|  |
| --- |
|  |
| UNMP |
| CCPL |

|  |
| --- |
| Codescape Consultants PL  [5/3/2011] |

# Product Name: NMS

Feature: NMS TCP Discovery Server

Requested By: Vivek Bansal

Reviewed By: Peeyush Raj

Implemented By: Rahul Gautam

Verified By: Utkarsh Jain

SW release version in which Feature included:

## Revision history (in case multiple revisions)

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | By | Description |
| 0.01 | 7/11/2011 | Rahul Gautam | Implementation Detailing |
| 0.0.2 | 7/16/2011 | Peeyush Raj | Verification and Restructuring |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 

# Feature Description: TCP Unicast Discovery

*In a practice network scenario, multicast or broadcast discovery would degrade the network’s performance, to rectify the degrade issue TCP based unicast discovery, where element would initialize the request to NMS server for configuration details.*

## Supported System Requirements

* Linux System, preferably Ubuntu/Debian

## System Use Cases

## Device Discovered and successfully added to the NMS

## Device Discovered, but connection breaks down at NMS Server

## Device Discovered, but connection breaks down at Device Level

## Device not able to connect to the NMS server

### Device Discovered and successfully added to NMS

The NMS server is up and accepting the incoming connection from the device, all the configuration details are set and properly ACKd to the NMS server by the device. The NMS server would then add the device in the monitored device list, and a unique NeId is generated from the NMS, would be maintained at NMS and the Device throughout the device lifetime in the network. Device then would be shown to the user to get and set the configuration of the device.

### Device Discovered, but connection breaks down at NMS server

Device would have to wait indefinitely in this case, and as soon as the connection with NMS server is made, device should restart the configuration process.

Alert should be generated as NMS TCP server busy.

### Device Discovered, but the connection breaks down at Device Level

Whenever device comes up, within a pre-defined time, the device would then re-initiate the registration process, NMS should respond to the request and the process would continue till the device is successfully added to the NMS, if device doesn’t comes up within a pre-defined duration, alert should be generated from NMS.

Alert should be of class Device Down while registration.

.

### Device is not able to connect to the NMS server.

Device would have to wait indefinitely in this case, and as soon as the connection with NMS server is made, device should restart the configuration process.

Alert should be generated as NMS TCP server busy.

# Feature Design Description

## TCP Discovery Server

* Discovery Server is a keep alive server
* Multithreaded (each connection has its own thread for communication).

## Database as configuration source

* NMS database would handle the configuration details of the device
* Database should provide unique NeId for each newly discovered device
* NMS would maintain the NeId of previously discovered device, same NeId should be provided back to device in case device state changes from down to up
* NMS database would handle the configuration details of the device
* NMS should maintain the data pre-configured in the device.

## Notification System

* When device is discovered
* When device changes state
* When NMS discovery server is busy/not responding
* If NMS server is sending corrupted data
* Connection resets between device and NMS server.

# Implementation Design Description

## Registration and Configuration

* Whenever OMC is up and NE comes up from down state, NE initiates Registration Procedure.
* Also if NE is up and OMC comes up from down state, registration is initiated from NE.

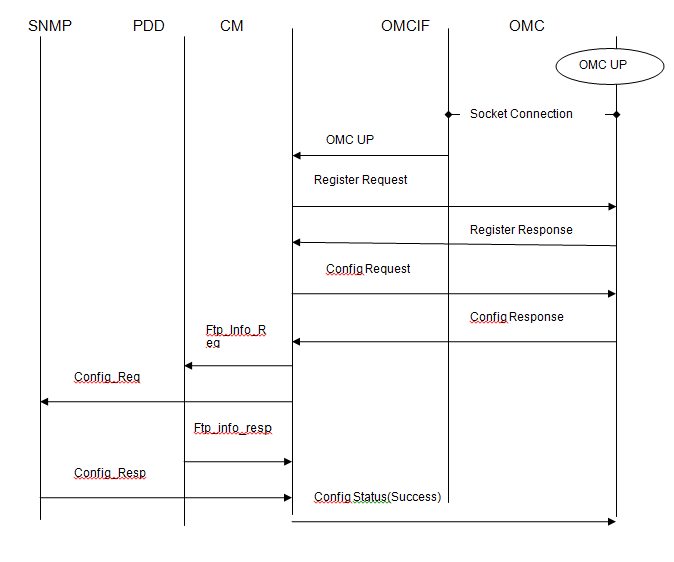
NE or node registration is an automated process that involves the following steps:



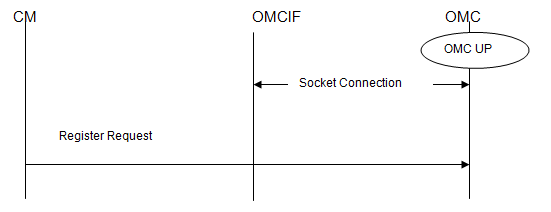
**The Registration Process is governed by following scenarios:**

### a) NE Registers and Configures successfully with OMC.

If NE is able to successfully establish the socket connection with TCP server then it can be assumed that OMC is up and running.



