# Ultra Cloud Core 5G Session Management Function, Release 2020.02 - Statistics Reference

**First Published:** 2020-04-30

**Last Modified:** 2020-05-29

**Americas Headquarters**
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

# CONTENTS

# About this Guide

This preface describes the *5G Session Management Function Guide*, how it is organized and its document conventions.

This guide describes the Cisco Session Management Function (SMF) and includes infrastructure and interfaces, feature descriptions, specification compliance, session flows, configuration instructions, and CLI commands for monitoring and troubleshooting the system.

• Conventions Used, on page v

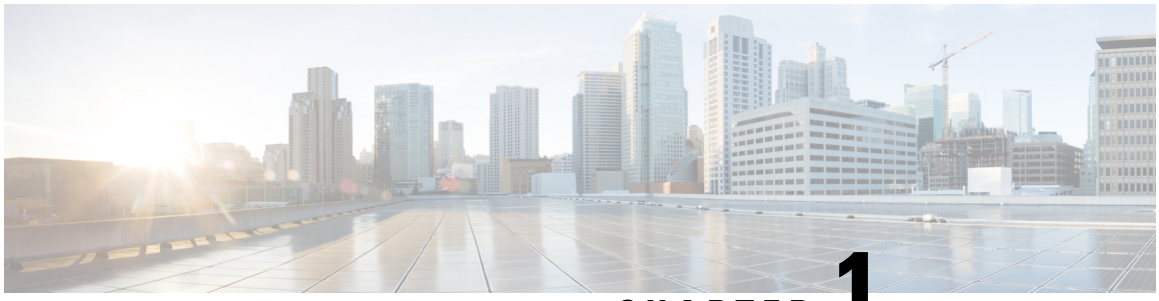
# Conventions Used

The following tables describe the conventions used throughout this documentation.

| Notice Type | Description |
|---|---|
| Information Note | Provides information about important features or instructions. |
| Caution | Alerts you of potential damage to a program, device, or system. |
| Warning | Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards. |

| Typeface Conventions | Description |
|---|---|
| Text represented as a `screen display` | This typeface represents displays that appear on your terminal screen, for example:<br>`Login:` |
| Text represented as **commands** | This typeface represents commands that you enter, for example:<br>**show ip access-list**<br>This document always gives the full form of a command in lowercase letters. Commands are not case sensitive. |

| Typeface Conventions | Description |
|---|---|
| Text represented as a **command** *variable* | This typeface represents a variable that is part of a command, for example:<br>**show card** *slot_number*<br>*slot_number* is a variable representing the desired chassis slot number. |
| Text represented as menu or sub-menu names | This typeface represents menus and sub-menus that you access within a software application, for example:<br>Click the **File** menu, then click **New** |



CHAPTER 1

# SMF Interface for Metrics



## Feature Summary and Revision History

### Feature Summary

| | |
|---|---|
| Applicable Product(s) or Functional Area | 5G-SMF |
| Applicable Platform(s) | SMI |
| Default Setting | Enabled – Always-on |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

### Revision History

| Revision Details | Release |
|---|---|
| First introduced. | 2020.02.0 |

## Feature Description

SMF uses Prometheus for gathering statistics/counters from its microservices.

Grafana is used as the user interface to view metrics. It pulls the data from the Prometheus data store. Default graphs for KPI are available using Grafana for rendering a graphical view of the statistics with timelines.

For each microservice, counters and a set of labels are defined. Counters are incremented/decremented with the set of labels depending on the functionality.

The following snapshot is a sample of the Grafana dashboard.

*Figure 1: Grafana Dashboard*


# SMF Rest EP Microservice

This section describes the supported counters and set of labels for the SMF Rest EP microservice.

## Counters

The SMF REST EP microservice includes the following counters:

*Table 1: SMF REST EP Microservice Counters*

| Number | Metric | Description |
|---|---|---|
| 1 | smf_restep_http_msg_total | This counter is incremented with every HTTP message received/sent at rest-ep microservice. |
| 2 | smf_restep_http_msg | This counter is incremented with every HTTP message received/sent at rest-ep microservice along with the time taken to serve the message. |

## Labels

The SMF REST EP microservice includes the following labels for the counters:

*Table 2: SMF REST EP Microservice Labels for Counters*

| Number | Label | Description |
|---|---|---|
| 1 | NF TYPE | This label can be any 5G Node that interacts with SMF. For example: AMF, PCF, NRF |

| Number | Label | Description |
|---|---|---|
| 2 | MESSAGE DIRECTION | Displays the direction of the HTTP message with respect to the REST EP microservice. The possible values are:<br>“inbound”<br>“outbound” |
| 3 | API NAME | Displays the service name being served. It can be:<br>"register_ue"<br>"deregister_ue"<br>"subscription_req"<br>"nf_registration"<br>"nf_discovery"<br>"slice_selection"<br>"amf_create_sm_context"<br>"amf_update_sm_context"<br>"amf_release_sm_context"<br>"amf_n1_n2_transfer"<br>"pcf_sm_policy_control_create"<br>"pcf_sm_policy_control_update"<br>"pcf_sm_policy_control_delete"<br>"pcf_sm_policy_control_update_notify"<br>"pcf_sm_policy_control_terminate_notify" |
| 4 | NF URI | Displays the rest-ep URI used in the HTTP message (can be FQDN). |
| 5 | RESPONSE STATUS | Displays the HTTP Response. It can be any 2xx, 4xx or 5xx response. |

# SMF Service

This section describes the supported counters and set of labels for the SMF service.

## Labels

The SMF service includes the following labels for the counters:

*Table 3: SMF Service Labels for Counters*

| Number | Label | Description |
|---|---|---|
| 1 | PROCEDURE TYPE | This label can take any value depending on the type of procedure queried for:<br>"pdu_sess_create"<br>"ue_req_pdu_sess_mod"<br>"smf_req_pdu_sess_mod"<br>"pcf_req_pdu_sess_mod"<br>"ue_req_pdu_sess_rel"<br>"smf_req_pdu_sess_rel"<br>"pcf_req_pdu_sess_rel"<br>"amf_req_pdu_sess_rel" |
| 2 | STATUS | Displays the status type. The possible values are:<br>“attempted”<br>“success”<br>“failure” |
| 3 | PDU CONNECTION TYPE | Displays the PDU connection type. The possible values are:<br>“ipv4”<br>“ipv6”<br>“ipv4v6” |
| 4 | PDU STATE | Displays the PDU state. The possible values are:<br>“idle”<br>“connected” |

# SMF Protocol Microservice

This section describes the supported counters and set of labels for the SMF Protocol microservice.

## Counters

The SMF service includes the following counters:

*Table 4: SMF Service Counters*

| Number | Metric | Description |
|---|---|---|
| 1 | smf_service_stats | This counter is incremented with every query made to the smf-service. |

| Number | Metric | Description |
|---|---|---|
| 2 | smf_service_counters | This is a gauge counter and can be incremented/decremented based on the functionality with every query made to the smf-service. |

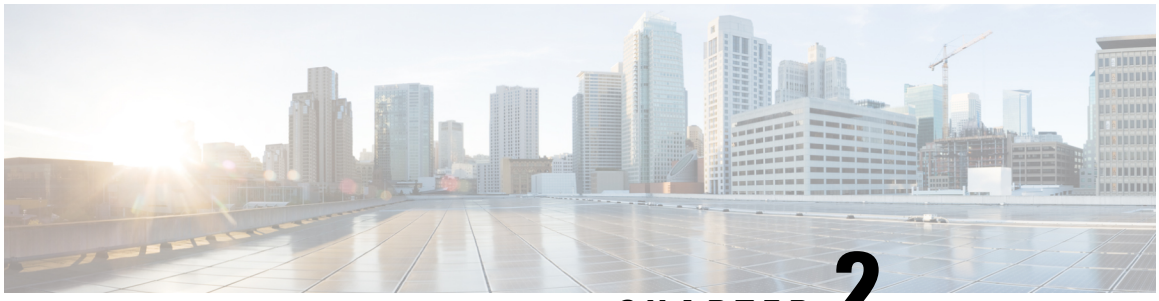
## Labels

The SMF Protocol service includes the following labels for the counters:

