](#), on page 6
- [advance-tuning overload-control diameter global limits](#), on page 7
- [advance-tuning overload-control rest global](#), on page 7
- [advance-tuning overload-control rest global action](#), on page 7
- [advance-tuning overload-control rest global limits](#), on page 8
- [rest-endpoint](#), on page 8
- [rest-endpoint certificate-status](#), on page 10
- [rest-endpoint discovered-profiles](#), on page 10
- [rest-endpoint discovered-profiles chf](#), on page 10
- [rest-endpoint discovered-profiles udr](#), on page 11
- [rest-endpoint interface](#), on page 12
- [rest-endpoint peer-status](#), on page 13
- [rest-endpoint registration-status](#), on page 13
- [traffic engine](#), on page 14
- [traffic engine rule](#), on page 14

### advance-tuning

Configures advanced tuning parameters.

**Command Modes** Exec > Global Configuration

**Syntax Description** `advance-tuning { n7-stale-session-error-codes error_codes | redis-password redis_password}`

**n7-stale-session-error-codes** *error\_codes*

Specify the comma-separated list of N7Notify stale session error codes.

Must be a string.

**redis-password** *redis\_password*

Specify the Redis password.

**Usage Guidelines**

Use this command to configure advanced tuning parameters.

## advance-tuning async-threading

Configures threading configuration for HTTP outgoing request from PCF.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

```
advance-tuning async-threading { default-drop-oldest-when-full { false | true } | default-priority default_priority | default-processing-threads processing_thread_number | default-queue-size default_queue_size | default-worker-threads default_number_worker_threads | request-timeout-ms request_timeout | thread-configuration service_name drop-oldest-when-full { false | true } | priority thread_priority | queue-size queue_size | threads thread_number}
```

**default-drop-oldest-when-full** { **false** | **true**}

Specify to drop the oldest packet when queue is full.

Must be one of the following:

- **false**
- **true**

Default Value: false.

**default-priority** *default\_priority*

Specify the default priority of thread.

Must be an integer.

Default Value: 5.

**default-processing-threads** *processing\_thread\_number*

Specify the default number of processing threads.

Must be an integer.

Default Value: 10.

**default-queue-size *default\_queue\_size***

Specify the default size of the queue.

Must be an integer.

Default Value: 100.

**default-worker-threads *default\_number\_worker\_threads***

Specify the default number of worker threads.

Must be an integer.

Default Value: 20.

**http2-connect-timeout-ms *http2\_connect\_timeout***

Specify the request timeout period in milliseconds.

Must be an integer.

Default Value: 100.

**http2-idle-connection-timeout-sec *http2\_client\_idle\_connect\_timeout***

Specify the idle connection timeout for HTTP2 client.

Must be an integer.

Default Value: 60.

**max-timeouts-to-reconnect *max\_requests\_timeouts***

Specify the maximum request timeouts to reconnect HTTP2 connection.

Must be an integer.

Default Value: 0.

**Usage Guidelines**

Use this command to configure threading configuration for HTTP outgoing request from PCF.

**Example**

The following command configures the threading configuration for HTTP outgoing request from PCF with default priority of 5:

```
advance-tuning async-threading default-priority 5
```

# advance-tuning async-threading threading-config

Configures threading configuration of servicethreading.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

**threading-config** *service\_name*

**drop-oldest-when-full { false | true}**

Specify to drop the oldest packet when the queue is full.

Must be one of the following:

- false
- true

**priority *thread\_priority***

Specify the threading priority.

Must be an integer.

**queue-size *queue\_size***

Specify the size of the queue.

Must be an integer.

**service-name *service\_name***

Specify the service name.

Must be a string.

**threads *thread\_number***

Specify the number of threads.

Must be an integer.

**Usage Guidelines**

Use this command to configure threading configuration of service.