### NE sends Registration Message to OMC, and OMC does not respond immediately or hangs due to an operation



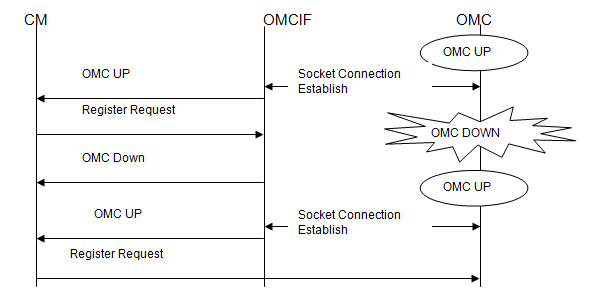
#### Expected Behavior at NE

##### To wait indefinitely for the response of this message or wait till timeout value is reached which is approximately 2 minutes.

#### Expected Behavior at OMC

##### To send the response whenever it is able to. Or in case if that is not possible, it is at the discretion of Operator to restart OMC.

### NE sends Registration Message, and OMC/TCP Link goes down i.e. it can be like OMC machine goes down which will automatically cause TCP server to be unavailable or it can be like TCP server itself goes down but OMC machine is still up.

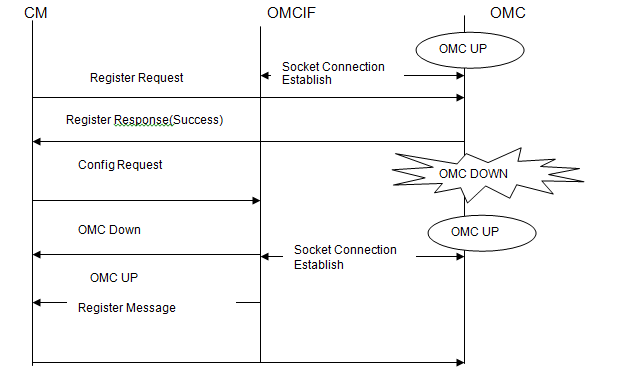


#### Expected Behavior at NE

##### To send Register Request after socket connection is established with OMC. OMCIF informs the CM about the TCP Connection status, so that CM can take the stipulated action for the same. OMC interface (OMCIF) is a software process that runs on the node side which conceptually acts as a router between node and OMC.

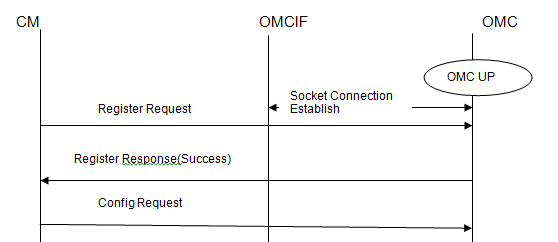
##### To respond to the Register Request Message

### NE registers with OMC successfully and OMC goes down



Expected Behavior at NE is to retry the Registration Process with OMC when OMC comes up.

### NE registers with OMC successfully and sends Config Request to OMC. OMC does not respond to that message.



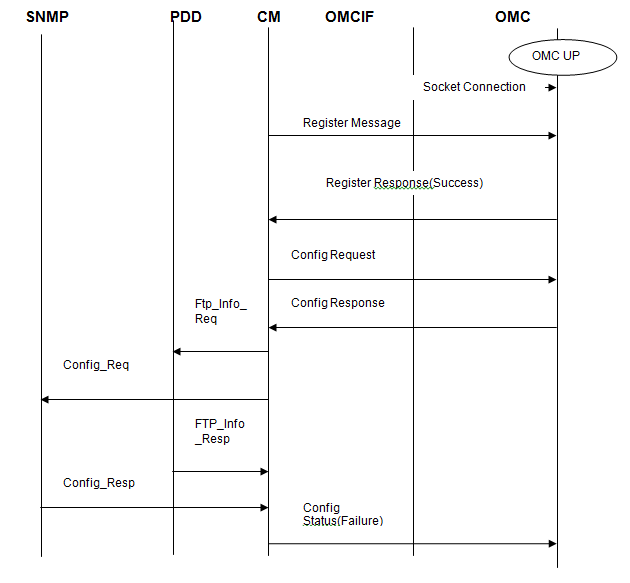
#### Expected Behavior at NE

##### To wait indefinitely for the response of this message or wait till timeout value is reached which is approximately 2 minutes.