***Table 5: SMF Protocol Service Labels for Counters***

| Number | Label | Description |
|---|---|---|
| 1 | MESSAGE NAME | This label can take any value depending on the procedure queried for:<br>"n4_session_establishment_req"<br>"n4_session_establishment_res"<br>"n4_session_modification_req"<br>"n4_session_modification_res"<br>"n4_session_report_req"<br>"n4_session_report_res"<br>"n4_session_deletion_req"<br>"n4_session_deletion_res"<br>"n4_association_setup_req"<br>"n4_association_setup_res"<br>"n4_association_update_req"<br>"n4_association_update_res"<br>"n4_association_release_req"<br>"n4_association_release_res"<br>"n4_prime_pfd_management_req"<br>"n4_prime_pfd_management_res"<br>"n4_heartbeat_req"<br>"n4_heartbeat_res"<br>"n4_node_report_req"<br>"n4_node_report_res" |

| Number | Label | Description |
|---|---|---|
| 2 | MESSAGE DIRECTION | Displays the direction of the HTTP message with respect to the REST EP microservice. The possible values are:<br>“inbound”<br>“outbound” |
| 3 | STATUS | Displays the status of the message. The possible values are:<br>“accepted”<br>“denied”<br>“discarded” |



CHAPTER 2

# SMF Metrics

• SMF User Plane Session counters Category, on page 30

# Discover Messages Time statistics Category

**nf_discover_total_time**

Description: Discover Messages Total time statistics

Sample Query: `nf_discover_total_time{nf_type=\"amf\", host=\"http://10.105.227.109:8082/nnrf-nfm/v1\", result=\"timeouOrRPCError\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `host`

  Label Description: End Point address

  Example: http://10.105.227.109:8082/nnrf-nfm/v1

- Label: `result`

  Label Description: result of discover message

  Example: 200, 201, 204, success, timeout_rpc_error, response_parse_failure

# Discover Messages statistics Category

**nf_discover_messages_total**

Description: Discover Messages statistics

Sample Query: `nf_discover_messages_total{nf_type=\"amf\", host=\"http://10.105.227.109:8082/nnrf-nfm/v1\", result=\"timeouOrRPCError\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `host`

  Label Description: End Point address

  Example: http://10.105.227.109:8082/nnrf-nfm/v1

- Label: `result`

  Label Description: result of discover message

Example: 200, 201, 204, success, timeout_rpc_error, response_parse_failure

# NF End point selections Category

**nf_endpoint_selections_total**

Description: NF End Point Selection Statistics

Sample Query: `nf_endpoint_selections_total{nf_type=\"amf\", host=\"http://10.105.227.109:8082/nnrf-nfm/v1\", req=\"initial\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `host`

  Label Description: End Point address

  Example: http://10.105.227.109:8082/nnrf-nfm/v1

- Label: `req`

  Label Description: req type

  Example: initial, fallback,

# NF failure handling stats Category

**nf_failure_handling_stats_total**

Description: NF Failure handling stats

Sample Query: `nf_failure_handling_stats_total{nf_type=\"amf\", host=\"http://10.105.227.109:8082/nnrf-nfm/v1\", req=\"initial\", response=\"202\", status=\"final\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `host`

  Label Description: End Point address

  Example: http://10.105.227.109:8082/nnrf-nfm/v1

- Label: `req`

Label Description: Request type

Example: initial, fallback,

- Label: `response`

  Label Description: Response from the server

  Example: 200, 201, 204, timeout_rpc_error,

- Label: `status`

  Label Description: Status from the server

  Example: retry, final

# NF management message time statistics Category

**nf_management_total_time**

Description: NF management messages total time taken

Sample Query: `nf_management_total_time{host=\"http://10.105.227.109:8082/nnrf-nfm/v1\", direction=\"outbound\", message_type=\"registration\",result=\"timeouOrRPCError\" }`

Labels:

- Label: `host`

  Label Description: End Point address

  Example: http://10.105.227.109:8082/nnrf-nfm/v1

- Label: `direction`

  Label Description: Direction indicates about the message going out or coming in

  Example: inbound, outbound

- Label: `message_type`

  Label Description: Type of Message

  Example: registration, hearbeat, subscription, notification

- Label: `result`

  Label Description: result of discover message

  Example: 200, 201, 204, success, timeout_rpc_error, response_parse_failure, request_parse_failure, invalid_notify_event, invalid_nf_instance_uri, internal_error

# NF management messages statistics Category

**nf_management_stats_total**

Description: NF management messages statistics

Sample Query: `nf_management_stats_total{host=\"http://10.105.227.109:8082/nnrf-nfm/v1\", direction=\"outbound\", message_type=\"registration\",result=\"timeouOrRPCError\" }`

Labels:

- Label: `host`

  Label Description: End Point address

  Example: http://10.105.227.109:8082/nnrf-nfm/v1

- Label: `direction`

  Label Description: Direction indicates about the message going out or coming in

  Example: inbound, outbound

- Label: `message_type`

  Label Description: Type of Message

  Example: registration, hearbeat, subscription, notification

- Label: `result`

  Label Description: result of discover message

  Example: 200, 201, 204, success, timeout_rpc_error, response_parse_failure

# NRF Discovery Category

**nf_discover_events_total**

Description: NF Discover Stats

Sample Query: `nf_discover_events_total{nf_type=\"pcf\", response_type=\"local\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `response_type`

  Label Description: Discovery response choosen from

  Example: local, cache, expired-cache

# PDU UE Sync Procedure Category

**pdu_ue_sync_proc**

Description: PDU UE Sync Procedure counter

Sample Query: `pdu_ue_sync_proc{status=\"attempted\"}`

Labels:

- Label: `status`

  Label Description: call flow procedure status counter

  Example: attempted, success, failures, suspend, resume, abort

# Policy control PCF update statistics Category

**policy_pcf_updates_total**

Description: Statistics for triggers sent to PCF in SmPolicyUpdate Request to PCF

Sample Query: `sum (policy_pcf_updates_total{trigger=\"rat_change\"})`

Labels:

- Label: `trigger`

  Label Description: Trigger sent in the policy update request sent to PCF

  Example: ue_ip_change, plmn_change, res_mod_req, access_type_change , ue_ip_change , credit_mon_sess_fail , def_qos_change , sess_ambr_change , no_credit , serving_area_change , revalidation_timeout ,resoure_release,resource_alloc, rat_change

# Policy control current flow Category

**policy_pdu_flows_current**

Description: QoS flow current counts

Sample Query: `sum (policy_pdu_flows_current{flow_type=\"gbr\"}) by(qos_5qi, arp)`

Labels:

- Label: `rat_type`

  Label Description: RAT type on which the flow is created

  Example: nr, eutran

- Label: `ssc_mode`

  Label Description: SSC mode for the session which created the QoS flow

  Example: one, two, three

- Label: `pdn_type`

  Label Description: PDN type of the session which created the QoS flow

  Example: v4, v6, v4v6

- Label: `dnn`

Label Description: DNN for which the flow is created

Example: cisco.com

- Label: `flow_type`

  Label Description: Flow type for the QoS flow

  Example: gbr, non_gbr

- Label: `qos_5qi`

  Label Description: 5Qi applicable for the QoS flow

  Example: 1, 2, 5

- Label: `arp`

  Label Description: Priority level of ARP applicable for the QoS flow

  Example: 10, 20

# Policy control dynamic pcc rule statistics Category

**policy_dynamic_pcc_rules_total**

Description: PCC Rule total statistics for dynamic rules pushed from PCF

Sample Query: `sum (policy_dynamic_pcc_rules_total{rule_id=\"Rule-1\"}) by(qos_5qi, arp)`

Labels:

- Label: `rule_id`

  Label Description: Rule Id for the received dynamic pcc rule

  Example: PccRule-1

- Label: `operation`

  Label Description: Operation performed on the dynamic pcc rule

  Example: install, modify, remove

- Label: `event`

  Label Description: Event associated with the operation performed on the pcc rule

  Example: attempted, success, failure, abort

- Label: `qos_5qi`

  Label Description: 5Qi applied on the dynamic pcc rule

  Example: 1, 2, 5

- Label: `arp`

  Label Description: Priority level of ARP applied on the dynamic pcc rule

  Example: 10, 20

- Label: `tc_event`

  Label Description: Traffic Control event applied on the dynamic pcc rule

  Example: enabled_ul, enabled_dl, enabled, disabled, removed

- Label: `charging_type`

  Label Description: Charging type applied on the dynamic pcc rule

  Example: online, offline, online-offline

- Label: `charging_method`

  Label Description: Charging method applied on the dynamic pcc rule

  Example: volume, time, vol_time

- Label: `details`

  Label Description: Details on the operation applied on the dynamic pcc rule

  Example: success, validation failure

# Policy control pre-defined pcc rule statistics Category

**policy_predefined_pcc_rules_total**

Description: PCC Rule total statistics for pre-defined rules activated by PCF

Sample Query: `sum (policy_predefined_pcc_rules_total{rule_id=\"Rule-1\"}) by(event, operation)`

Labels:

- Label: `rule_id`

  Label Description: Rule Id for the received pre-defined pcc rule

  Example: PccRule-1

- Label: `rulebase`

  Label Description: Rulebase to which this pre-defined rule belongs

  Example: Rulebase-1

- Label: `operation`

  Label Description: Operation performed on the pre-defined pcc rule

  Example: install, modify, remove

- Label: `event`

  Label Description: Event associated with the operation performed on the pre-defined rule

  Example: attempted, success, failure

- Label: `qos_5qi`

  Label Description: 5Qi applied on the pre-defined pcc rule

Example: 1, 2, 5

- Label: `arp`

  Label Description: Priority level of ARP applied on the pre-defined pcc rule

  Example: 10, 20

- Label: `charging_type`

  Label Description: Charging type applied on the pre-defined pcc rule

  Example: online, offline, online-offline

- Label: `charging_method`

  Label Description: Charging method applied on the pre-defined pcc rule

  Example: volume, time, vol_time

# Policy control session rule statistics Category

**policy_session_rules_total**

Description: Session total statistics for session rules applied

Sample Query: `sum (policy_session_rules_total{rule_id=\"SessRule-1\"})`

Labels:

- Label: `rule_id`

  Label Description: Rule Id for the received session rule from PCF

  Example: SessRule-1

- Label: `operation`

  Label Description: Operation performed on the session rule

  Example: install, modify, remove

- Label: `event`

  Label Description: Event associated with the operation performed on the rulebase

  Example: attempted, success, failure

# Policy control static pcc rule statistics Category

**policy_static_pcc_rules_total**

Description: PCC Rule total statistics for static rules activated via rulebase

Sample Query: `sum (policy_static_pcc_rules_total{rulebase=\"Rulebase-1\"})`

Labels:

- Label: `rulebase`

  Label Description: Rulebase to which the static rules belong

  Example: Rulebase-1

- Label: `operation`

  Label Description: Operation performed on the rulebase

  Example: install, remove

- Label: `event`

  Label Description: Event associated with the operation performed on the rulebase

  Example: attempted, success, failure

# Policy control total flow statistics Category

**policy_pdu_flows_total**

Description: QoS flow total statistics

Sample Query: `sum (policy_pdu_flows_total{flow_type=\"gbr\"}) by(qos_5qi, arp)`

Labels:

- Label: `operation`

  Label Description: Operation performed on the QoS flow

  Example: install, modify, remove

- Label: `event`

  Label Description: Event associated with the operation performed on QoS flow

  Example: attempted, success, failure, abort

- Label: `rat_type`

  Label Description: RAT type on which the flow is created

  Example: nr, eutran

- Label: `ssc_mode`

  Label Description: SSC mode for the session which created the QoS flow

  Example: one, two, three

- Label: `pdn_type`

  Label Description: PDN type of the session which created the QoS flow

  Example: v4, v6, v4v6

- Label: `dnn`

  Label Description: DNN for which the flow is created

Example: cisco.com

- Label: `flow_type`

  Label Description: Flow type for the QoS flow

  Example: gbr, non_gbr

- Label: `init_or_ho`

  Label Description: Flow operation phase

  Example: initial, ho

- Label: `qos_5qi`

  Label Description: 5Qi applicable for the QoS flow

  Example: 1, 2, 5

- Label: `arp`

  Label Description: Priority level of ARP applicable for the QoS flow

  Example: 10, 20

# SLA Transaction Category

**smf_sla_transaction_stats**

desription: Transaction SLA stats

Sample Query: `sum(smf_sla_transaction_stats) by (smf_sla_transaction_stats,smf_proc_type,status, message_type)`

Labels:

- Label: `smf_proc_type`

  Label Description: procedure type counter

  Example: pdu_sess_create

- Label: `status`

  Label Description: gives status of the procedure

  Example: abort

- Label: `message_type`

  Label Description: gives the message type received during sla transaction

  Example: IntSelfTxnSla

# SMF ALWAYS ON PDU SESSION Category

metrics: SMF_ALWAYS_ON_SESSION_STATS

Description: Always On Pdu Session Statistics

- Label: `status`

  Label Description: always on status statistics

  Example: pdusetup_req_alwayson_requested, pdusetup_acc_alwayson_allowed, pdusetup_acc_alwayson_not_allowed, pdumod_req_alwayson_requested, pdumod_cmd_alwayson_allowed, pdumod_cmd_alwayson_not_allowed, pdumod_cmd_nw_init_alwayson_allowed, pdu_utwifi_to_nr_alwayson_requested, pdu_utwifi_to_nr_alwayson_allowed, pdu_utwifi_to_nr_alwayson_not_allowed

- Label: `rat_type`

  Label Description: Type of the radio access associated with the request

  Example: EUTRA, NR, WLAN, VIRTUAL, rat_type_unknown

- Label: `pduType`

  Label Description: name of the nssai associated with the request

  Example: ipv4, ipv6, ipv4v6, unknown

- Label: `dnn`

  Label Description: name of the dnn associated with the request

  Example: Any string

- Label: `sscMode`

  Label Description: Type of ssc mode associated with the request

  Example: ssc_mode_1, ssc_mode_2, ssc_mode_3, ssc_mode_unknown

# SMF Charging Application Error Stats Category

### chf_appl_err_stats

Description: Statistics for application error received from CHF

Sample Query: `chf_appl_err_stats{appl_err_code=\"HTTP_STATUS_CODE_403_FORBIDDEN\"}`

Labels:

- Label: `http2_err_code`

  Label Description: HTTP2 error code received from CHF

  Example: HTTP_STATUS_CODE_403_FORBIDDEN

- Label: `appl_err_code`

  Label Description: Application error code received from CHF

  Example: END_USER_REQUEST_REJECTED, QUOTA_LIMIT_REACHED,CHARGING_FAILED, USER_UNKNOWN, END_USER_REQUEST_DENIED, QUOTA_LIMIT_REACHED, CHARGING_NOT_APPLICABLE

- Label: `appl_err_action`

  Label Description: Action taken on failure from CHF

  Example: Terminate, Drop Traffic, Disable Charging

- Label: `appl_err_exchg_type`

  Label Description: CHF Exchange in which failure occurred

  Example: update, initial

# SMF Charging Quota Event Stats Category

**chf_quota_event_stats**

Description: The current count for quota event received from CHF

Sample Query: `chf_quota_event_stats{quota_type=\"initial\"}`

Labels:

- Label: `quota_method`

  Label Description: Quota method received from CHF

  Example: time, volume, time_volume

- Label: `quota_status`

  Label Description: Resutlt for the quota received from CHF

  Example: SUCCESS, END_USER_SERVICE_DENIED, QUOTA_MANAGEMENT_NOT_APPLICABLE, QUOTA_LIMIT_REACHED, END_USER_SERVICE_REJECTED, RATING_FAILED

