Monitoring Support

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

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

Revision History

Table 2: Revision History

Revision Details	Release
The **monitor subscriber** CLI command includes the **imei** keyword to support subscriber IMEI.	2021.02.0
Added show commands for the RPC feature in App-Infra SMF.	2020.03.0
First introduced.	Pre-2020.02.0

Monitor Subscriber and Monitor Protocol

Feature Description

The SMF supports the Monitor Subscriber and Monitor Protocol on the Kubernetes environment. This feature allows to capture messages of subscribers and protocols.

Converged Core Refactoring Changes

This section describes the changes related to converged core refactoring in this chapter.

In release 2021.01: The **monitor subscriber supi** CLI command is enhanced to add the **namespace** option to display the output pertaining to the respective namespace.

In 2021.02 and later releases, the **namespace** keyword is deprecated and replaced with the **nf-service** keyword.

Configuring the Monitor Subscriber and Monitor Protocol Feature

Monitoring the Subscriber

To monitor the subscriber in the SMF, use the following CLI command:

```
monitor subscriber [ capture-duration duration | gr-instance gr_instance_id
| imei imei_id | imsi imsi_value | internal-messages [ yes ] | namespace [
sgw | smf ] | nf-service [ sgw | smf ] | supi supi_id | transaction-logs [
 yes ] ]
```

NOTES:

- **capture-duration** *duration*: Specify the duration in seconds during which monitor subscriber is enabled. The default value is 300 seconds (5 minutes). This is an optional parameter.
- **gr-instance** *gr_instance_id*: Specify the GR instance ID. The instance ID 1 denotes the local instance ID.
- **imei** *imei_id*: Specify the subscriber IMEI. For example: 123456789012345, *
- **imsi** *imsi_value*: Specify the subscriber IMSI. For example: 123456789, *
- **internal-messages [yes]**: Enable internal messages when set to **yes**. By default, it is disabled. This is an optional parameter.
- **namespace [sgw | smf]**: Enable the specified namespace. By default, namespace is set to none. This is an optional parameter.

  Important This keyword is deprecated in release 2021.02.0 and replaced with **nf-service** keyword.

- **nf-service [sgw | smf]** : Enable the specified NF service. By default, nf-service is set to none. This is an optional parameter.

Important The **nf-service** keyword replaces the **namespace** keyword in release 2021.02 and beyond.

- **supi** *supi_id*: Specify the subscriber identifier. For example: imsi-123456789, imsi-123*
- **transaction-logs [yes]**: Enable transaction logs when set to **yes**. By default, it is disabled. This is an optional parameter.

The **monitor subscriber** CLI command can be run simultaneously on multiple terminals. For example, run the CLI simultaneously in two SMF Ops Center terminals for two subscribers (for example, imsi-123456789012345 and imsi-456780123456789) to implement the following:

- Monitor the duration when the monitor subscriber is enabled
- View internal messages for the specified subscriber
- View transaction logs for the specified subscriber

Terminal 1: The following command monitors and displays subscriber messages for the specified subscriber.

```
monitor subscriber supi imsi-123456789012345 capture-duration 1000 internal-messages yes
```

Terminal 2: The following command monitors and displays transaction logs for the specified subscriber.

```
monitor subscriber supi imsi-456780123456789 capture-duration 500 internal-messages yes
transaction-logs yes
```

After the capture duration is completed, stop the CLI by using the **Ctrl+C** keys. The captured messages are reordered and stored in a file. To retrieve the list of stored files, use the **monitor subscriber list** CLI command.

For example:

```
monitor subscriber list
RELEASE_NAMESPACE: 'smf'
'monsublogs/subscriberID_imsi-*_AT_2019-10-22T09:19:05.586237087.txt.sorted'
monsublogs/subscriberID_imsi-123456789012345_AT_2019-10-22T09:20:11.122225534.txt.sorted
```

Enabling or Disabling the Transaction Messages

To enable or disable the presence of request response messages in the transaction logs, use the following sample configuration:

```
config
  logging transaction message { disable | enable }
  commit
end
```

NOTES:

- **logging transaction message { disable | enable }**: Specify whether to enable or disable messages in transaction logging.