#### Expected Behavior at OMC

##### To send the Config Response whenever it is able to. Or in case if that is not possible, it is discretion at operator end to restart the OMC.

### NE registers with OMC successfully and NE sends Config Nack to OMC due to wrong SNMP/FTP Details



#### Expected Behavior at NE

##### NE does a null check on SNMP Communities received from OMC.

##### NE does a bind on the Port received.

##### NE tries to connect to FTP Server running at OMC.

##### NE, after connecting to FTP Server, tries to login to that FTP Server with the specified username and password.

##### In case of Failure of any of these, NE sends NACK to OMC mentioning the specific error code in the message. Only that error code is sent whichever is giving NACK first.

#### Expected Behavior at OMC

##### In case of any critical error may need to restart the OMC

##### Configuration changes at NE should happen every time whenever Config Response is received from OMC. This is defined by ftp connect, ftp login check (in PDD) ,Port Binding (SNMP) , Community validation(SNMP) with SNMP/FTP Info received.

##### Registration of an NE towards OMC should start only when node has come up completely i.e. active state of card has been reached.

## TCP/IP MESSAGE FORMAT

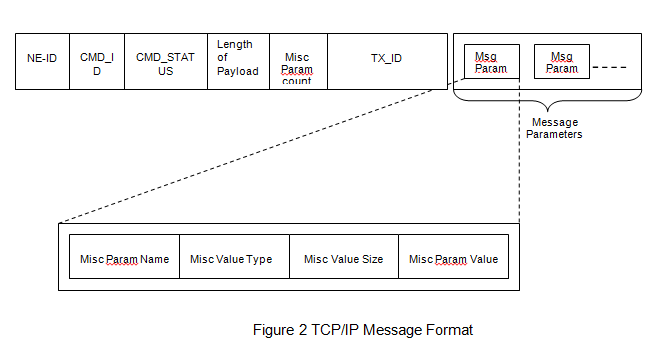
The message format is a generalized messaged format that can be understood by both OMC and the IMR-NE.

Figure 2 TCP/IP Message Format

### Header

Total Header Size  16 Bytes

NE-ID  Long (4 bytes)

CMD-ID  Long (4 bytes)

CMD\_Status  Long (4 bytes)

Length of Payload -> short (2 bytes)

Misc Param Count -> Char (1 byte)

Tx-ID -> Char (1 byte) (Min Value =0 and Max Value = 127)

The above figure shows a message format for TCP/IP communication between the IMR-OMC and IMR-NE and vice versa. The various fields shown above are described as follows:

* + - NE-ID: The NE-ID is the unique identifier that shall be set by OMC in configuration response message from the IMR-NE. This unique identifier shall remain the same through-out the life of the NE until the node is deleted from network.
    - CMD ID: This field specifies which type of message being exchanged between NE and OMC like Registration Request, Configuration Request etc. For each type of message there is a unique id defined which is called as CMD ID.
  + Command IDs are shared between OMC and NE.
    - CMD\_STATUS: This specifies the status the command whether success or failure. CMD\_SUCCESS (i.e. 0) means success and
  + CMD\_FAILURE (i.e. 1) means failure.
  + Note: For Request Messages, this field has no significance, but it will be set as CMD\_SUCCESS as a general convention.
    - TX-ID: This is the transaction ID, which is set to a new value whenever a new request message is sent to the peer. This has to be locally generated for a request type message by each entity and the corresponding peer will fill the same in the response.
  + **Note**: NE is not maintaining TX-ID when node comes up after reboot or shutdown. While OMC is maintainng TX-ID even if it comes up after reboot or shutdown.
  + Transaction IDs are locally generated by each peer entity (NE or OMC). Transaction ID is maintained only at the generating side.
  + The peer entity receiving the message with TxnId will reply with the same TxnId.
    - Length of Payload – This specifies the number of bytes of payload in the TCP Message.
    - Misc Param Count – This specifies the Number of parameters in the payload.

### Payload

* Misc Param Name  2 Bytes. Null Terminated String which will identify the parameter in the message. This is the param id.and has to be unique and will be interpreted as a number at OMC.
* Misc Value Type -> 1 Byte. This specifies the type of the param value. This will become obsolete and removed in next release because all param values are considered to be null terminated strings.
* Misc Value Size  1 Byte. This field will contain the total number of bytes of Misc\_Param\_Value.
* Misc\_Param\_Value this will be a null terminated string of characters or number depending on the Param Name. For example”245” as param value will be translated to 245 at OMC.

