NORTEL

Carrier Voice over IP

OSSGate User Guide

NE10004-512

Document status: Preliminary
Document version: 10.03
Document date: 05/26/2006

May 2006
10.03 Preliminary version - (I)SN09FF and up

May 2006
10.02 Preliminary version - (I)SN09FF and up

April 2006
10.01 Preliminary version - (I)SN09FF and up

January 08, 2006
First draft

November 22, 2005
Template creation

Contents

New in this Release

The following table highlights the features introduced in this release. Refer to the OSS Advanced Feature Guide for more information about new features in this release.

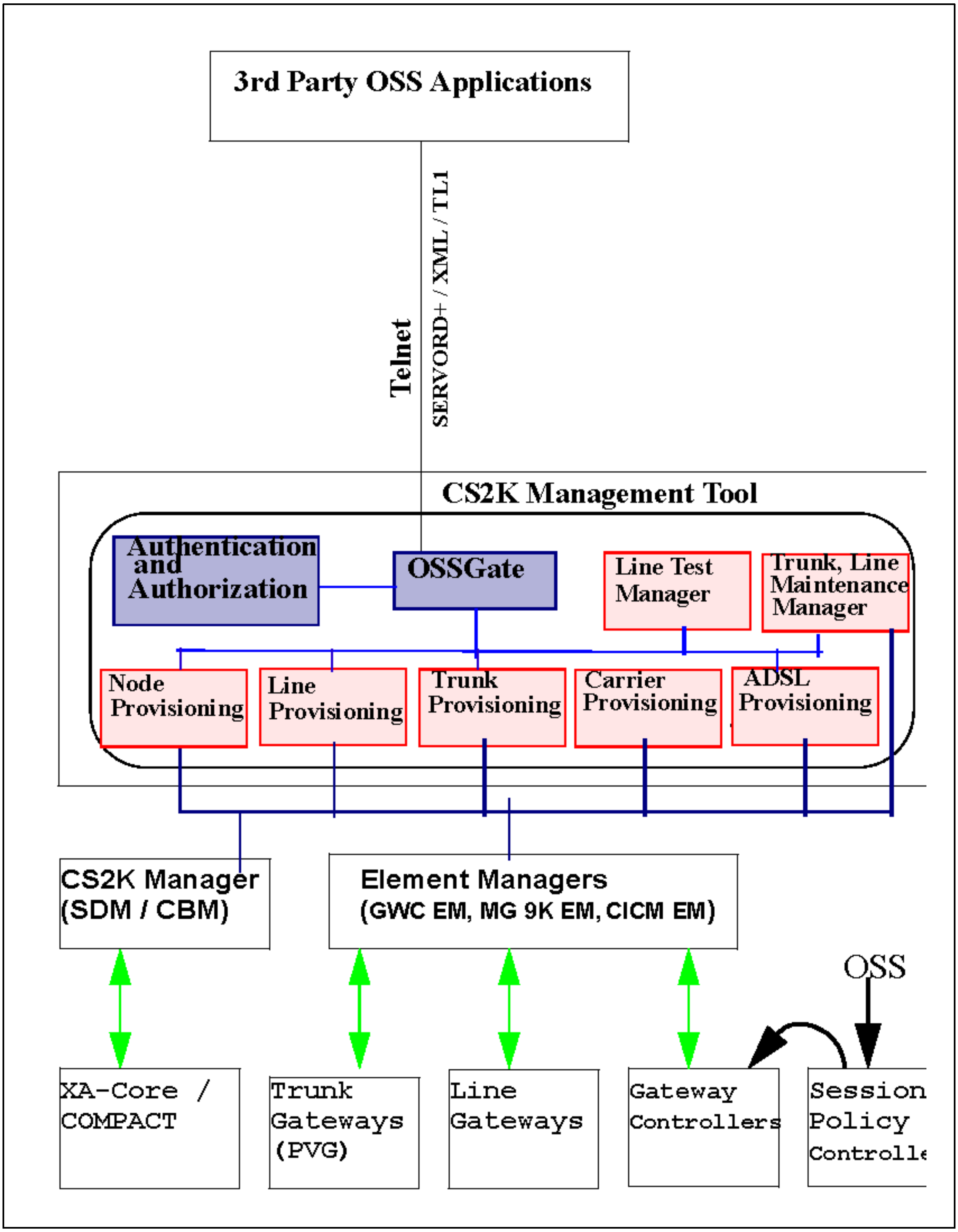
Feature descriptions	
SN09FF	
	Added note to description of parameter circuitNumber in section Terminology and description of method parameters in chapter Provisioning third party large line gateways and endpoints. CircuitNumber description reflects that this parameter equals the terminalNumber -1. Updated subsequent request and response examples to reflect this change. Added note to cardCode that this parameter must be specified for the international Core. In chapter Line Maintenance Manager (LMM), added new subsection Limitations and restrictions to section LMM commands, including a bulleted item that LMM commands post -gw, bsy -gw, rts -gw, and tst -gw do not support SIP lines. Added description and example of LMM CLUI help command.
	Added information on feature SESM Support for SIP Lines Part 2, affecting the following chapters: • New chapter Line Maintenance Manager (LMM) describes usage of the LMM configuration tool and the LMM CLUI, a replacement for the LMM GUI. • New section SERVORD+ commands for SIP Lines Provisioning added to chapter Line provisioning with OSSGate. This section describes SIP Lines provisioning information as it affects SERVORD+ commands
	Added XML command examples to chapter Provisioning third party large line gateways and endpoints from feature SESM Flex Large Lines GWC Profile Support - Phase 1. (This feature provides generic support for large line gateways supporting up to a maximum of 2047 endpoints in a virtual media gateway. Additional support is provided for the Audiocodes TP6310 card when configured in a Mediant 8000 chassis.)
	Added description of Configuration Menu to chapter Batch Provisioning Tool. Also added brief usage of Stop Mode command to section OSSGate commands in chapter Using OSSGate. Added example of delete GWC request and delete GWC response to chapter Nodes Provisioning with OSSGate.

OSSGate Introduction

OSSGate is an application that provides a machine interface for provisioning components within Carrier VoIP. The main functionality of OSSGate is to act as a gateway to the Node, Carrier, Trunk, Line, ADSL Provisioning applications and the Trunk Maintenance application. It provides the end user with an alternative to the GUI interface as a method for provisioning Carrier VoIP components; one that allows more automation (less human intervention) than the GUI interface.

OSSGate provides TCP/IP connectivity as an alternative to the asynchronous connections used in the DMS100F. It implements a telnet protocol over a simple TCP/IP socket connection to allow clients to connect to OSSGate. Although it is based on telnet it does not implement a complete set of telnet capabilities. It implements the telnet echo option, which is used to negotiate with telnet client to turn on or off local echo of characters, especially during login time. It does not require the client to support all the telnet options. It does require the client to support line mode. The client or OSS establishes a TCP/IP socket connection to a specific port number (user configurable) on the server running the OSSGate application.

OSSGate supports three modes - XML, CI and TL1. The XML interface is used to send provisioning commands to the Nodes, Carrier, and Trunk applications and maintenance commands to the TMM application. The CI interface is used to send provisioning commands to the Lines Provisioning application. The TL1 interface supported from (I)SN08 is used to send line test commands for MG 9000 lines. OSSGate also performs basic syntax checks on XML, TL1 input before sending it to the Nodes, Carrier, ADSL, TMM, Trunk provisioning and LTM line test applications.


3rd Party OSS Applications
Telnet
SERVORD+ / XML / TL1
CS2K Management Tool
Authentication and Authorization
OSSGate
Line Test Manager
Trunk, Line Maintenance Manager
Node Provisioning
Line Provisioning
Trunk Provisioning
Carrier Provisioning
ADSL Provisioning
CS2K Manager (SDM / CBM)
Element Managers (GWC EM, MG 9K EM, CICM EM)
OSS
XA-Core / COMPACT
Trunk Gateways (PVG)
Line Gateways
Gateway Controllers
Session Policy Controlle

Using OSSGate

The OSSGate server runs as part of the CS 2000 Management Tools application. The OSSGate server continuously listens for incoming connection requests. For each connection request, it starts a session after the username and password authentication. Such a session can be used for sending various provisioning commands (for example, SERVORD+ for Line Provisioning commands, XML for Node, Carrier, ADSL and Trunk provisioning). OSSGate can be used by either an OSS or a standard telnet client.

Supported provisioning connections

Following are recommended usage limits apply when each application is being run in isolation (for example, you can have 10 people doing Nodes provisioning as long as zero people are doing Lines, Carriers, and LMM at the same time):

- Node provisioning (includes add/delete/query of GWs) - maximum simultaneous users: 10 (only 5 of which can be from the CS 2000 Management Tools GUI)
- Line provisioning - maximum simultaneous users: 30
- Carrier provisioning - maximum simultaneous users: 4
- LMM - maximum simultaneous GUI users: 10
- TMM - maximum simultaneous users: 10
- Trunk provisioning - maximum simultaneous users: 5

The maximum number of concurrent OSSGate connections is 45. Following are maximum recommended usage limits for concurrent operations (for example when performing Nodes provisioning, Lines Provisioning, LMM and TMM all at the same time, these limits apply):

- Nodes: 5 users
- Lines: 30 users
- Carriers: 4 users
- LMM: 2 users (GUI)
- TMM: 2 users
- TMM: 2 users

Note: System response will increase as the number of concurrent connections increase.

Connecting to OSSGate using telnet

A system connects to the OSSGate server by initiating a telnet session to the correct host name (or IP address) and port number. The user must belong to the primary authentication group "succssn" to login to OSSGate.

```
%telnet <ptm_hostname or IP address> <ossgate_primary_port>
```

> ***Note:*** The host name or IP address is assigned by the customer network administrator. It is assigned to the server where OSSGate is configured during installation and setup. Refer to your local network administrator for the correct address. The default primary port is 10023 (Standard Telnet Port 23 + 10000).