## advance-tuning brute-force-recovery

Enables HTTP2 connection recovery parameters via closing connection.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

```
brute-force-recovery { max-reconnects max_http2_reconnects | time-interval-mins
time_interval}
```

**max-reconnects *max\_http2\_reconnects***

Specify the maximum number of HTTP2 reconnect attempts to be allowed before restarting REST endpoint for recovery. Counter for previous reconnect attempts gets reset as per time-interval-mins. When set to 0, restart of rest endpoint is not performed even on reaching configured count and system continues attempting reconnect for recovery.

Must be an integer.

Default Value: 0.

**time-interval-mins** *time\_interval*

Specify the time interval in minutes. If there is no HTTP2 reconnect triggered due to timeout for specified duration plus 1 minute then counter for previous reconnect attempts is reset to 0.

Must be an integer.

Default Value: 0.

**Usage Guidelines**

Use this command to enable HTTP2 connection recovery parameters via closing connection.

## advance-tuning http2-threading

Configures threading configuration for HTTP incoming request to PCF.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

```
advance-tuning http2-threading { disable-validation { false | true } |
idle-thread-timeout-ms idle_thread_timeout | max-queue-capacity max_packet_capacity
| max-thread-pool-size max_thread_pool_size | min-thread-pool-size
min_thread_pool_size | request-timeout-ms request_timeout}
```

**disable-validation** { **false** | **true**}

Specify to enable or disable validation.

Must be one of the following:

- **false**
- **true**

Default Value: false.

**idle-thread-timeout-ms** *idle\_thread\_timeout*

Specify the thread idle timeout in milliseconds.

Must be an integer.

Default Value: 60000.

**max-queue-capacity** *max\_packet\_capacity*

Specify the maximum packet capacity of the queue.

Must be an integer.

Default Value: 5000.

**max-thread-pool-size** *max\_thread\_pool\_size*

Specify the maximum size of pool of threads.

Must be an integer.

Default Value: 20.

**min-thread-pool-size** *min\_thread\_pool\_size*

Specify the minimum size of pool of threads.

Must be an integer.

Default Value: 5.

**Usage Guidelines**

Use this command to configure threading configuration for HTTP incoming request to PCF.

**Example**

The following command configures the threading configuration for HTTP incoming request to PCF with maximum queue capacity of 500:

```
advance-tuning http2-threading max-queue-capacity 500
```

## advance-tuning overload-control

Configures threading configuration for HTTP incoming request to PCF.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

**advance-tuning overload-control**

**Usage Guidelines**

Use this command to configure threading configuration for HTTP incoming request to PCF.

## advance-tuning overload-control diameter global

Configures Overload Control configuration for all Diameter interfaces.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

**advance-tuning overload-control diameter global**

**Usage Guidelines**

Use this command to configure Overload Control configuration for all Diameter traffic.

## advance-tuning overload-control diameter global action

Configures the action to take on overload detection.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

**action throttle-action** *throttle\_action*

**action throttle-action** *throttle\_action*

Specify the action to take when overload traffic is detected.

Must be one of the following:

- **DROP**
- **REJECT**

**Usage Guidelines** Use this command to configure the action that must be taken on overload detection.

## advance-tuning overload-control diameter global limits

Configures the limits for the overload handling parameters for the REST or Diameter interface.

**Command Modes** Exec > Global Configuration

**Syntax Description** **limits max-requests-per-sec** *maximum\_requests*

**max-requests-per-sec** *maximum\_requests*

Specify the maximum number of requests that are allowed per second before throttling is applied.

Must be an integer.

**Usage Guidelines** Use this command to configure the limits for overload handling parameters for the REST or Diameter interface.

## advance-tuning overload-control rest global

Configures Overload Control configuration for all REST traffic.

**Command Modes** Exec > Global Configuration

**Syntax Description** **advance-tuning overload-control rest global**

**Usage Guidelines** Use this command to configure threading configuration for HTTP incoming request to PCF.

## advance-tuning overload-control rest global action

Configures the action to take on overload detection.

**Command Modes** Exec > Global Configuration

**Syntax Description** **action throttle-action** *throttle\_action*

**action throttle-action** *throttle\_action*

Specify the action to take when overload traffic is detected.