## TCP/IP MESSAGES DESCRIPTION

This section describes these messages, their formats and the request and the response of the corresponding message.

### Registration Request for a NE (REG\_REQ)

Communication Path: NE----🡪 OMC

#### Description:

The Registration Request contains the NE Information and used for Registration purposes at OMC.

NE will fill the data in this message from sysOmcRegister table present at NE

NE-ID : NULL

MSG CMD ID (2001) : REG\_REQ

MSG CMD STATUS : CMD\_SUCCESS

PAYLOAD LENGTH : <length of the payload>

MISC PARAM COUNT : 17

TX-ID : <transaction ID>

MISC PARAM NAME1 : Misc Param ID for NE\_COUNTRY

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Country information like”India”

MISC PARAM NAME2 : Misc Param ID for NE\_STATE

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALE : State information like”Rajasthan”

MISC PARAM NAME3 : Misc Param ID for NE\_CITY

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : City information like”Jaipur”

MISC PARAM NAME4 : Misc Param ID for NE\_SITE\_BLDG

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Building information like” B1 Mayur Apartments”

MISC PARAM NAME5 : Misc Param ID for NE\_SITE\_FLOOR

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Floor information like”0”

MISC PARAM NAME6 : Misc Param ID for NE\_SITE\_DIRECTION

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Direction information like”Left” or”Right”

MISC PARAM NAME7 : Misc Param ID for NE\_SITE\_LANDMARK

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Landmark information like”Civil Lines”

MISC PARAM NAME8 : Misc Param ID for NE\_SITE\_NICKNAME

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : nick name information in string HW ADDRS

MISC PARAM NAME9 : Misc Param ID for NE\_SITE\_LONGITUDE

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Longitude information like” 75.830383”

MISC PARAM NAME10 : Misc Param ID for NE\_SITE\_LATITUDE

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Latitude information like” 26.934314 ”

MISC PARAM NAME11 : Misc Param ID for CONTACT\_ADDR

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Contact Address of the site as a null terminated

string.

MISC PARAM NAME12 : Misc Param ID for CONTACT\_PERSON

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Name of the person

MISC PARAM NAME13 : Misc Param ID for CONTACT\_MOBILE

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Mobile Number

MISC PARAM NAME14 : Misc Param ID for ALTERNATE\_CONTACT\_ NO

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Alternate contact number in form of a numeric

string.

MISC PARAM NAME15 : Misc Param ID for CONTACT\_EMAIL

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Email Address.

MISC PARAM NAME16 : Misc Param ID for PROD\_ID

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : NE information whether NE is a BSC or MSC.

MISC PARAM NAME17 : Misc Param ID for ACTIVE\_CARD\_HW\_ID

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Active Card Hardware ID

##### Sample data in ASCII format

Note : (‘|’ is only used as field separator & is not part of actual command)