Example of OSSGate logging when OSSGate Password Control option is on. (User input is highlighted)

```
$telnet sesmhost 10023

Trying 47.14.10.45
Connected to sesmhost.
Escape character is '^]'.
Enter User Name

>tom

Enter password

>

tom logged in on 04/03/25 at 06:10:58.
******************************************************
**                                                  **
**                   OSS Gateway                       **
**           This is a PRIVATE Database.          **
**                                                  **
**      All activity is subject to monitoring.    **
**                                                  **
**   Any UNAUTHORIZED access or use is PROHIBITED  **
**           and may result in PROSECUTION         **
**                                                  **
******************************************************
**  WARNING: Different service types require      **
**   (I)SN08 Build: Jan 31, 2005 1:36:44 AM        **
******************************************************
```

Example of OSSGate login when OSSGate Password Control option is set to default value off (User input is highlighted)

```
$ telnet sesmhost 10023

Trying 42.120.94.57...
```

```
Connected to sesmhost.
Escape character is '^]'.

Enter username and password

>maint maint

maint logged in on 03/11/08 at 08:08:05.
******************************************************
**                                                  **
**                  OSS Gateway                    **
**                                                  **
**          This is a PRIVATE Database.            **
**                                                  **
**    All activity is subject to monitoring        **
**                                                  **
** Any UNAUTHORIZED access or use is PROHIBITED   **
**         and may result in PROSECUTION.          **
**                                                  **
******************************************************
** WARNING: Different service types require        **
** different formats. Consult the OSSGate User's  **
** Guide, NE10004-512, for detailed information    **
** regarding command formats and usage rules.      **
** Failure to do so can cause a mismatch between  **
** XA-Core and SESM data.                          **
******************************************************

(I)SN08 Build: Jan 31, 2005 1:36:44 AM
******************************************************
```

OSS/Telnet Client Requirements

Non-Secure channel between client and OSSGate

In order for a telnet connection into OSSGate to work, there are some requirements that the telnet client must support. Listed below are the options that the telnet client should support:

- Support Line Mode. For example. the telnet client should NOT send character by character to OSSGate. Instead, it should be able to send one line at a time.
- The telnet client should be able to implicitly add a Carrier Return (CR) to any data coming from the OSSGate server.
- The telnet client should implement telnet echo option since this will be negotiated by OSSGate to turn on/off during login.

Any client that supports these options will work.

Putty Configuration

A freeware Windows telnet client (putty.exe) which supports the above options can be found at http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html. After starting up putty, remember to set the above options by checking the following in the Terminal category:

- Implicit CR in every LF
- Auto wrap mode initially on

Note: In (I)SN08 OSSGate was tested using Putty Release beta 0.53b version on a Windows 2000 machine. Nortel neither recommends nor provides support for putty.

Unix telnet clients work with OSSGate without the need of any explicit options setting.

Setting Secure channel between client and OSSGate using SSH Port Forwarding

In order to have a secure (i.e., encrypted) channel of communication between OSS/telnet clients and OSSGate server, SSH port forwarding needs to be set up. Refer to the instructions that follow on how to do this for Unix and Windows. Once set up, SSH Port Forwarding establishes a port forwarding session from client to server, wherein all data forwarded are encrypted and hence secure. Regardless of Unix or Windows client, when using SSH port forwarding, OSSGate needs to be configured to set ssh port_forward=yes, since this will disallow any non-secure telnet clients to connect to OSSGate. Shown below is example of how to set up port forwarding using openSSH implementation on Unix. The list of SSH clients, both commercial and otherwise for various platforms, is given in section “Other SSH Products.”

Unix

To set up SSH Port Forwarding on the Unix machine, install the SSH software first. After installation, establish a port forwarding session between this machine and the CS 2000 management tool server. The following command does that.

```
# ssh -L <local-port>:<remote-host>:<remote-port> remote-Host
```

The value for remote-port is typically 10023 (unless OSSGate is configured to listen on a different port). The value for remote-host is the host name of the CS 2000 Management Tools server. User can choose any value for local-port, but higher numbers like 2000 and above are preferred.

The first time this command is run on the user’s machine in an attempt to forward data to remote-host, the user will be prompted with information as shown here:

```
The authenticity of host 'remote-Host (1.2.3.4)' can't
be established.  RSA key fingerprint is <finger print
information>.  Are you sure you want to continue connecting
(yes/no)?
```

SSH is verifying with the user whether the host remote-host is a trusted host and whether the user wants to continue connecting to it. The user should enter 'yes'. Next the user will be prompted for a password for the current user logged in. Once the entered password is verified, a successful port forwarding session is established. This means that all new sessions by this user connecting to localhost:local-port will be forwarded to remote-host:remote-port in a secure channel. See the example below:

> **Example**
> For example, a user sets up SSH port forwarding on machine A with the following values:
> **`# ssh -L 2001:CS2000 Management Tools host:10023 CS 2000 Management Tools host`**

To securely transmit data from machine A to the server (where OSSGate is running), the user needs to open a window logged into machine A and type the following:

```
# telnet localhost 2001
```

The telnet connection automatically gets secured between machine "A" and the server running the CS 2000 Management Tools applications. The OSSGate session will look normal, the user does not see any visible change.

Windows:

The freeware telnet client putty mentioned in the section "Non-Secure channel between client and OSSGate" (page 20) also supports SSH. In order to secure the connection between the telnet client and OSSGate server, the user has to configure the SSH port forwarding using the 'putty' software release 0.53B as shown below.

Step	Action
1	Choose Session Category
	a. Enter Host name with value of the SUN server that is running the CS 2000 Management Tools applications.
	b. Enter port number as 22
	c. Choose protocol as SSH
2	Choose Tunnels Category
	a. Under Destination, enter host:port where host is the CS 2000 Management Tools host name and port is the port number where

the OSSGate server listens for input. 10023 is the standard port where the OSSGate server listens for a client connection.

b. Choose ’Local’ radio button, and click Add.

c. Click on Open

—End—

The user will be prompted to enter a user and password to log in to the CS 2000 Management Tools application host/server. Once a valid user and password is entered, this port forwarding session is established.

The user can now establish a secure connection between the client machine and CS 2000 Management Tools host as long as the port forwarding session (above) exists, by doing the following.

Step	Action
1	Right-click on the top left side of the window. From the pull-down menu, choose New Session. A Putty Configuration dialog window opens up.
2	Under Session category, enter the string ’localhost’ under the HostName field. Enter port 2001, since the local port chosen was 2001 in the example above in the port field. Choose telnet protocol.
3	Choose Terminal options as explain in the section "Putty Configuration" (page 21) Click on Open.

—End—

A secure session with OSSGate server is established as detailed above.

> ***Note:*** Shown above are examples to set up SSH port forwarding to one CS 2000 Management Tools host from a client machine. If a client machine is to have multiple port forwarding sessions, the user must choose different ‘local’ port numbers. In the examples above, the local port chosen was 2001. If the user wants another port forwarding session to another host, he must choose one.

Please refer to "SSH Product Comparison" (page 411) for available SSH products and comparison.

OSSGate commands

The following commands are supported by OSSGate. Note that the following commands are accepted only in Control mode (‘?’ prompt). To change from input mode (‘>’ prompt) to Control mode, enter (Ctrl+B) at ‘>’ prompt.

Step	Action

1 login

Allows user to log in with username and password separated by a space. If a user enters this command when already logged in, OSSGate logs out the previous user. Note that when the user telnets into OSSGate, it automatically prompts for login.

Screen output shown below (User input highlighted):

```
> ^B

? login

admin logged out.
Enter username and password

> maint maint

maint logged in on 03/11/08 at 08:08:05.
******************************************************
**                                                  **
**                  OSS Gateway                     **
**                                                  **
**         This is a PRIVATE Database.              **
**                                                  **
**     All activity is subject to monitoring.       **
**                                                  **
**   Any UNAUTHORIZED access or use is PROHIBITED   **
**          and may result in PROSECUTION.          **
**                                                  **
******************************************************

** WARNING: Different service types require         **
** different formats. Consult the OSSGate User’s    **
** Guide, NE10004-512, for detailed
information      **
** regarding command formats and usage
rules.        **
** Failure to do so can cause a mismatch
between     **
** XA-Core and SESM data.
**
******************************************************

(I)SN08 Build: Jan 31, 2005 1:36:44 AM
******************************************************
```

```
>
```

If a login user id or password is invalid or the user does not belong to the primary authentication group “succssn,” login fails and user gets an appropriate error response

```
Enter username and password

>maint maint

User Login Failed, Reason:  Authentication failed.
Invalid Username / Password / Unauthorized Group.
```

If three consecutive attempts to login fail, the telnet session will terminate.

2 logout

Allows the user to log out of OSSGate.

```
> ^B

? logout

maint logged out.

>
```

The current or new user can login after a logout using the login command.

```
> ^B

? login

Enter username and password

>
```

3 security

Allows the user to know what security service is being used by OSSGate.

Example (User input is highlighted):

```
> ^B

? security

Security service is JAAS/PAM based
```

4 clearconv

Short for "clear conversation," it allows the user to end the OSSGate session.

```
> ^B

? clearconv
```

```
SESSION TERMINATED
Connection closed by foreign host.
```

5 sesm_version

this communicated and displays the version of CS 2000 Management Tools

```
> ^B

? sesm_version

(I)SN0x Build:  Jul 22, 2006, 12:26:02 PM
```

where

where x = current software release

6 mode (OSSGate mods>

When this command is issued without the parameter, it will display the current mode the OSSGate session is using. The default mode upon login is set to CI. OSSGate supports three modes, the CI mode, the XML mode, and the TL1 mode.

Examples: (User input is highlighted):

Example of querying the current mode

```
> ^B

? mode

Mode is XML.
```

Example of switching to CI mode

```
>^B

? mode ci

Mode is CI

>^B

? mode tl1

Mode is TL1.
```

> ***Note:*** For more information on TL1 mode, refer to "TL1 Line Test Interface for MG 9000 lines" (page 374)

7 stop mode

A user can enter the following Stop Mode-related commands during OSSGate use:

To query the current Stop Mode setting, enter the following command.

```
>^B

? stop
Stop Mode OFF
```

To switch the current Stop Mode setting to Off, enter the following command.

```
>^B

? stop off
Stop Mode OFF
```

To

To switch the current Stop Mode setting to On, enter the following command.

```
>^B

? stop on
Stop Mode ON
```

—End—

Disconnecting from OSSGate

To properly disconnect from OSSGate, the user (or OSS client) should first log out from the session and then close the session.