Viewing the Sorted File on SMF Ops Center

To view the sorted file on the SMF Ops Center screen, use the following CLI command:

```
monitor subscriber dump filename filename
```

For example:

```
monitor subscriber dump filename
monsublogs/subscriberID_imsi-123456789012345_AT_2019-10-22T09:20:11.122225534.txt.sorted
```

Monitoring the Interface Protocol

To monitor the interface protocol on the SMF, use the following CLI command:

```
monitor protocol interface endpoint_name [ capture-duration duration |
gr-instance gr_instance_id ]
```

NOTES:

- **interface** *endpoint_name*: Specify the endpoint name on which PCAP is captured. This CLI allows the configuration of multiple endpoint names in a single CLI command.
- **capture-duration** *duration*: Specify the duration in seconds during which pcap is captured. The default is 300 seconds (5 minutes).
- The configured endpoint names can be retrieved using the **show endpoint** CLI command.
- **gr-instance** *gr_instance_id*: Specify the GR instance ID. The instance ID 1 denotes the local instance ID.

The **monitor protocol** CLI can be run simultaneously on multiple terminals. Also, the **interface** *endpoint_name* CLI allows the configuration of multiple endpoint names in a single CLI command. For example:

```
monitor protocol interface sbi,N4:10.86.73.161:8805,gtpc capture-duration
 1000
```

Viewing Transaction History Logs

Use the following CLI command to view the transaction history on an OAM pod shell. On another terminal, use the **kubectl** command to tail the logs of the OAM pod and then run the following CLI from the Ops Center.

```
dump transactionhistory
```

Note The most recent transaction logs are stored in a circular queue of size 1024 transaction logs.