0|2001|0|206|17|1

2053|1|5|India

2054|1|8|Rajasthan

2055|1|6|Jaipur

2056|1|10| B1 Mayur Apartments

2062|1|1|0

2063|1|5|Right

2064|1|12|Civil lines

2065|1|17|FF:FF:FF:FF:FF:FF

2066|1|3|75.830383

2067|1|3|26.934314

2061|1|15|Contact Address

2057|1|11|Person Name

2058|1|10|9999999999

2059|1|10|1111111111

2060|1|11|rgautam@cscape.in

2052|1|4|6041

2051|1|7|ACHWID1

##### ASCII data Description

NE ID : 0

Cmd Id : 2001

Cmdstatus : 0

Payloadlength : 206

Miscparamcount : 17

TxID : 1

##### Sample data in Hexadecimal format

00 00 00 00 00 00 07 d1 00 00 00 00 00 0f 11 01

08 05 01 05 49 6e 64 69 61 08 06 01 09 52 61 6a

61 73 74 68 61 6e 08 07 01 06 4a 61 69 70 75 72

08 08 01 13 42 31 20 4d 61 79 75 72 20 41 70 61

72 74 6d 65 6e 74 73 08 0e 01 01 30 08 0f 01 05

52 69 67 68 74 08 10 01 0b 43 69 76 69 6c 20 4c

69 6e 65 73 08 11 01 11 46 46 3a 46 46 3a 46 46

3a 46 46 3a 46 46 3a 46 46 08 12 01 09 37 35 2e

38 33 30 33 38 33 08 13 01 09 32 36 2e 39 33 34

33 31 34 08 0d 01 10 4d 61 79 75 72 20 41 70 61

72 74 6d 65 6e 74 73 08 09 01 0b 50 65 72 73 6f

6e 20 4e 61 6d 65 08 0a 01 0b 30 39 39 39 39

39 39 39 39 39 39 08 0b 01 0a 31 31 31 31 31

31 31 31 31 31 08 0c 01 11 72 67 61 75 74 61

6d 40 63 73 63 61 70 65 2e 69 6e 08 04 01 04 36

30 32 31 08 03 01 07 41 43 48 57 49 44 31

##### Hex data Description

###### Header data

00 00 00 00 = 0 -> ne id -> 4 bytes

00 00 07 d1 = 2001 -> cmd id -> 4 bytes

00 00 00 00 = 0 -> cmd status -> 4 bytes

00 0f = 240 -> payloadlength -> 2 bytes

11 = 17 -> Miscparamcount -> 1 byte

01 = 1 -> Tx ID -> 1 byte

###### Payload data

08 05 = 2053 -> Misc Param name -> country

01 = 1 -> Misc Value Type

05 = 5 -> Misc Value size

49 6e 64 69 61 = 73 110 100 105 97 -> India -> Misc Param Value

08 06 = 2054 -> Misc Param name -> state

01 = 1 -> Misc Value Type

07 = 7 -> Misc Value size

52 61 6a 61 73 74 68 61 6e -> Rajasthan -> Misc Param Value

08 07 = 2055 -> Misc Param name -> city

01 = 1 -> Misc Value Type

07 = 7 -> Misc Value size

4a 61 69 70 75 72-> Jaipur -> Misc Param Value

08 08 = 2056 -> Misc Param name -> SITE\_BLDG

01 = 1 -> Misc Value Type

0a = 10 -> Misc Value size

42 31 20 4d 61 79 75 72 20 41 70 61 72 74 6d 65 6e 74 73 -> B1 Mayur Apartments

08 0e = 2062 -> Misc Param name -> SITE\_FLOOR

01 = 1 -> Misc Value Type

02 = 2 -> Misc Value size

31 30 = 49 48 -> 0 -> Misc Param Value

08 0f = 2063 -> Misc Param name -> SITE\_DIRECTION

01 = 1 -> Misc Value Type

04 = 4 -> Misc Value size

52 69 67 68 74 -> Right -> Misc Param Value

08 10 = 2064 -> Misc Param name -> SITE\_LANDMARK

01 = 1 -> Misc Value Type

0b = 11 -> Misc Value size

43 69 76 69 6 20 4c 69 6e 65 73-> Civil Lines -> Misc Param Value

08 11 = 2065 -> Misc Param name -> SITE\_NICKNAME

01 = 1 -> Misc Value Type

11 = 17 -> Misc Value size

46 46 3a 46 46 3a 46 46 3a 46 46 3a 46 46 3a 46 46-> FF:FF:FF:FF:FF:FF-> Misc Param Value

08 12 = 2066 -> Misc Param name -> SITE\_LONGITUDE

01 = 1 -> Misc Value Type

03 = 3 -> Misc Value size

37 35 2e 38 33 30 33 38 33-> 75.830383 -> Misc Param Value

08 13 = 2067 -> Misc Param name -> SITE\_LATITUDE

01 = 1 -> Misc Value Type

03 = 3 -> Misc Value size

32 36 2e 39 33 34 33 31 34-> 26.934314 -> Misc Param Value

08 0d = 2061 -> Misc Param name -> CONTACT\_ADDRESS

01 = 1 -> Misc Value Type

0f = 15 -> Misc Value size

4d 61 79 75 72 20 41 70 61 72 74 6d 65 6e 74 73-> Mayur Apartments -> Misc Param Value

08 09 = 2057 -> Misc Param name -> CONTACT\_PERSON

01 = 1 -> Misc Value Type

0b = 11 -> Misc Value size

50 65 72 73 6f 6e 20 4e 61 6d 65 = 80 101 114 115 111 110 32 78 97 109 101 -> Person Name -> Misc Param Value

08 0a = 2058 -> Misc Param name -> MOBILE\_PHONE\_NUMBER

01 = 1 -> Misc Value Type

0a = 10 -> Misc Value size

39 39 39 39 39 39 39 39 39 39 = 57 57 57 57 57 57 57 57 57 57 -> 9999999999 -> Misc Param Value

08 0b = 2059 -> Misc Param name -> ALT\_CONTACT\_NUMBER

01 = 1 -> Misc Value Type

0a = 10 -> Misc Value size

31 31 31 31 31 31 31 31 31 31 = 49 49 49 49 49 49 49 49 49 49 -> 1111111111 -> Misc Param Value

08 0c = 2060 -> Misc Param name -> EMAIL

01 = 1 -> Misc Value Type

0b = 11 -> Misc Value size

72 67 61 75 74 61 6d 40 63 73 63 61 70 65 2e 69 6e -> rgautam@cscape.in -> Misc Param Value