- Label: `quota_type`

  Label Description: Quota type as received from CHF

  Example: intial, update, initial_final, update_final, fail

- Label: `quota_fail_action`

  Label Description: Action on quota failure

  Example: No Action , Disable charging, Drop Traffic, Offline Converted

- Label: `rating_group`

  Label Description: Rating group for which quota is received from CHF

  Example: Any string

# SMF Charging Usage Report Stats Category

### chf_usage_report_stats

Description: The current count for usage reports towards CHF

Sample Query: `chf_usage_report_stats{charging_method=\"offline\"}`

Labels:

- Label: `charging_method`

  Label Description: Metering method for the PDU Session

  Example: online, offline, online_offline

- Label: `charging_trigger_type`

  Label Description: Trigger for usage report

  Example: QUOTA_THRESHOLD, QHT, FINAL, QUOTA_EXHAUSTED, VALIDITY_TIME, OTHER_QUOTA_TYPE, FORCED_REAUTHORISATION, UNIT_COUNT_INACTIVITY _TIMER, ABNORMAL_RELEASE, QOS_CHANGE, VOLUME_LIMIT, TIME_LIMIT, EVENT_LIMIT, PLMN,CHANGE, USER_LOCATION_CHANGE, RAT_CHANGE, UE_TIMEZONE_CHANGE, TARIFF_TIME_CHANGE, MAX_NUMBER_OF_CHANGES_IN_CHARGING_CONDITIONS, MANAGEMENT_INTERVENTION, CHANGE_OF_UE_PRESENCE_IN_PRESENCE_REPORTING_AREA, CHANGE_OF_3GPP_PS_DATA_OFF_STATUS, SERVING_NODE_CHANGE, REMOVAL_OF_UPF, ADDITION_OF_UPF, START_OF_SERVICE_DATA_FLOW, AMBR_CHANGE

- Label: `rating_group`

  Label Description: Rating Group for which usage is being reported

  Example: Any string

- Label: `service_identifier`

  Label Description: Service Identifier for which usage is being reported

  Example: Any string

# SMF Disconnect stats Category

### smf_disconnect_stats

Description: SMF Disconnect stats counters

Sample Query: `smf_disconnect_stats{reason=\"disc_pdurel_amf_init_detach\"}`

Labels:

- Label: `reason`

  Label Description: The reason associated with an call disconnect

Example: disc_pdusetup_create_over_create, disc_pdusetup_admin_clear, disc_pdusetup_n1_decode_failure, disc_pdusetup_n1_content_not_found, disc_pdusetup_sess_abs_timeout, disc_pdusetup_ssc_mode_not_supported, disc_pdusetup_ssc_mode_denied, disc_pdusetup_identity_conflict, disc_pdusetup_pdutype_unsupported, disc_pdusetup_pdutype_denied, disc_pdusetup_init_chg_data_err, disc_pdusetup_ip_alloc_failed, disc_pdusetup_pdu_fetch_failure, disc_pdusetup_udm_reg_failed, disc_pdusetup_udm_sub_fetch_failure, disc_pdusetup_udm_sub_notify_failed, disc_pdusetup_upf_setup_cause_not_accepted, disc_pdusetup_sm_cxt_invalid, disc_pdusetup_sm_cxt_invalid_ie, disc_pdusetup_sm_cxt_sess_id_err, disc_pdusetup_sm_cxt_invalid_json, disc_pdusetup_sm_cxt_n1_process_failed, disc_pdusetup_sm_cxt_man_param_missing, disc_pdusetup_pcf_create_exchg_failure, disc_pdusetup_pcf_create_rsp_failure, disc_pdusetup_rm_exchg_failure, disc_pdusetup_rm_rsp_failure, disc_pdusetup_pcf_update_exchg_failure, disc_pdusetup_pcf_update_rsp_failure, disc_pdusetup_chf_data_exchg_failure, disc_pdusetup_chf_data_rsp_failure, disc_pdusetup_upf_setup_exchg_failure, disc_pdusetup_upf_setup_rsp_failure, disc_pdusetup_n1n2_transfer_exchg_failure, disc_pdusetup_n1n2_transfer_rsp_failure, disc_pdusetup_n1n2_transfer_failure_rsp_code, disc_pdusetup_amf_assign_ebi_failure, disc_pdusetup_upf_modify_exchg_failure, disc_pdusetup_upf_modify_rsp_failure, disc_pdusetup_upf_modify_failed, disc_pdusetup_upf_serv_data_nill, disc_pdusetup_upf_dl_tunnel_info_not_found, disc_pdusetup_upf_tunnel_id_not_found, disc_pdusetup_upf_mod_gnb_tun_params_failed, disc_pdusetup_upf_mod_rsra_tun_params_failed, disc_pdusetup_upf_mod_tun_param_tos-failed, disc_pdurel_amf_sends_ue_not_found, disc_pdusetup_dnn_missing, disc_pdurel_ue_init_detach, disc_pdurel_amf_init_detach, disc_pdurel_pcf_init_detach, disc_pdurel_udm_init_detach, disc_pdurel_gnb_init_detach, disc_pdurel_chf_init_detach, disc_n2ho_n4_modify_failed, disc_n26_4g_5g_ho_n4_modify_failed, disc_pdumodify_context_not_found, disc_pduim_context_not_found

# SMF GTPC messages Category

**smf_gtpc_app_events**

Description: SMF GTPC message counter

Sample Query: `smf_gtpc_app_events{service_name=\"gtpc-ep\"}`

Labels:

- Label: `event_type`

  Label Description: Gtpc Event type

  Example: NumRxCreateSessionReq, NumTxCreateSessionRes, NumRxDeleteSessionReq, NumTxDeleteSessionRes, NumRxModifyBearerReq, NumTxModifyBearerRes, NumTxDeleteBearerReq, NumRxDeleteBearerRsp, NumTxCreateBearerReq, NumRxCreateBearerRes, NumTxUpdateBearerReq, NumRxUpdateBearerRes, NumTxModifyBearerFailureInd, NumModifyBearerTimeout, NumRxDeleteBearerCmd, NumCreateBearerFailure, NumCreateBearerSuccess, NumCreateSessionSuccess, NumCreateSessionFailure, NumDeleteSessionSuccess, NumDeleteSessionFailure, NumCreateBearerReqRetrans, NumUpdateBearerReqRetrans, NumDeleteBearerReqRetrans

# SMF IPAM Address Events Current Counter Category

### IPAM_address_allocations_current

Description: Current state of SMF IPAM Address allocations

Sample Query:

```
IPAM_address_allocations_current{dnn=\"dnn1\",servingArea=\"area1\",nssai=\"slice1\",pool=\"p1\",allocationType=\"dynamic\",addressType=\"IPv4\",upf=\"upf1\"}
```

Labels:

- Label: `dnn`

  Label Description: name of the dnn associated with the request

  Example: Any string

- Label: `servingArea`

  Label Description: name of the serving area associated with the request

  Example: Any string

- Label: `nssai`

  Label Description: name of the nssai associated with the request

  Example: Any string

- Label: `pool`

  Label Description: name of the pool associated with the request

  Example: Any string

- Label: `allocationType`

  Label Description: type of allocation associated with the request

  Example: static/dynamic

- Label: `addressType`

  Label Description: address type associated with the request

  Example: IPv4/IPv6PD

- Label: `upf`

  Label Description: upf identifier associated with the request

  Example: Any string

# SMF IPAM Address Events Total Counter Category

### IPAM_address_events_total

Description: Total number of SMF IPAM Address events

Sample Query:

```
IPAM_address_events_total{dnn=\"dnn1\",servingArea=\"area1\",nssai=\"slice1\",pool=\"p1\",eventType=\"Allocation\",allocationType=\"dynamic\",addressType=\"IPv4\",upf=\"dp1\"}
```

Labels:

- Label: `dnn`

  Label Description: name of the dnn associated with the request

  Example: Any string

- Label: `servingArea`

  Label Description: name of the serving area associated with the request

  Example: Any string

- Label: `nssai`

  Label Description: name of the nssai associated with the request

  Example: Any string

- Label: `pool`

  Label Description: name of the pool associated with the request

  Example: Any string

- Label: `eventType`

  Label Description: type of event associated with the request

  Example: Allocation/Release

- Label: `allocationType`

  Label Description: type of allocation associated with the request

  Example: static/dynamic

- Label: `addressType`

  Label Description: address type associated with the request

  Example: IPv4/IPv6PD

- Label: `upf`

  Label Description: upf identifier associated with the request

  Example: Any string

# SMF IPAM Chunk Events Current Counter Category

### IPAM_chunk_allocations_current

Description: Current state of SMF IPAM Address Chunk allocations

Sample Query:

```
IPAM_chunk_allocations_current{dnn=\"dnn1\",servingArea=\"area1\",nssai=\"slice1\",pool=\"p1\",addressType=\"IPv4\",upf=\"dp1\"}
```

Labels:

- Label: `dnn`

  Label Description: name of the dnn associated with the request

  Example: Any string

- Label: `servingArea`

  Label Description: name of the serving Area associated with the request

  Example: Any string

- Label: `nssai`

  Label Description: name of the nssai associated with the request

  Example: Any string

- Label: `pool`

  Label Description: name of the pool associated with the request

  Example: Any string

- Label: `addressType`

  Label Description: address type associated with the request

  Example: IPv4/IPv6PD

- Label: `upf`

  Label Description: upf identifier associated with the request

  Example: Any string

# SMF IPAM Chunk Events Total Counter Category

### IPAM_chunk_events_total

Description: Total number of SMF IPAM Address Chunk events

Sample Query:

```
IPAM_chunk_events_total{dnn=\"dnn1\",servingArea=\"area1\",nssai=\"slice1\",pool=\"p1\",eventType=\"Allocation\",addressType=\"IPv4\",upf=\"dp1\"}
```

Labels:

- Label: `dnn`

  Label Description: name of the dnn associated with the request

  Example: Any string

- Label: `servingArea`

  Label Description: name of the serving area associated with the request

  Example: Any string

- Label: `nssai`

Label Description: name of the nssai associated with the request

Example: Any string

- Label: `pool`

  Label Description: name of the pool associated with the request

  Example: Any string

- Label: `eventType`

  Label Description: type of event associated with the request

  Example: Allocation/Release

- Label: `addressType`

  Label Description: address type associated with the request

  Example: IPv4/IPv6PD

- Label: `upf`

  Label Description: upf identifier associated with the request

  Example: Any string

# SMF PDU Status Category

**smf_service_counters**

Description: The current count of SMF pdu sessions

Sample Query: `smf_service_counters{pdu_state=\"all_pdu\"}`

Labels:

- Label: `pdu_state`

  Label Description: PDU session status indicated by N3 UPF tunnel status

  Example: all_pdu, idle, connected

# SMF Procedure Category

**smf_service_stats**

Description: SMF call flow procedure counters

Sample Query: `smf_service_stats{procedure_type=\"pdu_sess_create\"}`

Labels:

- Label: `procedure_type`

  Label Description: The procedure type associated with an call flow procedure

Example: pdu_sess_create, ue_req_pdu_sess_mod, smf_req_pdu_sess_mod, pcf_req_pdu_sess_mod, udm_req_pdu_sess_mod, gnb_req_pdu_sess_mod, ue_req_pdu_sess_rel, smf_req_pdu_sess_rel, pcf_req_pdu_sess_rel, amf_req_pdu_sess_rel, udm_req_pdu_sess_rel, gnb_req_pdu_sess_rel, chf_req_pdu_sess_rel, admin_req_pdu_sess_rel, ue_req_active_to_idle, ue_req_idle_to_active, nw_req_service_active, upf_notify_downlink_data, xn_path_switch,pdn_sess_create,pdn_5g_4g_handover,pcf_req_ded_brr_create, pcf_req_ded_brr_delete, pcf_req_ded_brr_mod, n2_handover, xn_handover, n26_4g_to_5g_handover, n26_4g_to_5g_im_mobility, pdu_im, pdn_sess_create, pcf_req_ded_brr_create, pcf_req_ded_brr_delete, pcf_req_ded_brr_mod, pcf_initiated_pdn_detach, smf_initiated_pdn_detach, upf_initiated_pdn_detach

- Label: `status`

  Label Description: call flow procedure counter

  Example: attempted, success, failures

- Label: `pdu_type`

  Label Description: Type of pdu session

  Example: ipv4, ipv6, ipv4v6, unknown

- Label: `dnn`

  Label Description: Dnn requested by UE

  Example: intershat, cisco.com

- Label: `reason`

  Label Description: Reason for failure status. For sucessess and attempted it will be Empty

  Example: udm_registration_failure, udm_subscription_fetch_failure, udm_subscribe_notify_failure, pcf_create_failure, pcf_update_failure, charging_data_failure, pdn_5g_4g_handover, idft_setup_failure, mbr_setup_failure, sgw_failure, upf_failure, pcf_req_ded_brr_mod, pcf_req_ded_brr_create, pcf_req_ded_brr_delete, ue_req_ded_brr_mod, udm_req_ded_brr_mod, smf_req_ded_brr_mod, rm_failure, upf_setup_failure, upf_modify_failure, ebi_assign_failure, n1n2_transfer_failure, smcontext_modify_req_processing_failure, session_setup_timer_expired, pcf_update_exchg_failure, create_over_create, sm_cxt_sess_id_err, sm_cxt_invalid_json, sm_cxt_invalid_ie, sm_cxt_invalid, sm_cxt_n1_process_failed, n1n2_transfer_exchg_failure, n2ho_pdu_not_establised, n2ho_ie_validation_failed, n2ho_n4_ho_preparing_failed, n2ho_n4_ho_prepared_failed, n2ho_n4_ho_completed_failed, n2ho_ho_cancelled, n2ho_resource_alloc_unsuccess_transfer, n2ho_invalid_state, n2ho_preparation_unsuccess_transfer, n2ho_n1n2_transfer_failure, , xnho_pdu_not_establised, xnho_tobe_switched_flag_not_set, xnho_dl_tunnel_info_not_found, xnho_invalid_accepted_qfi_list, xnho_n4_modification_failed, xnho_n1n2_transfer_failure, xnho_pdu_ctx_not_found, , n26ho_5g_4g_n1n2_transfer_failure, n26ho_5g_4g_invalid_state, n26ho_5g_4g_n4_failed_prepared_state, n26ho_5g_4g_resource_alloc_unsuccess_transfer, n26ho_5g_4g_n4_failed_completed_state, n26ho_5g_4g_handover_cancelled, n26ho_5g_4g_pdu_ctx_not_found, n26ho_5g_4g_n4_failed_preparing_state, n26ho_5g_4g_n4_modify_failed, n26ho_5g_4g_invalid_eps_pdn_connlist, , pduim_n1n2_transfer_failure, pduim_n1n2_txfr_failure_notification, pduim_n4_modification_failed, pduim_misc_error, pduim_suspended_procedure, pduim_amf_ctx_not_found, pduim_internal_error, pduim_upstate_not_in_deactivated_state, n26ho_4g_5g_no_eps_5gs_continuity, n26im_mobility_4g_5g_no_eps_5gs_continuity, n26im_mobility_4g_5g_default_eps_bearer_inactive, pduimn26mob_n2_setup_response_sucess, pduimn26mob_n2_setup_response_failure

# SMF Protocol message counters Category

**smf_proto_udp_msg_total**

Description: SMF Protocol message statistics

Sample Query: `smf_proto_udp_msg_total{message_direction=\"inbound\",nf_type=\"amf\"}`

Labels:

- Label: `message_name`

  Label Description: name of N4 interface message

  Example: n4_session_establishment_req, n4_session_establishment_res, n4_session_modification_req, n4_session_modification_res, n4_session_report_req, n4_session_report_res, n4_session_deletion_req, n4_session_deletion_res, n4_association_setup_req, n4_association_setup_res, n4_association_update_req, n4_association_update_res, n4_association_release_req, n4_association_release_res, n4_prime_pfd_management_req, n4_prime_pfd_management_res, n4_heartbeat_req, n4_heartbeat_res, n4_node_report_req, n4_node_report_res