Sample Transaction Log

The following is an example of transaction log collected in Monitor Subscriber during SMF PDU session establishment.

```
Transaction Log received from Instance: smf.smf-rest-ep.unknown.smf.0
************* TRANSACTION: 00010 *************
TRANSACTION SUCCESS:
    Txn Type             : N10RegistrationRequest(33)
    Priority             : 1
    Session State        : No_Session
LOG MESSAGES:
    2020/03/03 05:31:39.345 [DEBUG] [infra.transaction.core] Processing transaction Id: 10
 Type: 33 SubscriberID: imsi-123456789012345 Keys: []
    2020/03/03 05:31:39.345 [DEBUG] [infra.transaction.core] Trace is disabled
    2020/03/03 05:31:39.346 [TRACE] [infra.message_log.core] >>>>>>>
IPC message
Name: N10RegistrationRequest
```

```
MessageType: N10RegistrationRequest
Key:
--body--
{"regInfo":{"ueId":"imsi-123456789012345","pduSessionId":5},"regReq":{"dnn":"intershat",

"pduSessionId":5,"pgwFqdn":"cisco.com.apn.epc.mnc456.mcc123","plmnId":{"mcc":"123","mnc":"456"},


"smfInstanceId":"c388eec5-e2ff-4bda-8154-b5dd9f10ad97","supportedFeatures":"0","singleNssai":{"sd":"Abf123","sst":2}},


"msgReq":{"Type":2,"ServiceName":4,"Versions":["v1"],"ProfileName":"UP1","FailureProfile":"FH1","SvcMsgType":3,

  "Filter":{"Bitmapfeilds":2,"Dnn":"intershat"}}}
   2020/03/03 05:31:39.346 [DEBUG] [nrfClient.Discovery.nrf] Message send Metadata [Type:UDM
 ServiceName:nudm-uecm
      Method:"Register"
Payload:"\022\030\n\024imsi-123456789012345\020\005\032o\n\tintershat\030\005\"\037cisco.com.apn.epc.mnc456.


mcc123*\n\n\003123\022\0034562$c388eec5-e2ff-4bda-8154-b5dd9f10ad97:\0010B\n\n\006Abf123\020\002\

      "%\010\002\030\004J\002v1b\
     003UP1j\r\010\002\022\tintershat\272\001\003FH1\300\001\003" Versions:"v1" MsgType:33
 ProfileName:"UP1" FailureProfile:"FH1"
      SvcMsgType:UdmUecmRegisterSMF Filter:<Bitmapfeilds:2 Dnn:"intershat" > ] client
locality [] Preferred locality [], Geo Locality []
    2020/03/03 05:31:39.347 [DEBUG] [nrfClient.generic.Int] GetAvailableSLAInMS returned
timeoutInMS [0]
    2020/03/03 05:31:39.347 [DEBUG] [infra.transaction.core] Requested host Setname:  Name:
 http://10.86.73.209:9020/nudm-uecm/v1 Version:  ApiRoot:
    2020/03/03 05:31:39.347 [DEBUG] [infra.transaction.core] Exact match found. Selected
remote host is Id 2
      Name: http://10.86.73.209:9020/nudm-uecm/v1 Setname:  Host:  Port: 0 Url:
http://10.86.73.209:9020/nudm-uecm/v1
    2020/03/03 05:31:39.347 [INFO] [infra.transaction.core] Calling RPC UDM on host
http://10.86.73.209:9020/nudm-uecm/v1 proc-name UDM proc-method: Register
    2020/03/03 05:31:39.348 [DEBUG] [rest_ep.app.n10] Sending registration request to udm:
 /imsi-123456789012345/registrations/
      smf-registrations/5 with payload
{"dnn":"intershat","pduSessionId":5,"pgwFqdn":"cisco.com.apn.epc.mnc456.mcc123",

"plmnId":{"mcc":"123","mnc":"456"},"singleNssai":{"sd":"Abf123","sst":2},"smfInstanceId":"c388eec5-e2ff-4bda-8154-b5dd9f10ad97",

    "supportedFeatures":"0"}
    2020/03/03 05:31:39.348 [DEBUG] [infra.rest_client.core] Sending rest mesasge to
http://10.86.73.209:9020/nudm-uecm/v1/imsi-123456789012345/registrations/smf-registrations/5

    2020/03/03 05:31:39.348 [TRACE] [infra.message_log.core] <<<<<<<<
Request
Name: UdmRegistrationRequest
Host:
http://10.86.73.209:9020/nudm-uecm/v1/imsi-123456789012345/registrations/smf-registrations/5
Method: PUT
RequestURI:
--- Headers ---
Content-Type: application/json
Body:{"dnn":"intershat","pduSessionId":5,"pgwFqdn":"cisco.com.apn.epc.mnc456.mcc123",
"plmnId":{"mcc":"123","mnc":"456"},
"singleNssai":{"sd":"Abf123","sst":2},"smfInstanceId":"c388eec5-e2ff-4bda-8154-b5dd9f10ad97","supportedFeatures":"0"}

    2020/03/03 05:31:39.376 [TRACE] [infra.message_log.core] >>>>>>>
Response
Name:
```

```
Response Status 201
--- Headers ---
Location:
http://10.86.73.209:9020/nudm-uecm/v1/imsi-123456789012345/registrations/smf-registrations/5
Content-Length: 225
Content-Type: application/json
Body:{"pgwFqdn": "cisco.com.apn.epc.mnc456.mcc123", "plmnId": {"mcc": "123", "mnc": "456"},
 "dnn": "intershat",
  "smfInstanceId": "524f5f8a-b584-47b8-86f5-a5292eabcdef", "pduSessionId": 5, "singleNssai":
 {"sd": "Abf123", "sst": 2}}
    2020/03/03 05:31:39.377 [INFO] [rest_ep.app.n10] Received registration success response
 with status = 201 and body =
     {"pgwFqdn": "cisco.com.apn.epc.mnc456.mcc123", "plmnId": {"mcc": "123", "mnc": "456"},