08 04 = 2052 -> Misc Param name -> PRODUCT\_ID

01 = 1 -> Misc Value Type

04 = 4 -> Misc Value size

36 30 34 31 = 54 48 52 49 -> 6041 -> Misc Param Value

08 03 = 2051 -> Misc Param name -> ACTIVE\_CARD\_HW\_ID

01 = 1 -> Misc Value Type

07 = 7 -> Misc Value size

41 43 48 57 49 44 31 = 65 67 72 87 73 68 49 -> ACHWID1 -> Misc Param Value

### Registration Response

Communication Path: OMC--- NE

#### Description:

The Registration Response is sent to NE after validation of Registration Request Message by IMR-OMC.

NE-ID : NULL

MSG CMD ID (2002) : REG\_RESP

MSG CMD STATUS : CMD\_SUCCESS (if Correctly Received) or

CMD\_FAILURE (if error)

PAYLOAD LENGTH : <length of the payload>

MISC PARAM COUNT : <Total Number of errors> e.g. 1

TX-ID: : <transaction ID>(Same as that of request)

MISC PARAM NAME1 : Misc Param ID for ERROR\_CODE

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <Number of bytes of Param Value>

MISC PARAM VALUE : Error Code ID

#### Sample data in Hexadecimal format

00 00 00 00 00 00 07 d2 00 00 00 00 00 00 00 01

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00

#### Hex data Description

##### Header data

00 00 00 00 = 0 -> ne id -> 4 bytes

00 00 07 d2 = 2002 -> cmd id -> 4 bytes

00 00 00 00 = 0 -> cmd status -> 4 bytes

00 00 = 0 -> payloadlength -> 2 bytes

00 = 0 -> Miscparamcount -> 1 byte

01 = 1 -> Tx ID -> 1 byte

##### Payload data : None

### Configuration Response

Communication Path: OMC--- NE

#### Description:

This message shall be sent by OMC to NE.

This message is in response to the Configuration request, and therefore contains information regarding the NE-ID, the SNMP and the FTP parameters required for communication of SNMP and FTP messages between the NE and the OMC and because this is an absolute confirmation message, therefore, the NE-ID shall be sent as a param value pair in this response message only.NE-ID will be sent from in subsequent TCP Message exchanges.

NE-ID : <NE-ID>

MSG CMD ID (2004) : CONFIG\_RESP

MSG CMD STATUS : CMD\_SUCCESS

PAYLOAD LENGTH : <length of the payload>

MISC PARAM COUNT : 9

TX-ID : <transaction ID>(Same as that of request)

MISC PARAM NAME1 : Misc Param ID for NE\_ID

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <in bytes>

MISC PARAM VAL : <Unique ID of IMR-NE>

MISC PARAM NAME2 : Misc Param ID for USERNAME

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <in bytes>

MISC PARAM VAL : <NE USER NAME FOR FTP Purpose>

MISC PARAM NAME3 : Misc Param ID for PASSWORD

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <in bytes>

MISC PARAM VAL : < NE PASSWORD FOR FTP Purpose >

MISC PARAM NAME4 : Misc Param ID for FTP SERVER IP

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <in bytes>

MISC PARAM VAL : <IP ADDRESS of the FTP server>

MISC PARAM NAME5 : Misc Param ID for SNMP\_WRITE\_COMMUNITY

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <in bytes>

MISC PARAM VAL : <NE the agent SNMP Write Community>

MISC PARAM NAME6 : Misc Param ID for SNMP\_READ\_COMMUNITY

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <in bytes>

MISC PARAM VAL : <NE the agent SNMP Read Community >

MISC PARAM NAME7 : Misc Param ID for FTP\_HOME\_PATH

(Note: ---This path will be Default Root Path like for example /home/omc/dump).

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <in bytes>

MISC PARAM VAL : <The Path Where the FTP client will put the file or

Get the file>

MISC PARAM NAME8 : Misc Param ID for SNMP\_REQUEST\_PORT

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <in bytes>

MISC PARAM VAL : <The port at which the NE will get the Request

Command from Manager>

MISC PARAM NAME9 : Misc Param ID for SNMP\_TRAP\_PORT

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <in bytes>

MISC PARAM VAL : <The port at which the NE will send the Trap

Command for Manager >

#### Sample data in Hexadecimal format

00 00 00 00 00 00 07 d4 00 00 00 00 00 59 09 02

08 19 01 05 61 64 6d 69 6e 08 18 01 05 61 64 6d

69 6e 08 15 01 04 35 31 36 32 08 17 01 06 70 75

62 6c 69 63 08 14 01 03 31 36 31 08 16 01 06 70

75 62 6c 69 63 08 1e 01 01 31 08 1d 01 0e 31 30

2e 31 30 30 2e 32 30 37 2e 31 34 30 08 1b 01 09

2f 68 6f 6d 65 2f 6f 6d 63 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00