Example (User input is highlighted):

```
> ^B

? logout

user1 logged out

> ^B

? clearconv

SESSION TERMINATED
Connection closed by foreign host.
```

Note: A user can terminate a session without explicitly logging out, since the clearconv command automatically logs out any logged-in user. The logout command is useful, for example, when a user has finished doing the operations and wants to let another user use the same session, the first user can log out and let another user log in without ending the session.

Configuring OSSGate

Configuration properties

The following properties can be configured for the OSSGate server:

- Telnet primary port
  - This property specifies the primary port on which OSSGate would listen for input. The default port is 10023, which can be overridden by changing this property value in the properties file. The maximum value for this port is 65535. If the user is to override this number, it’s advisable to choose a high number, since many standard servers like ftp (21), telnet (23) use lower range numbers.
- Telnet secondary port
  - This property specifies the secondary port on which OSSGate would listen for input if OSSGate failed to bind to the port specified in ossgate.telnet.primary_port property. The default value for this port is 11023, which can be overridden by changing this property value in the properties file. The maximum value for this port is 65535.

    Note: Check if the port numbers chosen is not already in use by some other standard server - look in the /etc/services file and ensure that the port number is not already used by some other application. If a port is chosen which another application is already assigned, an failure will occur on one of the two applications attempting to use the port.
- Telnet session timeout
  - This property specifies the time (in seconds) OSSGate will wait before timing out if there is no input on a session. The default timeout value is 600 seconds (10 minutes). The valid range for this property is a positive number less than 4294967296. (4 bytes)
- SSH port forwarding
  - This property, when set to “y” indicates that the connection between OSS and OSSGate is secured using SSH port forwarding. Hence, OSSGate will reject insecure connections coming directly to it from remote hosts. When the value is set to “n” OSSGate will accept connections from remote clients over a non-secure channel.
- OSSGate Password Control
  - This property, has default value of off. When this option is set to ’on’, and the user telnets into OSSGate, the user name is prompted

first and then on a second line, the password. The password entered is not echoed to screen. This is more secure option and is recommended to be used. When this option value is set to off (default) the user name and password are prompted in the same line and the password entered is displayed on the screen. Note that changing this option value to on will impact OSS software and they should be ready to handle the change in OSSGate login prompts.

Prerequisites to reset the properties

- If OSSGate's Telnet Primary port or Telnet Secondary port information is changed, then in order for the changes to be reflected, the CS 2000 management tool applications should be stopped (section 2.1.2), the properties value changed (section 2.1.3) and then CS 2000 management tool applications should be started (section 2.1.4).
- If any other property of OSSGate is changed, then just changing the property values will cause the values to be reloaded (section 2.1.3)

Stopping the CS 2000 Management Tools server

WARNING

The operation detailed below will stop all the CS 2000 Management Tools applications running on the server. It is recommended that the procedure below is performed during a maintenance window if any of the above properties need to be reset.

Step **Action**

1 Login as user root to the CS 2000 Management Tools server

2 Enter the following command and press the enter key.

```
# servstop SESMService
```

—End—

This will force stop all the applications on this server, including the proxy agent. The server stop can be verified by using the status option. The status check must show application as "NOTRUNNING" similar to details shown below.

```
#ptmctl status
```

```
SESM STATUS ---------------------------
COMPONENT      STATUS
---------                 ------
Proxy Agent           NOT RUNNING  RMI
Registry          NOT RUNNING
Snmpfactory           NOT RUNNING
MI2 Server            NOT RUNNING
Current number of SESM processes running: 0 (of 4)
SESM APPLICATION STATUS: No Applications are ready
```

Changing OSSGate parameters

Only user **root** can change the configuration parameters.

Example of changing configuration parameters

```
# configure
SESM configuration
1 - SESM common configuration (IP addresses,
Market, CM CLLI)
2 - SESM database tools
3 - SESM related applications configuration
(MG9K, LMM, CICM)
4 - SESM provisioning configuration
5 - SESM logging configuration (syslog, sesm debug log)
6 - view sesm configuration settings
7 - SESM refresh properties
X - exit
select - 4
SESM Provisioning Configuration
1 - XML Trunk Provisioning configuration
2 - OSSGate Provisioning configuration
3 - Office Provisioning Type configuration
X - exit
select - 2
SESM OSSGate Provisioning Configuration
Enter the Primary Port OSSGate will use for telnet
connections (default:  10023):  11123
Enter the Secondary Port OSSGate will use for telnet
connections (default:  11023):  12223
Enter the OSSGate Timeout period, in seconds
(default: 600):
Do you wish to have OSSGate Port Forwarding on
for SSH [y/n]: n
Do you wish to turn on/off OSSGate Password Control
option  (when this option is turned on, OSSGate
will not echo password on screen when logging onto
it) [default:off] : on
Primary Port :  11123
Secondary Port:  12223
Timeout      : 600
Port Forwarding for SSH: no
Password Control      : on
```

```
Are these information correct?  (y or n) :  y
  Updating the property files with changed values,
please wait...
Reloading the changed property values into memory, please
wait...
```

In the above example 11123 was entered for primary port, 12223 was entered for secondary port, a carriage return to pick the default value of 600 for timeout period and “n” for port forwarding on SSH. Once the correct data is entered, user can exit back to command prompt level.

Starting the CS 2000 Management Tools server

For OSSGate primary and secondary port configuration parameter changes to take effect the CS 2000 management tool server must be started. To start login as user root to the CS 2000 management tool server. Then enter the following command followed by the enter key.

```
# servstart SESMService
```

Successful startup of the server and all applications, picks up the updated configuration. Users can now telnet to OSSGate using the changed port information.

Configuring Log levels

CS 2000 Management tool server provides ability to run the server and its applications with different log levels. This enables the user to configure the server and applications to generate internal log files during certain situations when errors are detected. *ATM/IP Fault Management,* NN10408-900, details the procedure to set the system to different log levels.

User Authorization

User groups

Users of OSSGate applications must belong to the primary user group "succssn" and to on or more secondary user groups listed in the table below.

trkadm	lnadm	mgcadm	mgadm	emsadm
trkrw	lnrw	mgcrw	mgrw	emsrw
trksprov	lnsprov	mgcsprov	mgsprov	emsprov
trkmtc	lnmtc	mgcmtc	mgmtc	emsmtc
trkro	lnro	mgcro	mgro	emsro

A secondary user group consist of a user group domain which defines the range of applications to which a user group applies. The following are valid domains for OSSGate applications:

- **adm** groups have permissions to do everything rw, mtc, sprov, and ro can do and more
- **rw** groups can do everything mtc, sprov, and ro can do and more, but cannot do adm-specific operations
- **mtc** groups can do everything sprov and ro can do and more, but cannot do adm and rw-specific operations
- **sprov** groups can do everything ro can do and more, but cannot do adm, rw, or mtc-specific operations
- **ro** groups have the lease permissions. They can only do read only operations.

ATM/IP Security and Administration, NN10402-600, lists the various steps to create users, assign users to specific group’s user group domains etc. The mapping of application specific operations to user groups is listed in the application specific chapters discussed in this document. If the user does not have sufficient permissions to execute the command, an error message “Insufficient Security Privileges to perform this action” is returned as part of the response.

Security logs

This log is generated when a user logs into the system, attempts to perform an unauthorized operation, a session expires and when user logs out. The log file is located in the server at /var/log/securitylog file on the Succession Server Platform Foundation Software (SSPFS) machine.

Authorization Audit Logs

Authorization Audit Log: this log is generated when a user is successfully authorized to perform an operation. This log does not imply that the user successfully completes the operation. It only indicates the user was granted permission to perform the operation.

This is an Audit Log and the default location for the logs can be found at: /var/log/auditlog directory of the SSPFS machine.

User accounts and group administration

For details on how to set user accounts, assign users to different user groups please refer to *ATM/IP Security and Administration*, NN10402-600.

Line Maintenance Manager (LMM)

Configuration tool

The LMM configuration tool prompts the user to enter the following link information for open provision interface (OPI) and open maintenance interface (OMI):

- transport protocol to the Session Server Element Manager (SSEM) server
- IP address (host name) of the primary SSEM server
- IP address (host name) of the secondary SSEM server, if available. If a secondary SSEM server has been configured, enter the same information as for the primary SESM server.
- Hypertext Transmission Protocol Secure (HTTPS) communication port to the SSEM server (12121)
- SSEM provisioning manager administration user name - added to the SESM. properties as clear text
- SSEM provisioning manager administration password - stored separately and accessible to the root user only

The IP address and the port information generates the SSEM provisioning manager URL which is added to the sesm.properties file. The IP address also is validated for format, range and reachability. Other user input is validated for format, range, and values.