 "dnn": "intershat",
      "smfInstanceId": "524f5f8a-b584-47b8-86f5-a5292eabcdef", "pduSessionId": 5,
"singleNssai": {"sd": "Abf123", "sst": 2}}
    2020/03/03 05:31:39.377 [DEBUG] [infra.transaction.core] Last stage ( init_done ) ->
Next stage ( finished )
    2020/03/03 05:31:39.378 [TRACE] [infra.message_log.core] <<<<<<<<
IPC message
Name: N10RegistrationSuccess
MessageType: N10RegistrationSuccess
Key:
--body--
{"regRes":{"dnn":"intershat","pduSessionId":5,"pgwFqdn":"cisco.com.apn.epc.mnc456.mcc123",
"plmnId":{"mcc":"123","mnc":"456"},
  "smfInstanceId":"c388eec5-e2ff-4bda-8154-b5dd9f10ad97","supportedFeatures":"0",
  "singleNssai":{"sd":"Abf123","sst":2}},
  "msgRsp":{"Type":2,"Http2_Status":201,"MsgType":34,"ServiceName":4,"SelectedVersion":"v1",

  "SelectedEndPoint":

{"ipv4Address":"10.86.73.209","port":9020,"transport":1},"SelectedProfileName":"UP1","SelectedEndPointName":"EP1",

  "SelectedEndPointProfile":"EP1","SelectedLocality":"LOC1","FailureProfile":"FH1",
  "GroupID":"UDM-dnn=intershat;"}}
    2020/03/03 05:31:39.378 [DEBUG] [infra.transaction.core] no response message sent for
10
***********************************************
Transaction Log received from Instance: smf.smf-rest-ep.unknown.smf.0
************* TRANSACTION: 00011 *************
TRANSACTION SUCCESS:
    Txn Type             : N10SubscriptionFetchReq(36)
    Priority             : 1
    Session State        : No_Session
LOG MESSAGES:
    2020/03/03 05:31:39.384 [DEBUG] [infra.transaction.core] Processing transaction Id: 11
 Type: 36 SubscriberID: imsi-123456789012345 Keys: []
    2020/03/03 05:31:39.384 [DEBUG] [infra.transaction.core] Trace is disabled
    2020/03/03 05:31:39.384 [TRACE] [infra.message_log.core] >>>>>>>
IPC message
Name: N10SubscriptionFetchReq
MessageType: N10SubscriptionFetchReq
Key:
--body--
{"subInfo":{"ueId":"imsi-123456789012345"},"uriParams":{"supportedFeatures":"0",
"singleNssai":{"sd":"Abf123","sst":2},
  "dnn":"intershat","plmnId":{"mcc":"123","mnc":"456"}},"msgReq":{"Type":2,"ServiceName":3,

  "Versions":["v1"],"ProfileName":
  "UP1","FailureProfile":"FH1","SvcMsgType":1,"Filter":{"Bitmapfeilds":2,"Dnn":"intershat"}}}

   2020/03/03 05:31:39.384 [DEBUG] [nrfClient.Discovery.nrf] Message send Metadata [Type:UDM
 ServiceName:nudm-sdm Method:"Subscription"
Payload:"\022\026\n\024imsi-123456789012345\032&\n\0010\022\n\n\006Abf123\020\002\032\tintershat\

     "\n\n\003123\022\003456
     \"%\010\002\030\003J\002v1b\003UP1j\r\010\002\022\tintershat\272\001\003FH1\300\001\001"
 Versions:"v1" MsgType:36
      ProfileName:"UP1" FailureProfile:"FH1" SvcMsgType:UdmSdmGetUESMSubscriptionData
Filter:<Bitmapfeilds:2 Dnn:"intershat" > ]
      client locality [] Preferred locality [], Geo Locality []
    2020/03/03 05:31:39.385 [DEBUG] [nrfClient.generic.Int] GetAvailableSLAInMS returned
timeoutInMS [0]
    2020/03/03 05:31:39.385 [DEBUG] [infra.transaction.core] Requested host Setname:  Name:
 http://10.86.73.209:9020/nudm-sdm/v1 Version:  ApiRoot:
    2020/03/03 05:31:39.385 [DEBUG] [infra.transaction.core] Exact match found. Selected
remote host is Id 1 Name: http://10.86.73.209:9020/
      nudm-sdm/v1 Setname:  Host:  Port: 0 Url: http://10.86.73.209:9020/nudm-sdm/v1
    2020/03/03 05:31:39.385 [INFO] [infra.transaction.core] Calling RPC UDM on host
http://10.86.73.209:9020/nudm-sdm/v1 proc-name
      UDM proc-method: Subscription
    2020/03/03 05:31:39.385 [DEBUG] [rest_ep.app.n10] Sending sm subscription request to
udm: /imsi-123456789012345/sm-data?
      dnn=intershat&plmn-id=%7B%22mcc%22%3A%22123%22%2C%22mnc%22%3A%22456%22%7D
      &single-nssai=%7B%22sd%22%3A%22Abf123%22%2C%22sst%22%3A2%
      7D&supported-features=0
    2020/03/03 05:31:39.385 [DEBUG] [infra.rest_client.core] Sending rest mesasge to
http://10.86.73.209:9020/nudm-sdm/v1/

imsi-123456789012345/sm-data?dnn=intershat&plmn-id=%7B%22mcc%22%3A%22123%22%2C%22mnc%22%3A%22456%22%7D

     &single-nssai=%7B%22sd%22
      %3A%22Abf123%22%2C%22sst%22%3A2%7D&supported-features=0
    2020/03/03 05:31:39.385 [TRACE] [infra.message_log.core] <<<<<<<<
Request
Name: UdmSubscriptionRequest
Host:
http://10.86.73.209:9020/nudm-sdm/v1/imsi-123456789012345/sm-data?dnn=intershat&plmn-id=%7B%22mcc%22%3A%22123%22%2C%22mnc%22%3A%2

   2456%22%7D&single-nssai=%7B%22sd%22%3A%22Abf123%22%2C%22sst%22%3A2%7D&supported-features=0
Method: GET
RequestURI:
--- Headers ---
Content-Type: application/json