#### Hex data Description

##### Header data

00 00 00 00 = 0 -> ne id -> 4 bytes

00 00 07 d4 = 2004 -> cmd id -> 4 bytes

00 00 00 00 = 0 -> cmd status -> 4 bytes

00 59 = 89 -> payloadlength -> 2 bytes

09 = 9 -> Miscparamcount -> 1 byte

02 = 2 -> Tx ID -> 1 byte

##### Payload data

08 19 = 2073 -> Misc Param name -> FTP\_USER\_PASSWORD

01 = 1 -> Misc Value Type

05 = 5 -> Misc Value size

61 64 6d 69 6e = 97 100 109 105 110 -> admin -> Misc\_Param\_Value

08 18 = 2072 -> Misc Param name -> FTP\_USER\_NAME

01 = 1 -> Misc Value Type

05 = 5 -> Misc Value size

61 64 6d 69 6e = 97 100 109 105 110 -> admin -> Misc Param Value

08 15 = 2069 -> Misc Param name -> SNMP\_TRAP\_PORT

01 = 1 -> Misc Value Type

04 = 4 -> Misc Value size

35 31 36 32 = 53 49 54 50 -> 5162 -> Misc Param Value

08 17 = 2071 -> Misc Param name -> SNMP\_WRITE\_COMMUNITY\_PREFIX

01 = 1 -> Misc Value Type

06 = 6 -> Misc Value size

70 75 62 6c 69 63 -> 112 117 98 108 105 99 -> private -> Misc Param Value

08 14 = 2068 -> Misc Param name -> SNMP\_REQUEST\_PORT

01 = 1 -> Misc Value Type

03 = 3 -> Misc Value size

31 36 31 = 49 54 49 -> 161 -> Misc Param Value

08 16 = 2070 -> Misc Param name -> SNMP\_READ\_COMMUNITY\_PREFIX

01 = 1 -> Misc Value Type

06 = 6 -> Misc Value size

70 75 62 6c 69 63 = 112 117 98 108 105 99 -> public -> Misc Param Value

08 1e = 2078 -> Misc Param name -> NE\_ID

01 = 1 -> Misc Value Type

01 = 1 -> Misc Value size

31 = 49 -> 1 -> Misc Param Value

08 1d = 2077 -> Misc Param name -> FTP\_SERVER\_IP

01 = 1 -> Misc Value Type

0e = 14 -> Misc Value size

31 30 2e 31 30 30 2e 32 30 37 2e 31 34 30 -> 49 48 46 49 48 48 46 50 48 55 46 49 52 48 -> 10.100.207.140 -> Misc Param Value

08 1b = 2075 -> Misc Param name -> OMC\_HOME

01 = 1 -> Misc Value Type

09 = 9 -> Misc Value size

2f 68 6f 6d 65 2f 6f 6d 63 - > 47 104 111 109 101 47 111 109 99 -> /home/omc -> Misc Param Value

### Config Status

Communication Path: NE--- OMC

#### Description:

NE validates the information received in Config Response. The following is done at NE:

* NE tries to bind on the SNMP Port.
* It validates Communities received for NULL.
* It connects to FTP Server.
* After successful connection, it logins to FTP Server with the user access information received from OMC.

Success of all this will result in CMD\_SUCCESS being sent to OMC. Failure in any of these checks will be sent as a CMD\_FAILURE.

NE-ID : <NE-ID received as a part of Config Response>

MSG CMD ID (2005) : CONFIG\_STATUS

MSG CMD STATUS : CMD\_SUCCESS (if Correctly Received) or

CMD\_FAILURE (if error)

PAYLOAD LENGTH : <length of the payload>

MISC PARAM COUNT : 0 or 1 (if it is 1, the error code will be present in misc

param)

TX-ID : <transaction ID>

MISC PARAM NAME : Misc Param ID for ERROR\_CODE

MISC VALUE TYPE : MISC\_PARAM\_TYPE\_STR

MISC VALUE SIZE : <in bytes>

MISC PARAM VAL : <The Error Code ID > [Refer sec names as ERROR

CODES]

### ADD NODE STATUS

Communication Path: OMC--- NE

#### Description:

OMC sends the GetBulk request to node and add the Node at OMC and sends this message to the NE to indicate the NE to start the file Transfer from NE.