Accessing the LMM configuration tool

To configure SSEM access to the SSEM, enter the following command:

```
ucars00q3c-unit0(active):/> /opt/nortel/NTsesm/admin/bin/con
figure
```

The following shows an example of the LMM configuration tool output.

```
SESM configuration
1 - SESM common configuration (IP addresses, Market, CM CLLI)
2 - SESM database tools
3 - SESM related applications configuration (MG9K, LMM, CICM,
CS2KSS)
4 - SESM provisioning configuration
5 - SESM logging configuration (syslog, sesm debug log)
6 - view sesm configuration settings
7 - SESM refresh properties
X - exit
select 3
SESM Related Applications Configuration
```

```
1 - SESM MG9K configuration
2 - SESM LMM CM CLLI registration
3 - SESM CICM EM configuration
4 - SESM CS2KSS Provisioning Manager Server configuration
X - exit
select - 4
SESM-CS2KSS configuration
OPI - Provisioning Manager Interface configuration
Enter OPI transport protocol to EM server ( http/https -
default:  https)
[1] - http
[2] - https
[x] - exit
select - 2
Enter the IP address / Name of the primary CS2KSS prov server
(default:  47.142.91.164) :  47.142.91.164
Enter the primary CS2KSS prov server https port (default is
8443):  8443
Enter the IP address / Name of the secondary  CS2KSS prov
server (default: 47.142.91.164) : 47.142.91.164
Enter the secondary CS2KSS EM server https port (default is
8443):  8443
Enter the OPI administrator user name (default:  admin):
admin
Enter the OPI administrator password:
Re-enter the OPI administrator password:
OMI - Provisioning Manager Interface configuration
Enter OMI transport protocol to EM server ( http/https -
default:  https)
[1] - http
[2] - https
[x] - exit
select - 2
Enter the IP address / Name of the primary CS2KSS mtc server
(default:  47.142.91.164) :  47.142.91.164
Enter the primary CS2KSS mtc server https port (default is
12121):  12121
Enter the IP address / Name of the secondary  CS2KSS mtc
server (default: 47.142.91.164) : 47.142.91.164
Enter the secondary CS2KSS mtc server https port (default is
12121):  12121
Enter the OMI administrator user name (default:  admin):
admin
Enter the OMI administrator password:
Re-enter the OMI administrator password:
OPI - Provisioning Manager configuration
Transport protocol to EM server :  https://
Transport protocol to EM server :  https://
Primary CS2KSS Provisioning Manager IP/Name :  47.142.91.164
Primary CS2KSS Provisioning Manager port :  8443
```

```
Secondary CS2KSS Provisioning Manager IP/Name :  47.142.91
.164
Secondary CS2KSS Provisioning Manager port :  8443
The administrator user name :  admin
OMI - Maintenance Manager configuration
Transport protocol to Maintenance Manager :  https://
Primary CS2KSS Maintenance Manager IP/Name :  47.142.91.164
Primary CS2KSS Maintenance Manager port :  12121
Secondary CS2KSS Maintenance Manager IP/Name :  47.142.91.164
Secondary CS2KSS Maintenance Manager port :12121
The administrator user name :  admin
Is the information correct?  (y or n) :  y
```

LMM CLUI

The LMM command line user interface (CLUI) is a replacement for the LMM GUI that enables a user to perform the following functions:

- view LMM configuration
- generate query gateway reports
- execute line maintenance commands

The user can post Carrier Voice over IP (CVoIP) and non-CVoIP lines, as well as perform functions similar to the LTP level.

> ***Note:*** The LMM CLUI is intended for use with lines or gateway types other than Session Initiation Protocol (SIP) lines.

To access the LMM CLUI, first telnet to OSSGate. From the OSSGate, access the MTC level. The following commands show how to access the LMM CLUI.

```
>^B <enter>
```

```
?
```

The question mark indicates that the user is in the control mode. Change to the MTC mode.

```
? mode MTC <enter>
```

```
Mode is mtc.
```

MTC commands

To display the SDM IP and SMCLLI names, enter the CMCLLI command.

```
>mtc
```

```
>CMCLLI
```

```
SDM IP = 47.142.134.124
CM CLLI = RTP2
```

Enter the quit command to exit.

```
>quit
```

LMM commands

To enter the LMM CLUI, enter the following command.

```
>lmm
```

Enter the following commands to perform operations on a single line or a set of lines in the LMM mode.

Command format

```
>post -d <DN> [verbose]

>post -l <LEN> [verbose]

>post -gw <GW/VMG> [verbose]

>post -u <URI> [verbose]

>post -display [verbose]

>post -count

>bsy -d <DN> [inb]

>bsy -l <LEN> [inb]

>bsy -qw <GW or VMG> [inb]

>bsy -set [inb]

>rts -d <DN>

>rts -l <LEN>

>rts -gw <GW/VMG>

>rts -set

>tst -d <DN>

>tst -l <LEN>

>tst -gw <GW/VMG Name>

>tst -u <URI>

>tst -set

>frls -d <DN>

>frls -l <LEN>

>help
```

To exit from the LMM CLUI, enter the following command.

```
>exit
```

Exiting the LMM GUI returns the user to the MTC level.

Limitations and restrictions The following limitations and restrictions apply to LMM commands.

- A user can enter commands and parameters in either case. However, enter options (such as -d and -l) in lower case only. The GW name is case sensitive and must be entered exactly in the case in which it appears.
- Due to the potential large output, Nortel recommends specifying only one GW or VMB in a command.
- To enter multiple command on the same line, separate each command with a semicolon (;). A user can specify a maximum of four LENs, DNs, or URIs with the POST, BSY, TST or RTS commands.
- The following LMM commands do not support SIP lines:
  - POST -gw
  - BSY -gw
  - RTS -gw
  - TST -gw

Therefore, avoid using these commands for SIP lines.

Post command

The Post command enables a user to generate a set of lines for maintenance actions. Posting the contents to the screen in the brief format displays the following categories:

- DN
- LEN
- URI
- ClientIP
- GWC
- LCC
- CM line state
- SCC line state
- MCS line state (SIP lines only)

Posting the contents to the screen in the verbose format displays the same information as in the brief format, but includes the additional information categories:

- GW/VMG name
- GW IP address

- GW type
- name of the endpoint
- name of the domain or subdomain

Command usage Enter the following command to post SIP lines by DN.

```
>post -d 7550505

DN          = 6137550505
LEN         = SS 00 0 00 05
LCC         = IBN
GWC NAME    = GWC-5
GW NAME     = rtp7ssa
EPT NAME    = SS/000/0/0005
GW EPT ST   = ON-HOOK
CM EPT ST   = IDL
SIP URI     = user5@rtp7ssa.com
CLIENT IP   =
SS EPT ST   =
Number of endpoints returned = 1
```

Enter the following command to post non-SIP lines by DN.

```
>post -d 6210010

DN          = 6136210010
LEN         = UAIP 00 1 00 00
LCC         = IBN
GWC NAME    = GWC-14
GW NAME     = rtp7-1104b.us.nortel.com
EPT NAME    = aaln/1
GW EPT ST   = ON-HOOK
CM EPT ST   = ID
Number of endpoints returned = 1
```

Enter the following command to post a line by LEN.

```
> post -l ss 0 0 0 5

DN          = 6137550505
LEN         = SS 00 0 00 05
LCC         = IBN
GWC NAME    = GWC-5
GW NAME     = rtp7ssa
EPT NAME    = SS/000/0/0005
GW EPT ST   = ON-HOOK
CM EPT ST   = INB
SIP URI     = user5@rtp7ssa.com
CLIENT IP   =
SS EPT ST   =
```

Enter the following command to post a line by GW.

> ***Note:*** Posting a SIP line by VMG can be time consuming and display an extremely large output due to the minimum VMG size of 1023 endpoints.

```
> post -gw rtp7-1104b.us.nortel.com

DN         = 6136210010
LEN        = UAIP 00 1 00 00
LCC        = IBN
GWC NAME   = GWC-14
GW NAME    = rtp7-1104b.us.nortel.com
GW NAME    = rtp7-1104b.us.nortel.com
EPT NAME   = aaln/1
GW EPT ST  = ON-HOOK
CM EPT ST  =
DN         = 6136210011
LEN        = UAIP 00 1 00 01
LCC        = IBN
GWC NAME   = GWC-14
GW NAME    = rtp7-1104b.us.nortel.com
EPT NAME   = aaln/2
GW EPT ST  = ON-HOOK
CM EPT ST  =
DN         = 6136210012
LEN        = UAIP 00 1 00 02
LCC        = RES
GWC NAME   = GWC-14
GW NAME    = rtp7-1104b.us.nortel.com
EPT NAME   = aaln/3
GW EPT ST  = ON-HOOK
CM EPT ST  =
DN         = 6136210013
LEN        = UAIP 00 1 00 03
LCC        = RES
GWC NAME   = GWC-14
GW NAME    = rtp7-1104b.us.nortel.com
EPT NAME   = aaln/4
GW EPT ST  = ON-HOOK
CM EPT ST  =
```

Enter the following command to post a line by URI.

```
> post -u user5@rtp7ssa.com

DN         = 6137550505
LEN        = SS 00 0 00 05
LCC        = IBN
GWC NAME   = GWC-5
GW NAME    = rtp7ssa
EPT NAME   = SS/000/0/0005
GW EPT ST  = ON-HOOK
CM EPT ST  =
SIP URI    = user5@rtp7ssa.com
```

```
CLIENT IP =
SS EPT ST =
```

Enter the following command to post the contents of the post set using the verbose format.

```
>  post -display verbose
```

```
DN         = 6137550505
LEN        = SS 00 0 00 05
LCC        = IBN
GWC NAME   = GWC-5
GW NAME    = rtp7ssa
EPT NAME   = SS/000/0/0005
GW EPT ST  = ON-HOOK
CM EPT ST  = IDL
SIP URI    = user5@rtp7ssa.com
CLIENT IP  =
SS EPT ST  =
Node Num   = 213
Term Num   = 6
GWC IP     = 47.142.92.172
GW IP      = 47.142.91.154
GW TYPE    =
```

Enter the following command to post the number of entries in the post set.

```
>post -count
```

```
Post set size = 104
```

BSY command

The BSY command enables a user to notify the CM to perform a specific action. The CM then generates a response message indicating whether the command succeeded or failed.