Must be one of the following:

- **DROP**

- REJECT

**Usage Guidelines** Use this command to configure the action that must be taken on overload detection.

## advance-tuning overload-control rest global limits

Configures the limits for the overload handling parameters for the REST or Diameter interface.

**Command Modes** Exec > Global Configuration

**Syntax Description** **limits max-requests-per-sec** *maximum\_requests*

**max-requests-per-sec** *maximum\_requests*

Specify the maximum number of requests that are allowed per second before throttling is applied.

Must be an integer.

**Usage Guidelines** Use this command to configure the limits for overload handling parameters for the REST or Diameter interface.

## rest-endpoint

Configures the REST endpoint.

**Command Modes** Exec > Global Configuration

**Syntax Description** **rest-endpoint** { **ips** *ip\_address* | **port** *port\_number* | **replicas** *replica\_count* | **tracing-service-name** *service\_name*}

**certificate-name** *certificate\_name*

Specify the alias name for the certificate.

**http-connection-limit** *max\_inbound\_https\_connections*

Specify the maximum number of allowed inbound HTTPS connections.

Must be an integer.

Default Value: 200.

**http-idle-connection-timeout-on-server-seconds** *connection\_timeout*

Specify the server side idle connection timeout period in seconds.

Must be an integer.

Default Value: 60.

**inbound-request-timeout-ms** *inbound\_requests\_timeout*

Specify the timeout period for inbound requests in milliseconds.



Must be an integer.

Default Value: 2000.

**ips *ip\_address***

Specify the IP addresses for the REST service.

**outbound-request-timeout-ms *outbound\_requests\_timeout***

Specify the timeout period for outbound requests in milliseconds.

Must be an integer.

Default Value: 200.

**port *port\_number***

Specify port number of the REST service.

Must be an integer.

**replicas *replica\_count***

Specify the replica count.

Must be an integer.

Default Value: 1.

**repository *repository\_name***

Specify to override the Helm repository.

**tracing-service-name *tracing\_service\_name***

Specify the tracing service name for REST endpoint.

Must be a string.

**uri-scheme *uri\_scheme***

Specify the URI scheme.

Must be one of the following:

- **http**
- **https**

Default Value: http.

---

**Usage Guidelines**

Use this command to configure the REST endpoint.

**Example**

The following command configures the REST endpoint tracing service name as nPcf-pcf-rest-ep:

```
rest-endpoint tracing-service-name nPcf-pcf-rest-ep
```

## rest-endpoint certificate-status

Displays certificate status.

**Command Modes** Exec > Global Configuration

**Syntax Description** `show certificate-status`

**certificateName**

Displays the certificate name.

Must be a string.

**timeToExpire**

Displays the time to expire.

Must be a string.

**Usage Guidelines** Use this command to view certificate status.

## rest-endpoint discovered-profiles

Displays the PCF discovered and cached NFs.

**Command Modes** Exec > Global Configuration

**Syntax Description** `show discovered-profiles`

**Usage Guidelines** Use this command to view the PCF discovered and cached NFs.

## rest-endpoint discovered-profiles chf

Displays PCF Discover cache for CHF.

**Command Modes** Exec > Global Configuration

**Syntax Description** `show discovered-profiles chf`

**expiry**

The expiration time in seconds.

Must be a string.

**ipv4Address**

The NF IPv4 address.

Must be a string.

**nfnstanceld**

The NF Instance ID.

Must be a string.

**nfStatus**

Is the PCF instance registered to NRF.

Must be a string.

**nfType**

The NF type.

Must be a string.

**port**

The NF port number.

**Usage Guidelines**

Use this command to view PCF Discover cache for CHF.

## rest-endpoint discovered-profiles udr

Displays PCF Discover cache for UDR.

**Command Modes**

Exec > Global Configuration

**Syntax Description**

**show discovered-profiles udr**

**expiry**

The expiration time in seconds.

Must be a string.

**ipv4Address**

The NF IPv4 address.

Must be a string.

**nfnstanceld**

The NF Instance ID.

Must be a string.