- Label: `message_direction`

  Label Description: direction of message from SMF perspective

  Example: inbound, outbound

- Label: `status`

  Label Description: status of message processing

  Example: accepted, denied, discarded

# SMF REST message exchange time Category

**smf_restep_http_msg**

Description: SMF REST time between request and response messages

Sample Query: `smf_restep_http_msg{message_direction=\"inbound\",nf_type=\"amf\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `message_direction`

  Label Description: direction of message from SMF perspective

  Example: inbound, outbound

- Label: `api_name`

Label Description: API name

Example: register_ue, deregister_ue, subscription_req, sdm_subscription_req, sdm_data_change_notify, nf_registration, nf_discovery, slice_selection, amf_create_sm_context, amf_update_sm_context, amf_release_sm_context, amf_n1_n2_transfer, amf_n1_n2_transfer_notify_failure, amf_assign_ebi, amf_status_notify, pcf_sm_policy_control_create, chf_charging_data_request, pcf_sm_policy_control_update, pcf_sm_policy_control_delete, pcf_sm_policy_control_update_notify, cisco_control_clear_subscriber, cisco_control_show_subscriber, pcf_sm_policy_control_terminate_notify, chf_abort_notify

- Label: `nf_uri`

  Label Description: Network Function URI

  Example: actual HTTP URI of the message

- Label: `response_status`

  Label Description: HTTP response status code

  Example: 200, 201, 204

# SMF REST messages Category

**smf_restep_http_msg_total**

Description: SMF REST message counter

Sample Query: `smf_restep_http_msg_total{message_direction=\"inbound\",nf_type=\"amf\"}`

Labels:

- Label: `nf_type`

  Label Description: Network Function type

  Example: nrf, udm, amf, pcf, chf, ciscocontrol

- Label: `message_direction`

  Label Description: direction of message from SMF perspective

  Example: inbound, outbound

- Label: `api_name`

  Label Description: API name

  Example: register_ue, deregister_ue, subscription_req, sdm_subscription_req, sdm_data_change_notify, nf_registration, nf_discovery, slice_selection, amf_create_sm_context, amf_update_sm_context, amf_release_sm_context, amf_n1_n2_transfer, amf_n1_n2_transfer_notify_failure, amf_assign_ebi, amf_status_notify, pcf_sm_policy_control_create, chf_charging_data_request, pcf_sm_policy_control_update, pcf_sm_policy_control_delete, pcf_sm_policy_control_update_notify, cisco_control_clear_subscriber, cisco_control_show_subscriber, pcf_sm_policy_control_terminate_notify, chf_abort_notify

- Label: `nf_uri`

Label Description: Network Function URI

Example: actual HTTP URI of the message

- Label: `response_status`

  Label Description: HTTP response status code

  Example: 200, 201, 204

# SMF Session counters Category

**smf_session_counters**

Description: SMF current acitve Session counters

Sample Query:

```
smf_session_counters{ratType=\"rat_type_4g\",pduType=\"ipv4\",dnn=\"dnn1\",sscMode=\"ssc_mode_1\"}
```

Labels:

- Label: `rat_type`

  Label Description: Type of the radio access associated with the request

  Example: rat_type_2g, rat_type_3g, rat_type_4g, rat_type_5g, rat_type_4g_5g, rat_type_wifi, rat_type_unknown

- Label: `pduType`

  Label Description: name of the nssai associated with the request

  Example: ipv4, ipv6, ipv4v6, unknown

- Label: `dnn`

  Label Description: name of the dnn associated with the request

  Example: Any string

- Label: `sscMode`

  Label Description: Type of ssc mode associated with the request

  Example: ssc_mode_1, ssc_mode_2, ssc_mode_3, ssc_mode_unknown

- Label: `always_on`

  Label Description: always on status

  Example: enable, disable

# SMF Session stats Category

**smf_session_stats**

Description: SMF Session stats counters

Sample Query:
`smf_session_stats{ratType=\"rat_type_4g\",pduType=\"ipv4\",dnn=\"dnn1\",sscMode=\"ssc_mode_1\",status=\"attempted\"}`

Labels:

- Label: `rat_type`

  Label Description: Type of the radio access associated with the request

  Example: rat_type_2g, rat_type_3g, rat_type_4g, rat_type_5g, rat_type_4g_5g, rat_type_wifi, rat_type_unknown

- Label: `pduType`

  Label Description: name of the nssai associated with the request

  Example: ipv4, ipv6, ipv4v6, unknown

- Label: `dnn`

  Label Description: name of the dnn associated with the request

  Example: Any string

- Label: `sscMode`

  Label Description: Type of ssc mode associated with the request

  Example: ssc_mode_1, ssc_mode_2, ssc_mode_3, ssc_mode_unknown

- Label: `status`

  Label Description: PDU session status indicated at SMF

  Example: attempted, success, setup

# SMF User Plane Session counters Category

### smf_up_session_counters

Description: SMF current active User Plane Sessions

Sample Query:
`smf_up_session_counters{ratType=\"rat_type_4g\",pduType=\"ipv4\",dnn=\"dnn1\",sscMode=\"ssc_mode_1\"}`

Labels:

- Label: `rat_type`

  Label Description: Type of the radio access associated with the request

  Example: rat_type_2g, rat_type_3g, rat_type_4g, rat_type_5g, rat_type_4g_5g, rat_type_wifi, rat_type_unknown

- Label: `pduType`

  Label Description: name of the nssai associated with the request

  Example: ipv4, ipv6, ipv4v6, unknown

- Label: `dnn`

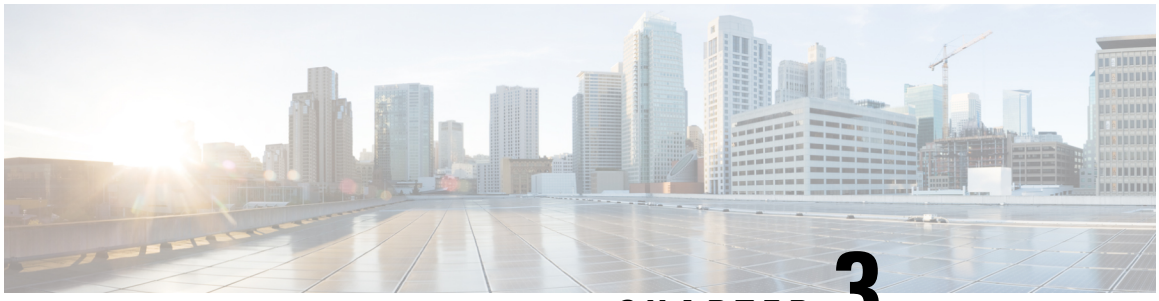
Label Description: name of the dnn associated with the request

Example: Any string

- Label: `sscMode`

  Label Description: Type of ssc mode associated with the request

  Example: ssc_mode_1, ssc_mode_2, ssc_mode_3, ssc_mode_unknown



CHAPTER 3

# MIB Reference



# CISCO-CNEE-MIB

### ciscoCneeMIB Module Identity

Last Updated: "201910120000Z"

Organization: "Cisco Systems, Inc."

Contact Info: "Cisco Systems Customer Service Postal: 170 W Tasman Drive San Jose, CA 95134 USA Tel: +1 800 553-NETS"

Description: The MIB module for the Cisco Cloud Native Execution Environment (CNEE) platform. This MIB only handles notifications from the CNEE.

{ ciscoMgmt 999 }

### ciscoCneeMIBNotifs Object ID

{ ciscoCneeMIB 0 }

### ciscoCneeMIBFaults Object ID

{ ciscoCneeMIB 1 }

### ciscoCneeMIBConform Object ID

{ ciscoCneeMIB 2 }

### cneeFaultId Object Type

Syntax: Octet string of 1-64 characters.

Max Access: not-accessible

Status: current

Description: Uniquely identify the fault within a monitored entity.

{ ciscoCneeMIBFaults 1 }

### cneeFaultSource Object Type

Syntax: Octet string of 1-128 characters.

Max Access: not-accessible

Status: current

Description: Uniquely identify the monitored entity It can be a hostname or IP Address or human readable identity.