Command usage Enter the following command to maintenance/manually busy a line by DN.

```
>bsy -d 7550505
```

```
DN = 7550505
CM RETURN CODE = CM_OK
```

Enter the following command to maintenance/manually busy all lines on a GW.

```
> bsy -gw rtp7-1104b.us.nortel.com
```

```
DN = 6136210010
CM RETURN CODE = CM_OK
DN = 6136210011
CM RETURN CODE = CM_OK
DN = 6136210012
CM RETURN CODE = CM_OK
```

```
DN = 6136210013
CM RETURN CODE = CM_OK
```

Enter the following command to busy the posted set.

```
> bsy -set
```

```
DN = 6136210010
CM RETURN CODE = CM_OK
DN = 6136210011
CM RETURN CODE = CM_OK
DN = 6136210012
CM RETURN CODE = CM_OK
DN = 6136210013
CM RETURN CODE = CM_OK
```

Enter the following command to busy an idle line. Executing this command on an idle line busies the line and then performs an INB.

```
>  bsy -d 7550505 inb
```

```
DN = 7550505
CM RETURN CODE = CM_OK
DN = 7550505
CM RETURN CODE = CM_OK
```

On the other hand, if the line is already busy, enter the following command to perform an INB on the line.

```
> bsy -d 7550505 inb
```

```
DN = 7550505
CM RETURN CODE = CM_ALRDY_IN_REQ_STATE
DN = 7550505
CM RETURN CODE = CM_OK
```

RTS command

Command usage Enter the following command to return a line to service by DN.

```
> rts -d 7550505
```

```
DN =  7550505
CM RETURN CODE = CM_OK
```

Enter the following command to return a line to service by LEN.

```
>  rts -l ss 0 0 0 5
```

```
LEN = ss 0 0 0 5
CM RETURN CODE = CM_OK
```

Enter the following command to return all lines to service on a GW.

```
> rts -gw rtp7-1104b.us.nortel.com
```

```
DN = 6136210010
CM RETURN CODE = CM_OK
```

```
DN = 6136210011
CM RETURN CODE = CM_OK
DN = 6136210012
CM RETURN CODE = CM_OK
DN = 6136210013
CM RETURN CODE = CM_OK
```

Enter the following command to return the posted set to service.

```
> rts -set
```

```
DN = 6136210010
CM RETURN CODE = CM_OK
DN = 6136210011
CM RETURN CODE = CM_OK
DN = 6136210012
CM RETURN CODE = CM_OK
DN = 6136210013
CM RETURN CODE = CM_OK
```

TST command

Use the TST command to perform a diagnostic test on endpoints. This command supports only GCP and MGCP GWs.

Command usage Enter the following command to run diagnostics on SIP lines by DN.

```
> tst -d 7550505
```

```
DN = 6137550505
LINE_STATE = PASSED
ENDPOINT_STATE = UNKNOWN
CLIENT = user5@47.142.166.65
TEST RESULT = PASSED
```

Enter the following command to run diagnostics on MGCP lines by DN

```
> tst -d 6210010
```

```
DN = 6136210010
LINE_STATE = ON-HOOK
ENDPOINT_STATE = OK
```

Enter the following command to run diagnostics on SIP lines by LEN.

```
> tst -l ss 0 0 0 5
```

```
DN = 6137550505
ENDPOINT_STATE = UNKNOWN
CLIENT = user5@47.142.166.65
TEST RESULT = PASSED
```

Enter the following command to run diagnostics by URI.

```
> tst -u user5@rtp7ssa.com
```

```
URI = user5@rtp7ssa.com
LINE_STATE = PASSED
ENDPOINT_STATE = UNKNOWN
```

```
CLIENT = user5@47.142.166.65
TEST RESULT = PASSED
```

Enter the following commands to run diagnostics by GW or VMG. The first command and output applies to MGCP-based GWs. The second command and output applies to SIP GWs

> ***Note:*** Running diagnostics by VMG can be time consuming and display an extremely large output.

```
> tst -gw rtp7-1104b.us.nortel.com
```

```
DN = 6136210010
LINE_STATE = ON-HOOK
ENDPOINT_STATE = OK
DN = 6136210011
LINE_STATE = ON-HOOK
ENDPOINT_STATE = OK
DN = 6136210012
LINE_STATE = ON-HOOK
ENDPOINT_STATE = OK
DN = 6136210013
LINE_STATE = ON-HOOK
ENDPOINT_STATE = OK
```

Enter the following command to

```
> tst -gw rtp7ssa
```

```
DN = 6137550500
LINE_STATE = FAILED
ENDPOINT_STATE = UNKNOWN
ERROR = Test Executed
DN = 6137550501
LINE_STATE = FAILED
ENDPOINT_STATE = UNKNOWN
ERROR = Test Executed
```

Report command

Used in connection with the GW, the Report command enables the user to list, view and delete trouble reports. By adding the generate (or gen) parameter, the user can create a report.

The following sections show examples of the Report command with its accompanying parameters and options.

Command format

```
report <-l>

report <-v> <REPORT NAME>

report <-gen> <GWC List> | all>
```

Command usage Enter the following command to display a list of currently available reports.

```
>report -l
```

```
The following reports are available:
Report-2005-08-10-22-06-14
```

Enter the following command to view a specific report.

```
>report -v Report-2005-08-10-22-06-14
```

```
GW Name   Results for GWC-0
t1124ent.us.nortel.com   GW DISABLED
GW Name   Results for GWC-10
test4   GW DISABLED
```

Enter the following command to request a new GW in a trouble report for GWC 1.

```
>report -gen 1
```

Enter one of the following commands to request a new GW in a trouble report for GWC 1, 2, and 3.

```
>report -gen 1, 2, 3
```

or

```
>report -gen 1-3
```

or

```
>report -gen 1, 2-3
```

Help command

Used to display available commands within the LMM mode.

Command usage The user can display command-specific help information for each available LMM command - bsy, post, frls, report, rts, and tst.

For example, to display help information on the post command, enter

```
>post -h
```

```
usage:  post
-count Display size of post set
-d {dn} | {dn, dn..} Post DN or multiple DNs (max 4)
-display Display post set
-gw {gw/vmg name} Post all GW/VMG endpoints
-l {len} | {len, len..} Post LEN or multiple LENs (max 4)
-u {uri} | {uri, uri..} Post URI or multiple URIs (max 4)
```

Map Level Service Control Application Programming Interface

This capability is applicable to North American loads only.

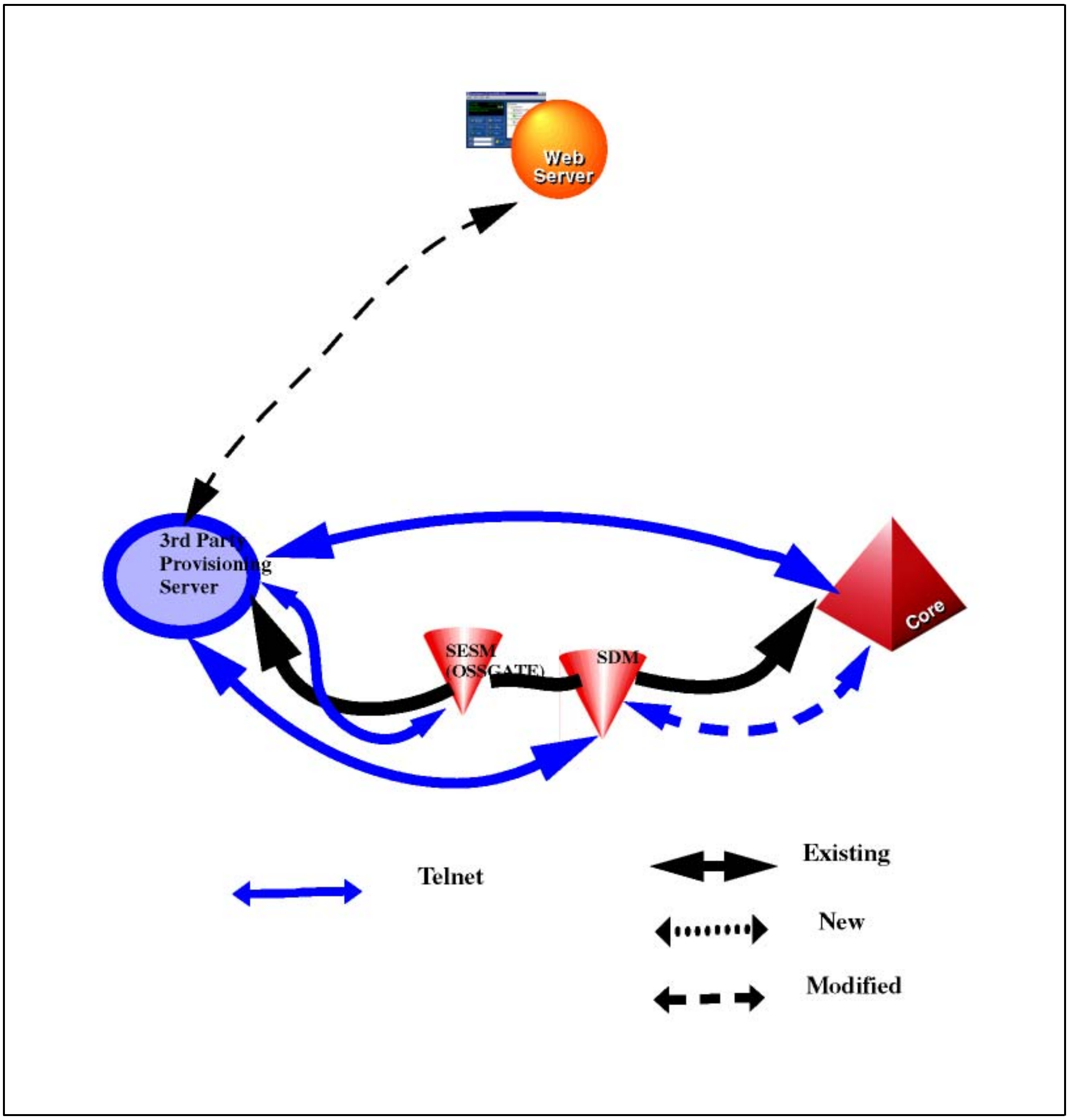
The Map Level Service Control Application Programming Interface addresses the development associated with enabling in DMS and CS 2000, the update and/or query capabilities for the switch based services. With the addition of this set of capabilities, subscribers will have the ability to both query the status and/or programmed information as well as activate/deactivate and update their corresponding service. Based on the service provider’s interface for the end user, it takes out the complexity faced by the user to activate, deactivate, add, delete, change or edit specific data. The command performance will be in the range of 1 to 3 seconds.

Functional Behaviour

For CS 2000, the commands issued by the end user will be sent via a 3rd party provisioning server to the OSSGATE via telnet. This is then sent to the SDM (Supernode Data Manager or CS 2000 Manager) which is the interface to the core for processing. The output will then be sent back and displayed to the end user via the web server. For TDM lines, there is a direct telnet connection between the 3rd party provisioning server and the SDM. This architecture can be used both for DMS and CS 2000.

The Query command

Depending on the feature, the Query command can be used to query different fields such as the status, the list (if present), the delay interval and so on. When issued at the CI prompt, the query command will take the input parameters and call the query procedure for that feature. Depending on whether the feature is present or not, the corresponding return code is output.