NE-ID : <NE-ID copied from the Configuration Status.>

MSG CMD ID (2006) : ADD\_NODE\_STATUS

MSG CMD STATUS : CMD\_SUCCESS <if successful> or

CMD\_FAILURE <if un-successful>

PAYLOAD LENGTH : <length of the payload>

MISC PARAM COUNT : 0

TX-ID : <transaction ID of Config Status>

#### Sample data in Hexadecimal format

1. 00 00 01 00 00 07 d5 00 00 00 00 00 00 00 03

#### Hex data Description

##### Header data

00 00 00 00 = 0 -> ne id -> 4 bytes

00 00 07 d5 = 2005 -> cmd id -> 4 bytes

00 00 00 00 = 0 -> cmd status -> 4 bytes

00 00 = 0 -> payloadlength -> 2 bytes

00 = 0 -> Miscparamcount -> 1 byte

03 = 3 -> Tx ID -> 1 byte

##### Payload data : None

### NE-STATUS

Communication Path: NE--- OMC

#### Description:

The NE -STATUS from NE indicates the health of NE.

NE-ID : <NE-ID>

MSG CMD ID : <NE STATUS>

MSG CMD STATUS : 0

PAYLOAD LENGTH : 0

MISC PARAM COUNT : 0

TX-ID : <transaction ID>

## TCP/IP Command Ids

BASE\_CMDID\_OMC= 2000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Command Name** | **CMD\_ID** | **Direction** | **Comments** |
|  | REG\_REQ | BASE\_CMDID\_OMC+1 | NE to OMC |  |
|  | REG\_RESP | BASE\_CMDID\_OMC+2 | OMC to NE |  |
|  | CONFIG\_REQ | BASE\_CMDID\_OMC+3 | NE to OMC |  |
|  | CONFIG\_RESP | BASE\_CMDID\_OMC+4 | OMC to NE |  |
|  | CONFIG\_STATUS | BASE\_CMDID\_OMC+5 | NE to OMC |  |
|  | ADD\_NODE\_STATUS | BASE\_CMDID\_OMC+6 | OMC to NE | Add Node successful / unsuccessful |

## TCP MESSAGE PARAMS

BASE\_PARAMID\_OAM\_OMC = 2050

**Param Name Param Value**

ACTIVE\_CARD\_HW\_ID BASE\_PARAMID\_OAM\_OMC +1

PRODUCT\_ID BASE\_PARAMID\_OAM\_OMC +2

NE\_COUNTRY BASE\_PARAMID\_OAM\_OMC +3

NE\_STATE BASE\_PARAMID\_OAM\_OMC +4

NE\_CITY BASE\_PARAMID\_OAM\_OMC +5

NE\_SITE\_BLDG BASE\_PARAMID\_OAM\_OMC +6

CONTACT\_PERSON BASE\_PARAMID\_OAM\_OMC +7

CONTACT\_MOBILE BASE\_PARAMID\_OAM\_OMC +8

ALTERNATE\_CONTACT\_NO BASE\_PARAMID\_OAM\_OMC +9

CONTACT\_EMAIL BASE\_PARAMID\_OAM\_OMC +10

CONTACT\_ADDRESS BASE\_PARAMID\_OAM\_OMC +11

NE\_SITE\_FLOOR BASE\_PARAMID\_OAM\_OMC +12

NE\_SITE\_DIRECTION BASE\_PARAMID\_OAM\_OMC +13

NE\_SITE\_LANDMARK BASE\_PARAMID\_OAM\_OMC +14

NE\_SITE\_NICKNAME BASE\_PARAMID\_OAM\_OMC +15

NE\_SITE\_LONGITUDE BASE\_PARAMID\_OAM\_OMC +16

NE\_SITE\_LATITUDE BASE\_PARAMID\_OAM\_OMC +17

SNMP\_REQUEST\_PORT BASE\_PARAMID\_OAM\_OMC +18

SNMP\_TRAP\_PORT BASE\_PARAMID\_OAM\_OMC +19

SNMP\_READ\_COMMUNITY\_PREFIX BASE\_PARAMID\_OAM\_OMC +20

SNMP\_WRITE\_COMMUNITY\_PREFIX BASE\_PARAMID\_OAM\_OMC +21

FTP\_USER\_NAME BASE\_PARAMID\_OAM\_OMC +22

FTP\_USER\_PASSWORD BASE\_PARAMID\_OAM\_OMC +23

OMC\_HOME BASE\_PARAMID\_OAM\_OMC +25

FTP\_SERVER\_IP BASE\_PARAMID\_OAM\_OMC +27

NE\_ID BASE\_PARAMID\_OAM\_OMC +28

# Test Report

*<Describe what “development” / Integration unit test has been done – and what the test results here are>*