    2020/03/03 05:31:39.400 [TRACE] [infra.message_log.core] >>>>>>>
Response
Name:
Response Status 200
--- Headers ---
Content-Length: 812
Content-Type: application/json
Body:[{"sharedDnnConfigurationsIds": "012345", "internalGroupIds": ["1aAbB00866564",
"dhsjdha67673AA"], "singleNssai": {"sd": "Abf123",
  "sst": 2}, "dnnConfigurations": {"intershat": {"3gppChargingCharacteristics": "1",
"5gQosProfile": {"5qi": 5, "arp": {"preemptVuln":
  "NOT_PREEMPTABLE", "preemptCap": "NOT_PREEMPT", "priorityLevel": 15}, "priorityLevel":
1}, "sscModes": {"allowedSscModes": ["SSC_MODE_1",
  "SSC_MODE_2"], "defaultSscMode": "SSC_MODE_1"}, "ladnIndicator": true, "pduSessionTypes":
 {"defaultSessionType": "IPV4",
  "allowedSessionTypes": ["IPV6", "IPV4V6", "UNSTRUCTURED", "ETHERNET"]}, "staticIpAddress":
 [{"ipv4Addr": "1.1.1.1"}, {"ipv6Addr": "::1"}],
  "upSecurity": {"upIntegr": "REQUIRED", "upConfid": "PREFERRED"}, "sessionAmbr": {"downlink":
 "125 Mbps", "uplink": "100 Mbps"},
  "iwkEpsInd": false}}}]
    2020/03/03 05:31:39.400 [INFO] [rest_ep.app.n10] Received sm subscription success
response with status = 200 and body =
      [{"sharedDnnConfigurationsIds": "012345", "internalGroupIds": ["1aAbB00866564",
"dhsjdha67673AA"], "singleNssai": {"sd":
     "Abf123", "sst": 2}, "dnnConfigurations": {"intershat": {"3gppChargingCharacteristics":
 "1", "5gQosProfile": {"5qi": 5, "arp":
     {"preemptVuln": "NOT_PREEMPTABLE", "preemptCap": "NOT_PREEMPT", "priorityLevel": 15},
 "priorityLevel": 1}, "sscModes":
      {"allowedSscModes": ["SSC_MODE_1", "SSC_MODE_2"], "defaultSscMode": "SSC_MODE_1"},
"ladnIndicator": true, "pduSessionTypes":
     {"defaultSessionType": "IPV4", "allowedSessionTypes": ["IPV6", "IPV4V6", "UNSTRUCTURED",
 "ETHERNET"]}, "staticIpAddress":
     [{"ipv4Addr": "1.1.1.1"}, {"ipv6Addr": "::1"}], "upSecurity": {"upIntegr": "REQUIRED",
 "upConfid": "PREFERRED"}, "sessionAmbr":
      {"downlink": "125 Mbps", "uplink": "100 Mbps"}, "iwkEpsInd": false}}}]
    2020/03/03 05:31:39.401 [INFO] [rest_ep.app.n10] Unmarshalled smData:
smData:<dnnConfiguration:<key:"intershat" value:
      <_3gppChargingCharacteristics:"1" ladnIndicator:true
pduSessionTypes:<allowedSessionTypes:IPV6 allowedSessionTypes:IPV4V6
     allowedSessionTypes:UNSTRUCTURED allowedSessionTypes:ETHERNET defaultSessionType:IPV4
 > sessionAmbr:<downlink:"125 Mbps"
      uplink:"100 Mbps" > sscModes:<allowedSscModes:SSC_MODE_1 allowedSscModes:SSC_MODE_2
defaultSscMode:SSC_MODE_1 >
      upSecurity:<upConfid:PREFERRED > subsQosProfile:<_5qi:5 arp:<preemptCap:NOT_PREEMPT
preemptVuln:NOT_PREEMPTABLE priorityLevel:15 >
     priorityLevel:1 > staticIpAddress:<ipv4Addr:"1.1.1.1" > staticIpAddress:<ipv6Addr:"::1"
 > > > singleNssai:<sd:"Abf123" sst:2 >
      internalGroupIds:"1aAbB00866564" internalGroupIds:"dhsjdha67673AA"
sharedDnnConfigurationsIds:"012345" >
    2020/03/03 05:31:39.402 [DEBUG] [infra.transaction.core] Last stage ( init_done ) ->
Next stage ( finished )
    2020/03/03 05:31:39.403 [TRACE] [infra.message_log.core] <<<<<<<<
IPC message
Name: N10SubscriptionFetchSuccess
MessageType: N10SubscriptionFetchSuccess
Key:
--body--
{"subInfo":{"ueId":"imsi-123456789012345"},"uriParams":{"supportedFeatures":"0","singleNssai":{"sd":"Abf123","sst":2},"dnn":"intershat",

"plmnId":{"mcc":"123","mnc":"456"}},"smData":[{"dnnConfiguration":{"intershat":{"_3gppChargingCharacteristics":"1","ladnIndicator":true,

"pduSessionTypes":{"allowedSessionTypes":[2,3,4,5],"defaultSessionType":1},"sessionAmbr":{"downlink":"125
 Mbps","uplink":"100 Mbps"},

"sscModes":{"allowedSscModes":[1,2],"defaultSscMode":1},"upSecurity":{"upConfid":2},"subsQosProfile":{"_5qi":5,"arp":{"preemptCap":1,

"preemptVuln":1,"priorityLevel":15},"priorityLevel":1},"staticIpAddress":[{"ipv4Addr":"1.1.1.1"},{"ipv6Addr":"::1"}]}},"singleNssai":

{"sd":"Abf123","sst":2},"internalGroupIds":["1aAbB00866564","dhsjdha67673AA"],"sharedDnnConfigurationsIds":"012345"}],"msgRsp":

{"Type":2,"Http2_Status":200,"MsgType":37,"ServiceName":3,"SelectedVersion":"v1","SelectedEndPoint":{"ipv4Address":"10.86.73.209",

"port":9020,"transport":1},"SelectedProfileName":"UP1","SelectedEndPointName":"EP1","SelectedEndPointProfile":"EP1","SelectedLocality":

  "LOC1","FailureProfile":"FH1","GroupID":"UDM-dnn=intershat;"}}
    2020/03/03 05:31:39.403 [DEBUG] [infra.transaction.core] no response message sent for
11
***********************************************
*
```