QueryRequest

For a query request, the following are the input parameters:

- The DN of the subscriber associated with the given request would be a 10 digit DN.
- The corresponding feature/service would be simple enumerated type similar to the one defined in SERVORD and representing all supported services
- The list of service attributes being queried would be simple set or enumerated type representing all possible attributes

QueryResponse

The response will include the following:

Values corresponding to the service attributes queried

- Only attributes for a single service per query will be supported in (I)SN09.
- For list queries, the privacy indicator should be checked prior to displaying the corresponding DN

List of supported Query Attributes

Supported Query Attributes are listed in the following table:

Enumeration	Return Value	Description
status	active or inactive	Query the status of the corresponding service
list	list of DNs 10 digits in length with/without priv. ind	Query all entries in the corresponding service's screening list.
listSize	positive integer less than 255	Query the number of entries in the corresponding service's screening list
forwardDN	DN upto 30 digits in length	Query the forwarding DN of the corresponding call diversion service
delayInterval	Positive integer between 1 and 10	Query the delay Interval of the corresponding service.This represents the number of rings
scList	List of DNs variable in length up to 30 digits	Query all entries in the corresponding Speed Call list
all	Some aggregate of the above	Query all attributes of the corresponding service

Service to Query attribute matrix

The matrices in the table below show the attributes present for each service.

Service	Status	Delay Interval	List	List Size	Forward DN	Speed Call List
MWI	X					
VMWI	X					
AMWI	X					
ACRJ	X					
ACB	X					
AR	X					
CSMI	X					

Service	Status	Delay Interval	List	List Size	Forward DN	Speed Call List
CWT	X					
LDSA	X					
SUPRESS	X					
MSB	X					
DRCW	X		X	X		
SCRJ	X		X	X		
SIMRING	X		X	X		
CFU	X				X	
CFDA	X				X	
CFB	X				X	
SCA	X		X	X		
SCF	X		X	X	X	
CFDVT		X				
CNDB						
CNAB						
COT						
CCW						
SCS						X
SCL						X
SCU						X

Service to Query attribute matrix

The matrices in the table below, show the attributes present for each service

Return Code	Description
Failure - Invalid_Dn	DN entered is not valid.
Failure -Unavailable_Resources	The service is not subscribed on the DN.
	The feature SOC is idle.
	SLE SOC is idle.
Failure - SOC_Idle	The svcntrl SOC is idle.

Query Examples

Syntax to query the status of SCRJ for 6136631001:

```
>svcntrl query 6136631001 scrj status
```

```
Status - Service_Active
```

Syntax to query the SCRJ list for 6136631001:

```
>svcntrl query 6136631001 scrj list
List_Dn -
6136634567; 6136638901; 6136671001
6136671023; 6136789021; 6136779001
6136772021;
```

Syntax to query SCRJ for 6136631001:

```
>svcntrl query 6136631001 scrj all
Status - Service_Active
List_Dn -
6136634567; 6136638901; 6136671001
6136671023; 6136789021; 6136779001
6136772021;
List_Size - 7
```

Syntax to query the Speed Call List of 6136771052

```
>svcntrl query 6136771052 scl all
Speed Call Code:  12 Dn:  6631021
Speed Call Code:  13 Dn:  6631022
Speed Call Code:  29 Dn:  6136671021
>svcntrl query 6136771052 scl sclist
Speed Call Code:  12 Dn:  6631021
Speed Call Code:  13 Dn:  6631022
Speed Call Code:  29 Dn:  6136671021
```

For a DN with any of the Call Forwarding features like CFB, CFD, SCF, if the feature is subscribed on the DN, but there is no forward DN present, then the return code for a Query command will be ‘ForwardDn - Not_Available’.

```
>svcntrl query 6136671021 cfb forwarddn
ForwardDn - Not_Available
```

If the DN is not provisioned with the service, then the following error message displayed:

```
>Syntax to query status of CSMI on 6136671021
```

```
>svcntrl query 6136671021 CSMI status

Failure - Unavailable_Resources
```

The Query All Command

This command queries all the features and prints the response for each feature that is present on the User Dn. The feature name is not specified in the command. The syntax of the Query All command is:

```
>SVCNTRL QUERY <User DN> ALL
```

This command queries for all the features on a User DN and if any Service Management feature is subscribed on the User Dn, prints out the response. If a feature is not subscribed on the User DN, then no response for that feature is printed out. Only if the User Dn has no feature subscribed on it, the response of ‘Failure - Unavailable_Resources’ is printed.

There are certain conditions when the feature is subscribed on the User DN but returns ‘Failure - Unavailable_Resources’. For these conditions, only if no other feature is subscribed on the User DN, will this error message be printed. Else, it will be ignored and the other feature responses are printed. The following are the conditions:

- For the SLE features, the feature is subscribed on the User DN, but the required datafills in Table CUSTSTN are missing
- For certain features like CFDVT, if the feature specific SOC is idle
- Features belonging to Group 7(CNAB & CNDB) and Group 8(COT & CCW) are not supported for the Query command. So even if these features are present on the User Dn, the Query All command will not consider these features

Some examples of the Query All command:

```
>svcntrl query 9097502513 all

MSB

Status - Service_Inactive

SCRJ

Status - Service_Active

List_Dn -

6136211025; 6136631022;

List_Size - 2

SCA

Status - Service_Active

List_Dn -

6136218001; 6136671001;
```

```
List_Size - 2
```

f no Service Management feature is present on the User Dn, a response of 'Failure - Unavailable_Resources' is returned.

```
>svcntrl query 9097502514 all
```

```
Failure - Unavailable_Resources
```

The Dn 6136218001 has COT, CNAB and CNDB subscribed on it. The response will be Unavailable_Resources since these features are not supported by the Query Command.

```
>svcntrl query 6136218001 all
```

```
Failure - Unavailable_Resources
```

The Update command

Depending on the feature, the Update command can be used to update different attributes such as the status, the delay interval, the forward DN and so on. When issued at the CI prompt, the Update command will take the input parameters and call the Update procedure for that feature. Depending on whether the feature is updated or not, the corresponding return code is output.

Update Request

For an Update request, the following are the input parameters

- The DN of the subscriber associated with the given request will be a 10 digit DN
- The corresponding service will be represented as a simple enumerated type similar to the one defined in SERVORD and representing all supported services
- The action to be taken will be represented as a simple enumerated type representing all supported actions

Update Response

Will include the following:

Return codes indicating non availability of the API (ie is the CI SOC idle)

- The return code will be in the form of readable text

  Return codes indicating the confirmation or denial of the Update request.
- Simple enumerated type representing all supported Update return codes.
- The return code will be displayed as readable text.

For example, confirmation from an Update request to activate CSMI would be represented as ‘Success - Service_Activated’.

List of supported Update Actions

The following table gives the list of actions that is supported for each service.

Enumeration	Description
activate	Activate corresponding service
deactivate	Deactivate corresponding service
delayInterval	Set delay interval for corresponding service.This is the number of rings
adddn	Add specified DN to corresponding service’s screening list
deletedn	Delete specified DN from corresponding service’s screening list
deleteAlldn	Delete all DNs from corresponding service’s screening list
deleteAllPrivdn	Delete all private DNs from corresponding service’s screening list
setfwdDN	Set forwarding DN for corresponding call diversion service
clearFwdDN	Clear forwarding DN for corresponding call diversion service
toggle	Toggle status of the corresponding services
invoke	Invoke the corresponding service
changeList	Change a specified speed call cell entry

Service to Update Action matrix

The matrices in the two tables below show the attributes present for each service.

Service	Activate	Deactivate	Delay Interval	Adddn	Deletedn	DeleteAlldn
MWI	X					
VMWI	X					
AMWI	X					
ACRJ	X					
ACB	X					
AR	X					
CSMI	X					

Service	Activate	Deactivate	Delay Interval	Adddn	Deletedn	DeleteAlldn
CWT	X					
LDSA	X					
MSB	X					
SUPPRESS	X					
DRCW	X	X		X	X	X
SCRJ	X	X		X	X	X
SIMRING	X	X		X	X	X
CFU	X	X				
CFDA	X	X				
CFB	X	X				
SCA	X	X		X	X	X
SCF	X	X		X	X	X
CFDVT			X			
CNDB						
CNAB						
COT						
CCW						
SCS						
SCL						
SCU						

Service	Delete All Privatedn	Set Fwd Dn	Clear FwdDN	Toggle	Invoke	ChangeList
MWI						
VMWI						
AMWI						
ACRJ						
ACB						
AR						
CSMI						
CWT						
LDSA						
MSB						

Service	Delete All Privatedn	Set Fwd Dn	Clear FwdDN	Toggle	Invoke	ChangeList
SUPPRESS						
DRCW	X					
SCRJ	X					
SIMRING						
CFU		X	X			
CFDA		X	X			
CFB		X	X			
SCA	X					
SCF	X	X	X			
CFDVT						
CNDB				X		
CNAB				X		
COT					X	
CCW					X	
SCS						X
SCL						X
SCU						X

Update Examples

Syntax to activate Call Forward BusyLine on 4164731051

```
>svcntrl update 4164731051 cfb activate
Success - Service_Activated
```

Syntax to clear the forward DN on 4164731051

```
>svcntrl update 4164731051 cfb clearFwdDn
Success - ForwardingDn_Cleared
```

Syntax to set the forward DN on 4164731051

```
>svcntrl update 4164731051 cfb setFwdDn dn 4164631001
Success - ForwardingDn_Set
```

Syntax to add a DN to the Speed Call List of 6136771052

```
>svcntrl update 6136771052 scl changelist 12 6631021
Success - Service_Activated
```

```
Error conditions:
```

If we try to activate an already activated service:

```
>svcntrl update 4164731051 cfb activate
Failure - Service_Already_Active
```

If the user tries deleting a list which is empty, the response would be:

```
>svcntrl update 4164731051 simring deleteallDN
Failure - List_Is_Empty
```

If the SOC for SPRING (call forward ringing) is not turned on, the response would be:

```
>svcntrl update 4164731051 cfdvt delayInterval 4
Failure - Unavailable_Resources
```

Ex6: If the parameters entered are insufficient:

```
>svcntrl update
```

The response would be:

```
Failure - Missing_Parameter
```

List of supported Update responses

The following table lists the supported update responses.