**nfStatus**

Is the PCF instance registered to NRF.

Must be a string.

**nfType**

The NF type.

Must be a string.

**port**

The NF port number.

---

**Usage Guidelines** Use this command to view PCF Discover cache for UDR.

## rest-endpoint interface

Configures the NF interfaces.

---

**Command Modes** Exec > Global Configuration

---

**Syntax Description** `rest-endpoint interface interface_type { ip ip_address | notify-update-retry-count notify_update_retry_count | outbound-request-timeout-ms outbound_request_timeout | port port_number}`

**interface interface\_type**

Specify the NF interface type.

**ip ip\_address**

Specify the NF interface IP address.

**notify-update-retry-count notify\_update\_retry\_count**

Specify the retry count for N7 NotifyUpdate outbound requests in case of timeout.

Must be an integer.

Default Value: 0.

**outbound-request-timeout-ms outbound\_request\_timeout**

Specify the timeout period for outbound requests in milliseconds.

Must be an integer.

**port port\_number**

Specify the port number for NF interface.

Must be an integer.

**Usage Guidelines** Use this command to configure NF interfaces.

### Example

The following command configures the NF interface to restEndpoint with the IP address 11.11.11.11:

```
rest-endpoint interface restEndpoint ip 11.11.11.11
```

## rest-endpoint peer-status

Displays the NF's peer status.

**Command Modes** Exec > Global Configuration

**Syntax Description** `show peer-status`

### connectionDuration

Displays the NF connection uptime duration.

Must be a string.

### nfName

Displays the PCF Discover cache for UDR.

Must be a string.

### peerIp

Displays the NF peer IP address.

Must be a string.

### peerPort

Displays the NF peer port number.

Must be a string.

### podIp

Displays the PCF pod IP address.

Must be a string.

**Usage Guidelines** Use this command to view the NF's peer status.

## rest-endpoint registration-status

Displays NRF registration details.

<b>Command Modes</b>	Exec > Global Configuration
<b>Syntax Description</b>	<p><b>show registration-status</b></p> <p><b>nrfUri</b> Displays the registered NRF URI. Must be a string.</p> <p><b>podId</b> Displays the PCF pod ID. Must be a string.</p> <p><b>registered</b> Displays whether Is PCF instance is registered to NRF. Must be a string.</p>
<b>Usage Guidelines</b>	Use this command to view NRF registration details.

## traffic engine

Configures the default engine group to receive traffic.

<b>Command Modes</b>	Exec > Global Configuration
<b>Syntax Description</b>	<p><b>traffic engine default-destination</b> <i>default_destination</i></p> <p><b>default-destination</b> <i>default_destination</i></p> <p>Specify the default engine group to receive the traffic.</p>
<b>Usage Guidelines</b>	Use this command to configure the default engine group to receive traffic. Note that a similar command is available for PCRF.

### Example

The following command configures the default engine group as /policy-test:engine/test-test1:group:

```
traffic engine default-destination /policy-test:engine/test-test1:group
```

## traffic engine rule

Configures traffic routing rule parameters.

<b>Command Modes</b>	Exec > Global Configuration
----------------------	-----------------------------

---

**Syntax Description**

```
traffic engine rule rule_name { dnn dnn_route | supi supi_route | gpsi gpsi_route  
| hash-prefix hash_prefix_route | intf interface_route | destination engine_group}
```

**destination *engine\_group***

Specify the engine group to receive traffic.

**dnn *dnn\_route***

Specify the route on DNN - regex.

Must be a string.

**gpsi *gpsi\_route***

Specify the route on GPSI - regex.

Must be a string.

**hash-prefix *hash\_prefix\_route***

Specify the route on 2-digit hash - prefix.

Must be a string.

**intf *interface\_route***

Specify the route on interface type - exact.

Must be a string.

**supi *supi\_route***

Specify the route on SUPI - regex.

Must be a string.

***rule\_name***

Specify the rule name.

Must be a string.

---

**Usage Guidelines**

Use this command to configure the traffic routing rule parameters.