{ ciscoCneeMIBFaults 2 }

### cneeFaultSeverity Object Type

Syntax: Octet string of 1-16 characters.

Max Access: not-accessible

Status: current

Description: Indicates the level of urgency for operator attention Refer 3GPP TS32.111-5 v9.0.0 section 4.3.

{ ciscoCneeMIBFaults 3 }

### cneeFaultTime Object Type

Syntax: DateAndTime

Max Access: not-accessible

Status: current

Description: The date and time when the fault is detected.

{ ciscoCneeMIBFaults 4 }

### cneeFaultType Object Type

Syntax: Octet string of 1-64 characters.

Max Access: not-accessible

Status: current

Description: Indicates the type of fault Refer 3GPP TS32.111-5 v9.0.0 section 4.3.

{ ciscoCneeMIBFaults 5 }

### cneeFaultAdditionalInfo Object Type

Syntax: Octet string of 1-2048 characters.

Max Access: not-accessible

Status: current

Description: Additional Information about the fault.

{ ciscoCneeMIBFaults 6 }

### cneeFaultClusterName Object Type

Syntax: Octet string of 1-128 characters.

Max Access: not-accessible

Status: current

Description: The cluster name associated to the fault.

{ ciscoCneeMIBFaults 7 }

### cneeFaultNamespace Object Type

Syntax: Octet string of 1-128 characters.

Max Access: not-accessible

Status: current

Description: Identifies the namespace associated to the fault. This field is not always available for every fault.

{ ciscoCneeMIBFaults 8 }

### cneeFaultHostname Object Type

Syntax: Octet string of 1-128 characters.

Max Access: not-accessible

Status: current

Description: Identifies the hostname or ip address associated with the fault. This field is not always available for every fault.

{ ciscoCneeMIBFaults 9 }

### cneeFaultInstance Object Type

Syntax: Octet string of 1-128 characters.

Max Access: not-accessible

Status: current

Description: Identifies the instance associated to the fault. The instance is set by the alert rule creator and may not reference a host but could reference a process or KPI that is associated to the fault. This field is not always available for every fault

{ ciscoCneeMIBFaults 10 }

### cneeVnfAlias Object Type

Syntax: Octet string of 1-128 characters.

Max Access: not-accessible

Status: current

Description: Alias for the monitored entity

{ ciscoCneeMIBFaults 11 }

### cneeFaultActiveNotif Notification Type

Objects: cneeFaultId, cneeFaultSource, cneeFaultSeverity, cneeFaultTime, cneeFaultType, cneeFaultAdditionalInfo, cneeFaultClusterName, cneeFaultNamespace, cneeFaultHostname, cneeFaultInstance, cneeVnfAlias

Status: current

Description: This notification is generated by CNEE whenever a fault gets triggered.

{ ciscoCneeMIBNotifs 1 }

### cneeFaultClearNotif Notification Type

Objects: cneeFaultId, cneeFaultSource, cneeFaultSeverity, cneeFaultTime, cneeFaultType, cneeFaultAdditionalInfo, cneeFaultClusterName, cneeFaultNamespace, cneeFaultHostname, cneeFaultInstance, cneeVnfAlias

Status: current

Description: This notification is generated by CNEE whenever a fault gets cleared.

{ ciscoCneeMIBNotifs 2 }

### ciscoCneeMIBCompliances Object ID

{ ciscoCneeMIBConform 1 }

### ciscoCneeMIBGroups Object ID

{ ciscoCneeMIBConform 2 }

### cneeMIBCompliance Module Compliance

Status: current

Description: The compliance statement for entities that support the Cisco CNEE Managed Objects

Module: -- this module

Mandatory Groups: cneeMIBFaultGroup, cneeMIBNotificationGroup

{ ciscoCneeMIBCompliances 1 }

### cneeMIBFaultGroup Object Group

Objects: cneeFaultId, cneeFaultSource, cneeFaultSeverity, cneeFaultTime, cneeFaultType, cneeFaultAdditionalInfo, cneeFaultClusterName, cneeFaultNamespace, cneeFaultHostname, cneeFaultInstance, cneeVnfAlias

Status: current

Description: The set of CNEE Fault groups defined by this MIB

{ ciscoCneeMIBGroups 1 }

### cneeMIBNotificationGroup Notification Group

Notifications: cneeFaultActiveNotif, cneeFaultClearNotif

Status: current

Description: The set of CNEE notifications defined by this MIB

{ ciscoCneeMIBGroups 2 }

# CISCO-SMI

### ciscoProducts Object ID

Status: current

Description: ciscoProducts is the root OBJECT IDENTIFIER from which sysObjectID values are assigned. Actual values are defined in CISCO-PRODUCTS-MIB.

{ cisco 1 }

### local Object ID

Status: current

Description: Subtree beneath which pre-10.2 MIBS were built.

{ cisco 2 }

### temporary Object ID

Status: current

Description: Subtree beneath which pre-10.2 experiments were placed.

{ cisco 3 }

### pakmon Object ID

Status: current

Description: reserved for pakmon

{ cisco 4 }

### workgroup Object ID

Status: current

Description: subtree reserved for use by the Workgroup Business Unit

{ cisco 5 }

### otherEnterprises Object ID

Status: current

Description: otherEnterprises provides a root object identifier from which mibs produced by other companies may be placed. mibs produced by other enterprises are typically implemented with the object identifiers as defined in the mib, but if the mib is deemed to be uncontrolled, we may reroot the mib at this subtree in order to have a controlled version.

{ cisco 6 }

### ciscoSB Object ID

Status: current

Description: ciscoSB provides root Object Identifier for Management Information Base for products of Cisco Small Business. This includes products rebranded from linksys aquisition. MIB numbers under this root are managed and controlled by ciscosb_mib@cisco.com.

{ otherEnterprises 1 }

### ciscoSMB Object ID

Status: current

Description: ciscoSMB provides root Object Identifier for Management Information Base for products of Cisco built for Small and Medium Business market.The MIB numbers under this root are managed and controlled by ciscosmb_mib@cisco.com

{ otherEnterprises 2 }

### ciscoAgentCapability Object ID

Status: current

Description: ciscoAgentCapability provides a root object identifier from which AGENT-CAPABILITIES values may be assigned.

{ cisco 7 }

### ciscoConfig Object ID

Status: current

Description: ciscoConfig is the main subtree for configuration mibs.

{ cisco 8 }

### ciscoMgmt Object ID

Status: current

Description: ciscoMgmt is the main subtree for new mib development.

{ cisco 9 }

### ciscoExperiment Object ID

Status: current

Description: ciscoExperiment provides a root object identifier from which experimental mibs may be temporarily based. mibs are typicially based here if they fall in one of two categories 1) are IETF work-in-process mibs which have not been assigned a permanent object identifier by the IANA. 2) are cisco work-in-process which has not been assigned a permanent object identifier by the cisco assigned number authority, typicially because the mib is not ready for deployment. NOTE WELL: support for mibs in the ciscoExperiment subtree will be deleted when a permanent object identifier assignment is made.

{ cisco 10 }

### ciscoAdmin Object ID

Status: current

Description: ciscoAdmin is reserved for administratively assigned OBJECT IDENTIFIERS, i.e. those not associated with MIB objects

{ cisco 11 }

### ciscoModules Object ID

Status: current

Description: ciscoModules provides a root object identifier from which MODULE-IDENTITY values may be assigned.

{ cisco 12 }

### lightstream Object ID

Status: current

Description: subtree reserved for use by Lightstream

{ cisco 13 }

### ciscoworks Object ID

Status: current

Description: ciscoworks provides a root object identifier beneath which mibs applicable to the CiscoWorks family of network management products are defined.

{ cisco 14 }

### newport Object ID

Status: current

Description: subtree reserved for use by the former Newport Systems Solutions, now a portion of the Access Business Unit.

{ cisco 15 }

### ciscoPartnerProducts Object ID

Status: current

Description: ciscoPartnerProducts is the root OBJECT IDENTIFIER from which partner sysObjectID values may be assigned. Such sysObjectID values are composed of the ciscoPartnerProducts prefix, followed by a single identifier that is unique for each partner, followed by the value of sysObjectID of the Cisco product from which partner product is derived. Note that the chassisPartner MIB object defines the value of the identifier assigned to each partner.