Response	Meaning
Success - Service_Activated	Successful activation of the service
Success - Service_Deactivated_or_Cancelled	Successful deactivation of the service
Success - AnonymousEntry_Added	Successful addition of a private no. to the list of a DN
Success - PublicEntry_Added	Successful addition of a DN to the list of another DN
Success - AnonymousEntry_Removed	Successful deletion of a private number from a DNs list
Success - PublicEntry_Removed	Successful deletion of a DN from a DNs list
Success - All_Anonymous_Entries_Removed	Successful deletion of all private DNs from the list
Success - All_Entries_Removed	Successful deletion of all DNs from the list
Success - ForwardingDn_Set	Successful setting of a FwdDN to another DN
Success - ForwardingDn_Cleared	Successful clearing of a FwdDN from another DN

Response	Meaning
Success - DelayInterval_Updated	Successful updation of delay interval of a DN
Failure - Service_Already_Active	Not updated because the service is already active on the DN
Failure - Service_Not_Activated	Gives this response because the service might be inactive and the user is trying to deactivate or in the case of ACB/AR if the feature queue is not present
Failure -Invalid_Forwarding_Dn	FwdDN not set because the FwdDN did not pass validation
Failure - List_Is_Empty	When trying to delete a DN from the list of another DN and if the list is empty
Failure -List_Is_Full	When trying to add a DN to the list of another DN and if the list is full and cannot accommodate more DNs
Failure - Public_Dn_Already_On_List	If the DN you are trying to add to the list of another DN is already present
Failure - Anonymous_Dn_Already_On_List	If the private DN you are trying to add to the list of another DN is already present
Failure - Dn_Not_On_List	If the DN you are trying to delete is not on the list
Failure - No_Match	When trying to update the delay interval, if the ring control is not programmable ring type
	For the SLE features, when trying to delete a private DN from the list of the subscriber DN, if no private DN is present on the list
Failure - Unsuccessful_Update	This message comes when the update is not successful and for different features and actions, are different
	For ACB/AR, activation of the feature is not supported
	For CFB, if Fixed or Programmable version is not provisioned
	For CFBL & CFDA, with control N type
	FOR CFDA if IECFD is provisioned or if CFD Normal is provisioned
	For Speed Call, if SCU is provisioned
Failure - Invalid_Dn	DN entered is not valid
Failure - MSRID_Does_Not_Match_User_Prof ile	For all types of MWT an Msr Id has to be input.This will be validated against Table MSRTAB.If there is a mismatch, then it returns this code

Response	Meaning
Failure - Unavailable_Resources	The service is not subscribed on the DN
	The feature SOC is idle
	For ACB and AR, checks the validity of the feature and if the feature is not allowed gives this response
	For ACRJ, COT if universal access is not permitted
	For CFB, if IECFB is provisioned
	For CFD, if IECFD is provisioned
	For CFU, if CFU/CFI/CFF is not provisioned
	For CNAB & CNDB, if the call is not up
	FOR MWI, if EMW or CALLOG is assigned to the line
	The SLE SOC is idle
	Deactivate MWT when MWT is not active
	For services that use SLE, if SLE datafills are missing in Table CUSTSTN
Failure - SOC_Idle	The SVCNTRLI SOC is idle

SIMRING Virtual DN

A SIMRING Virtual DN can be provisioned using the NEWDN command in SERVORD. It is the only feature in Service Management that can be provisioned on a Virtual DN. On issuing an Update/Query command for a SIMRING Virtual DN, the DN is validated and then the query is sent to the Service Management framework. The following is the response received for a SIMRING Virtual DN:

```
>svcntrl query 6136672000 simring all
```

Status - Service_Inactive

List_Dn -

6136634567; 6136638901; 6136671001

List_Size - 3

```
> svcntrl update 6136672000 simring activate
```

Success - Service_Activated

When the Query/Update command for a SIMRING Virtual DN is given with other features, the response will be Failure - Unavailable_Resources.

```
>svcntrl query 6136672000 msb all
```

Failure - Unavailable_Resources

```
>svcntrl query 6136672000 scf all
```

Failure - Unavailable_Resources

```
> svcntrl update 6136672000 msb activate
```

Failure - Unavailable_Resources

```
> svcntrl update 6136672000 scf adddn dn 6631022
```

Failure - Unavailable_Resources

When the User DN is a Virtual DN of a type other than SIMRING like ACD, AIN etc, then the comand will be blocked at the CI level with a response of Failure - Invalid_Dn.

```
>svcntrl query 6136671000 simring all
```

Failure - Invalid_Dn

```
>svcntrl query 6136671000 msb all
```

Failure - Invalid_Dn

```
> svcntrl update 6136671000 drcw adddn dn 6671001
```

Failure - Invalid_Dn

Error responses for Invalid Entries

Return Value	Description
Failure - Invalid_Action	When any other value other than update or query is entered after svcntrl
Failure - Invalid_DNformat	When the DN entered has alpha numeric values or if the DN is not of 10 digit form
Failure - Unrecognized_Service	When the service entered is invalid
Failure - Invalid_Attribute	When the attribute entered for that service is invalid
Failure - Invalid_Attribute_Parameter	When the parameters for the attributes are specified incorrectly. For e.g., if the value of delay interval to be updated is greater than 10

Ex 1: When the action given is not update/query

```
>svcntrl updation 6136631001 acrj activate
```

The response would be:

```
Failure - Invalid_Action
```

Ex2: If the DN is invalid

```
>svcntrl query abc6790123 drcw list
```

The response would be:

```
Failure - Invalid_DNformat
```

Ex3: If the DN entered is not 10 digit

```
>svcntrl query 6631001 cfb ForwardDN
```

The response would be:

```
Failure - Invalid_DNformat
```

Ex4: If the service is not valid

```
>svcntrl update 6136671021 cssi activate
```

The response would be:

```
Failure - Unrecognized_Service
```

Ex5: If the attribute is not valid for that service

```
>svcntrl update 4164671021 cndb FwdDn
```

The response would be:

```
Failure - Invalid_Attribute
```

Ex6: If the parameters entered are insufficient

```
>svcntrl update
```

The response would be:

```
Failure - Missing_Parameter

>svcntrl query scs

Failure - Missing_Parameter
```

EX7: If the attribute parameters are invalid

```
>svcntrl update 4164671001 cfdvt delayInterval 65
```

The response would be:

```
Failure - Invalid_Attribute_Parameter
```

Hardware Requirements or Dependencies

In order to enable this capability for affected subscribers, the following must be present in the provider’s network:

- Some Provider Network Server designed to support service queries/updates via some Web or PC Client based interface

Software Requirements or Dependencies

The new SOC SMGT0001 will control the CI interface to Service Management. This SOC will be independent of the AIN TCAP Service Management SOC control. The following table gives the possible supported state of both the SOCs:

AIN SOC	Service Management SOC (SMGT0001)	Implication
ON	ON	Both interfaces will be active at the same time
IDLE	IDLE	Both interfaces will be inactive at the same time
ON	IDLE	AIN interface to be active and SVCNTRL to be inactive.
IDLE	ON	AIN interfaces to be inactive and SVCNTRL to be active
ON for some	ON	Some combination of AIN interfaces active and SVCNTRL active
ON for some	IDLE	Some combination of AIN interfaces active and SVCNTRL inactive

Limitations and restrictions

If some invalid characters are entered after a valid command, then the command will not be rejected. The command will be processed with the valid input.

List of supported services/features

The following table gives a list of supported services/features and the market for which they are valid.

Service	SERVORD Acronym
Message Waiting Indicator ***Note:*** North American Market Only	MWI
Visual Message Waiting Indicator	VMWI
Audio Message Waiting Indicator	AMWI
Anonymous Call Rejection	ACRJ
Automatic Callback	ACB
Automatic Recall	AR
Call Screening	CSMI

Service	SERVORD Acronym
Outside Calling Area Alerting	LDSA
Calling ID delivery & Suppression	SUPPRESS
Call Waiting(requires CWTACT)	CWT
Make Set Busy	MSB
Distinctive Ringing Call Waiting ***Note:*** North American Dialplan Only	DRCW
Selective Call Rejection	SCRJ
Simultaneous Ringing	SIMRING
Call Forward Dont Answer ***Note:*** This service will include both CFDA and CFD	CFDA
Call Forwarding Variable	CFU
Call Forward Busy Line ***Note:*** This service will include both CFBL and CFB	CFB
Selective Call Acceptance	SCA
Selective Call Forwarding	SCF
Call Forwarding Ringing Control	CFDVT
Calling Number Delivery Blocking	CNDB
Calling Name Delivery Blocking	CNAB
Customer Originated Trace	COT
Cancel Call Waiting	CCW
Speed Calling Short	SCS
Speed Calling Long	SCL
Speed Calling User	SCU

List of supported DN specifiers

The following table gives the list and description of supported DN specifiers.

Enumeration	Description
user	DN being added or deleted provided explicitly by the subscriber
speed call	Add/Delete the DN in the subscriber’s corresponding speed call entry
icm	Add/Delete the DN in the subscriber’s Incoming Call Memory

Line provisioning with OSSGate

OSSGate is an application that provides a machine interface for provisioning components within Carrier VoIP. One of these interfaces is for Line Provisioning (part of the CS 2000 Management Tools application). In order to enter Lines Provisioning commands, the OSSGate session must be running in CI mode. CI mode is the default mode when first connecting to OSSGate.

The SERVORD+ syntax for provisioning Carrier VoIP lines is very similar to the SERVORD syntax used in the DMS. SERVORD commands that are executed from OSSGate are called SERVORD+ commands. With these commands, the endpoint generally replaces the LEN for Carrier VoIP lines.

For a complete description of the commands and their syntax, please refer to the *SERVORD Reference Manual* (NTP 297-8021-808 for North American and NTP 297-9051-8081 for international)

Some examples are given below for the types of commands that can be entered using OSSGate.