RPC Remote Host Management

Feature Description

Show RPC feature in App-Infra provides RPCs to connect to remote NFs. These RPCs are managed in App-Infra. This CLI shows the information about the RPC and RPC remote host information in tabular format.

RPC Remote Host Management OAM Support

This section provides information regarding show commands available to monitor and troubleshoot the SMF RPC Remote Host Management.

show rpc

The **show rpc** command lists all the RPCs from all the pods with RPC and remote host information.

- POD INSTANCE - Displays the instance info of the pod
- NAME - Displays the name of the RPC registered in pod
- REMOTE ADDRESS - Displays IP address and port of remote endpoint
- REMOTE HOST - Displays the name of the RPC host
- TYPE - Displays the type of connection such as Rest, Grpc, GrpcStream, and so on
- SET NAME - Displays the RPC set name for group of RPC hosts
- STATUS - Displays the current state of the RPC host. The possible status values are Started, Starting, and Stopped
- CONNECTED TIME - Displays the time when the RPC host was connected
- DISCONNECTED TIME - Displays the time when the RPC host was disconnected
- MONITOR RPC HOST - Indicates whether the RPC host is being monitored for connection status
- PROCESSING INSTANCE INFO - Indicates the processing instance name, if available
- VERSION - Displays the version of the RPC host API, if available