{ cisco 16 }

### ciscoPolicy Object ID

Status: current

Description: ciscoPolicy is the root of the Cisco-assigned OID subtree for use with Policy Management.

{ cisco 17 }

### ciscoPIB Object ID

Status: current

Description: ciscoPIB is the root of the Cisco-assigned OID subtree for assignment to PIB (Policy Information Base) modules.

{ ciscoPolicy 2 }

### ciscoPolicyAuto Object ID

Status: current

Description: ciscoPolicyAuto is the root of the Cisco-assigned OID subtree for OIDs which are automatically assigned for use in Policy Management.

{ cisco 18 }

### ciscoPibToMib Object ID

Status: current

Description: ciscoPibToMib is the root of the Cisco-assigned OID subtree for MIBs which are algorithmically generated/translated from Cisco PIBs with OIDs assigned under the ciscoPIB subtree. These generated MIBs allow management entities (other the current Policy Server) to read the downloaded policy. By convention, for PIB 'ciscoPIB.x', the generated MIB shall have the name 'ciscoPibToMib.x'.

{ ciscoPolicyAuto 2 }

### ciscoDomains Object ID

Status: current

Description: ciscoDomains provides a root object identifier from which different transport mapping values may be assigned.

{ cisco 19 }

### ciscoCIB Object ID

Status: current

Description: ciscoCIB is the root of the Cisco-assigned OID subtree for assignment to MIB modules describing managed objects that part of the CPE automatic configuration framework.

{ cisco 20 }

### ciscoCibMmiGroup Object ID

Status: current

Description: ciscoCibMmiGroup is the root of the Cisco-assigned OID subtree for assignment to MIB modules describing managed objects supporting the Modem Management Interface (MMI), the interface that facilitates CPE automatic configuration.

{ ciscoCIB 1 }

### ciscoCibProvGroup Object ID

Status: current

Description: ciscoCibStoreGroup is the root of the Cisco-assigned OID subtree for assignment to MIB modules describing managed objects contributing to the Configuration Information Base (CIB).

{ ciscoCIB 2 }

### ciscoPKI Object ID

Status: current

Description: ciscoPKI is the root of cisco-assigned OID subtree for PKI Certificate Policies and Certificate Extensions.

{ cisco 21 }

### ciscoLDAP Object ID

Status: current

Description: ciscoLDAP is the root of the Cisco-assigned OID subtree for assignment to LDAP (Lightweight Directory Access Protocol) modules.

{ cisco 22 }

### ciscoProxy Object ID

Status: current

Description: ciscoProxy OBJECT IDENTIFIERS are used to uniquely name party mib records created to proxy for SNMPv1.

{ ciscoAdmin 1 }

### ciscoPartyProxy Object ID

{ ciscoProxy 1 }

### ciscoContextProxy Object ID

{ ciscoProxy 2 }

### ciscoRptrGroupObjectID Object ID

Status: current

Description: ciscoRptrGroupObjectID OBJECT IDENTIFIERS are used to uniquely identify groups of repeater ports for use by the SNMP-REPEATER-MIB (RFC 1516) rptrGroupObjectID object.

{ ciscoAdmin 2 }

### ciscoUnknownRptrGroup Object ID

Status: current

Description: The identity of an unknown repeater port group.

{ ciscoRptrGroupObjectID 1 }

### cisco2505RptrGroup Object ID

Status: current

Description: The authoritative identity of the Cisco 2505 repeater port group.

{ ciscoRptrGroupObjectID 2 }

### cisco2507RptrGroup Object ID

Status: current

Description: The authoritative identity of the Cisco 2507 repeater port group.

{ ciscoRptrGroupObjectID 3 }

### cisco2516RptrGroup Object ID

Status: current

Description: The authoritative identity of the Cisco 2516 repeater port group.

{ ciscoRptrGroupObjectID 4 }

### ciscoWsx5020RptrGroup Object ID

Status: current

Description: The authoritative identity of the wsx5020 repeater port group.

{ ciscoRptrGroupObjectID 5 }

### ciscoChipSets Object ID

Status: current

Description: Numerous media-specific MIBS have an object, defined as an OBJECT IDENTIFIER, which is the identity of the chipset realizing the interface. Cisco-specific chipsets have their OBJECT IDENTIFIERS assigned under this subtree.

{ ciscoAdmin 3 }

### ciscoChipSetSaint1 Object ID

Status: current

Description: The identity of the Rev 1 SAINT ethernet chipset manufactured for cisco by LSI Logic.

{ ciscoChipSets 1 }

### ciscoChipSetSaint2 Object ID

Status: current

Description: The identity of the Rev 2 SAINT ethernet chipset manufactured for cisco by LSI Logic.

{ ciscoChipSets 2 }

### ciscoChipSetSaint3 Object ID

Status: current

Description: The identity of the Rev 3 SAINT ethernet chipset manufactured for cisco by Plessey.

{ ciscoChipSets 3 }

### ciscoChipSetSaint4 Object ID

Status: current

Description: The identity of the Rev 4 SAINT ethernet chipset manufactured for cisco by Mitsubishi.

{ ciscoChipSets 4 }

### ciscoTDomains Object ID

{ ciscoDomains 99999 }

### ciscoTDomainUdpIpv4 Object ID

Status: current

Description: The UDP over IPv4 transport domain. The corresponding transport address is of type CiscoTAddressIPv4.

{ ciscoTDomains 1 }

### ciscoTDomainUdpIpv6 Object ID

Status: current

Description: The UDP over IPv6 transport domain. The corresponding transport address is of type CiscoTAddressIPv6 for global IPv6 addresses and CiscoTAddressIPv6s for scoped IPv6 addresses.

{ ciscoTDomains 2 }

### ciscoTDomainTcpIpv4 Object ID

Status: current

Description: The TCP over IPv4 transport domain. The corresponding transport address is of type CiscoTAddressIPv4.

{ ciscoTDomains 3 }

### ciscoTDomainTcpIpv6 Object ID

Status: current

Description: The TCP over IPv6 transport domain. The corresponding transport address is of type CiscoTAddressIPv6 for global IPv6 addresses and CiscoTAddressIPv6s for scoped IPv6 addresses.

{ ciscoTDomains 4 }

### ciscoTDomainLocal Object ID

Status: current

Description: The Posix Local IPC transport domain. The corresponding transport address is of type CiscoTAddressLocal. The Posix Local IPC transport domain incorporates the well known UNIX domain sockets.

{ ciscoTDomains 5 }

### ciscoTDomainClns Object ID

Status: current

Description: The CLNS transport domain. The corresponding transport address is of type CiscoTAddressOSI.

{ ciscoTDomains 6 }

### ciscoTDomainCons Object ID

Status: current

Description: The CONS transport domain. The corresponding transport address is of type CiscoTAddressOSI.

{ ciscoTDomains 7 }

### ciscoTDomainDdp Object ID

Status: current

Description: The DDP transport domain. The corresponding transport address is of type CiscoTAddressNBP.

{ ciscoTDomains 8 }

### ciscoTDomainIpx Object ID

Status: current

Description: The IPX transport domain. The corresponding transport address is of type CiscoTAddressIPX.

{ ciscoTDomains 9 }

### ciscoTDomainSctpIpv4 Object ID

Status: current

Description: The SCTP over IPv4 transport domain. The corresponding transport address is of type CiscoTAddressIPv4.

Reference: RFC 2960 - Stream Control Transmission Protocol. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson. October 2000.

{ ciscoTDomains 10 }

### ciscoTDomainSctpIpv6 Object ID

Status: current

Description: The SCTP over IPv6 transport domain. The corresponding transport address is of type CiscoTAddressIPv6 for global IPv6 addresses and CiscoTAddressIPv6s for scoped IPv6 addresses.

Reference: RFC 2960 - Stream Control Transmission Protocol. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson. October 2000.

{ ciscoTDomains 11 }