Examples

```
>QDN <DN>

>QLEN <DR_LEN_TYPE>

>QLEN <MGName> <TPname>

>NEW $ <DN> 1FR NILLATA 0 NORT1 0 0 919 <MGName> <TPname>
3WC $ y y

>OUT $ 9917577 <MGName> <TPname>
```

Note: The media gateway name (MGname) and the termination point (TP) name have replaced the line equipment number (LEN) wherever the LEN was used in the SERVORD command. Please note that TP alone may not be unique in the Carrier VoIP call server, but the MG and TP pair will uniquely addresses a subscriber line. This can be seen above in the ‘OUT’ command.

LEN and gateway/termination usage in commands

For most Carrier VoIP lines, the gateway and termination point name replace the LEN in the command string. However, there is one gateway type which supports either gateway/termination names or LENs in the command. For cable (NCS) and MGCP IAD lines this is required. For Centrex IP lines (CICM) only LEN can be used. However, there are some gateway types which supports either gateway/termination names or LENs in the command. These include MG 9000, other large line GW such as Calix, Keymile and some 3rd party GW, and SIP (Session Server-Lines).

- MG 9000

  LENS or gateway/termination names may be used in SERVORD+ commands with certain limitations for MG 9000 lines.

- CICM

  LENS must be used for all (I)SN07 and later format Centrex IP lines. See "CICM flowthrough command support" (page 65) for more details.

DNH Huntgroup DEL command syntax exception

Another change is the modification in the syntax of the DEL command for DNH, DLH, and MLH members. To delete DNH members, users must enter both DNs, MG Name and TP names of the members. The legacy DEL command required only DNs to be entered for the DNH members.

Examples

```
>DEL $ DNH <DN> <MGname> <TPname> $ bldn

>DEL $ DLH <MGname> <TPname> <key> $

>DEL $ MLH <MGname> <TPname> <key> $ y y
```

Line Provisioning - Special Case For Lengthy SERVORD Commands

When executing Line Provisioning commands that are greater than 75 characters long via a telnet session to OSSGate, the commands must be entered following the convention described in this paragraph. If a SERVORD+ command is greater than 75 characters, the user should put up to 75 characters on one line, then enter a ‘+’ symbol and a carriage return (new line) after the 75th character. The remaining portion of the command should be entered on the next line. Using this convention communicates to OSSGate that the command is continued on the next line.

In the following example, the single SERVORD command is split over five lines of input. Each of the first four lines ends with a ‘+’ followed by a carriage return, and the prompt is given back to the user after each carriage return.

Example

```
>EST $ DNH 9195200570 1FR LATA1 0 les24.mgcp.net aaln/16
9195200571+

>les24.mgcp.net aaln/17 9195200572 les24.mgcp.net aaln/18
9195200573+

>les24.mgcp.net aaln/19 9195200574 les24.mgcp.net aaln/20
9195200575+

>les24.mgcp.net aaln/21 9195200576 les24.mgcp.net aaln/22
9195200577+
>les24.mgcp.net aaln/23 9195200578 les24.mgcp.net aaln/24
$ $ 15
```

> ***Note:*** The command when entered at OSSGate can be split at any position.

SERVORD commands supported

Line Provisioning commands and authorization levels describes the commands used for provisioning lines. Line provisioning options describes the line services or options that are available. The commands and options are not described or defined in this document. Refer to the SERVORD Reference Manual (NTP 297-8021-808 for North American and NTP 297-9051-8081 for international) for further details.

MG 9000 flowthrough command support

The NEW, EST, ADD, DEL, OUT, CDN, and CHDN commands update data on the MG 9000 and MG 9000 Manager. The “NEW”, “EST”, and “ADD” commands add the termination point, directory number and indicated line class code. The “DEL”, and “OUT” commands delete the termination point. The “CDN” and “CHDN” commands change the directory number. CLN, CKLN, and CHG commands are also supported. All flow through to the MG 9000 Manager is blocked when the MG 9000 is in an Emergency Stand Alone (ESA). Other conditions within the MG 9000 Manager and MG 9000 may also block the successful process of OSSGate commands. Please refer to MG 9000 documentation for more specific information regarding ESA and other failure modes.

CICM flowthrough command support

SERVORD+ commands are used to ensure that the necessary information is provisioned in the CS 2000 tables and in the CICM EM for CICM lines. This ensures key labelling consistency between the CICM client and the CS 2000 representation of the line. The current flow through provisioning model for CICM lines does NOT flow all SERVORD commands through to the CICM.

> ***Note:*** Hunt group commands (eg. EST/ADD/DEL) do not provision CICM correctly. The use of hunting features on CICM will require separate provisioning via the CICM Manager. The user must manually ensure that the CICM client key labels are updated via the CICM Manager. MADN also requires separate provisioning via CICM Manager.

Line provisioning commands sent to CICM gateway as part of flow through provisioning include: ADO, CDN, CHF, CHL, DEO, NEW, NEWACD, OUT. CICM client specific SERVORD+ commands (NEW, EST, OUT, etc.) may contain USERID and PASSWD option. The USERID and PASSWD options are used to specify CICM client information in the various SERVORD+ commands which create or modify lines. Data fill interaction include:

- Key feature addition overwrites any existing key feature on the CICM terminal. No SERVORD equivalent data fill rules are applied to CICM data fill. CS 2000 SERVORD checking still applies.
- Assignment of multiple features to a single key is not allowed (e.g., 4 rag 4 3WC). Duplicate feature assignments to key 1 only is allowed (e.g., 1 USERID 1234 $ 1 RAG).
- CICM specific key features USERID and PASSWD must be assigned to key 1. Duplicate feature assignments to key 1 only is allowed (e.g., 1 USERID 1234 $ 1 RAG).
- The USERID feature cannot be changed using the CHF command. Use DEO and ADO commands instead. Note that the password is also reset when the ADO command is executed.
- The PASSWD feature can be added, changed or deleted at anytime providing that the USERID feature is provisioned. If the USERID feature is provisioned, but the PASSWD feature has not been provisioned, there is no default PASSWD.
- USERID and PASSWD features will never be sent to the CS 2000 SERVORD interface.
- CICM and CICM EM applications must be running, otherwise the SERVORD+ command will fail. SERVORD+ commands are not saved or queued when CICM communications is unavailable. They will not be executed when connectivity to CICM is restored.

CICM line provisioning examples

- **`NEW $8906917 M5216 CSLINES 0 0 125 1 Y CICM 142 2 00 01 3 3WC 4 ACB NOAMA $ $`**

  ```
  RESULT: features 3WC, ACB, added to CICM LEN
  ```

- **`NEW $8906917 M5216 CSLINES 0 0 125 1 Y CICM 142 2 00 01 3 3WC 4 ACB 1 $ 1  USERID 9999 PROFILE SRV $ 1 PASSWD 1234 $`**

  ```
  RESULT: features 3WC, ACB, USERID, PASSWD added to CICM
  LEN
  ```

- **`ADO $ CICM 142 2 00 01 5 3WC 6 RAG $`**

  ```
  Result:  3WC added to key 5 and RAG added to key 6.
  ```

- **`DEO $ CICM 142 2 00 01 3 3WC 6 RAG $`**

  ```
  Result:  3WC removed from key 3 and RAG removed from key
  6.
  ```

- **`CHF $ CICM 142 2 00 01 5 3WC 6 RAG 7 ACBAMA $`**

  ```
  RESULT: feature 3WC added to key 5, RAG added to key 6,
  and ACBAMA added to key 7.
  ```

- **`OUT $ 1278701234 CICM 142 2 00 01 BLDN`**

  ```
  RESULT: All features and USERID and PASSWD options deleted
  from CICM
  ```

CICM line provisioning responses

The table below lists the new SERVORD command response messages that may be presented to the OSSGate user to indicate a CICM-specific command failure or problem.

Response Message	Cause
USERID option change is not supported. Use DEO and ADO commands.	Entered command (e.g., CHF) cannot be used to change the USERID option data.
{0} restricted to key 1. Example: USERID restricted to key 1.	The option string (represented by {0}) can only be added to key 1.
Multiple {0} options are not allowed. Example: Multiple USERID options are not allowed.	The option string (represented by {0}) can only be added to a single key.
Multiple options assigned to keys {0}. Example: Multiple options assigned to keys 3 and 4.	The key numbers (represented by {0}) can only be associated with a single option.
Multiple options assigned to key {0}. Example Multiple options assigned to key 3.	The key number (represented by {0}) can only be associated with a single option.
USERID option syntax is invalid.	The entered USERID option is not compatible with the syntax
PASSWD option syntax is invalid.	The entered PASSWD option is not compatible with the syntax
Addon {0} is not valid for LCC {1}. Example: Addon M622 is not valid for LCC M5216.	A valid addon was entered but is not compatible with the indicated LCC.
Addon {0} data is invalid. Quantity must be 0 - {1}. Example: Addon M522 data is invalid. Quantity must be 0 - 2.	Addons can be added in groups. The quantity added is dependant on the addon type.
{0} command with LCC = {1}; {2} is required. Example: NEWACD command with LCC = M5678; M5216 is required.	The SERVORD command ({0}) and LCC ({1}) are not valid for the supported LCC.

Response Message	Cause
NEWACD agent type must be SUPERVISOR or AGENT.	Self explanatory
Unable to locate ACD Group information for agent.	The SERVORD command syntax is invalid. The command parser was unable to locate the ACD group field(s).
Unable to locate ACD Group information for supervisor.	The SERVORD command syntax is invalid. The command parser was unable to locate the ACD group field(s).
Failed to define ACD options using template.	Internal application error. Collect details and report to the next level of support. Internal application error. Collect details and report to the next level of support.
Unable to locate template information.	The entered ACD template cannot be found in the KSETKEYS table. Check the table and try again.
LCC {0} is not valid. Example: LCC M5678 is not valid.	The entered Line Class Code is not supported by the Proxy for the entered CICM LEN.
Template {0} can not be used with the {1} ACD type. Example: Template NORTEL can not be used with the AGENT ACD type.	Data in table KSETKEYS is not consistent with the entered ACD type.
Template {0} is not compatible with the {1} LCC. Example: Template NORTEL is not compatible with the M5316 LCC.	Data in table KSETKEYS is not compatible with the entered LCC. Data in table KSETKEYS is not compatible with the entered LCC.
Template {0} has another {1} option. Only one {1} is allowed on an ACD.	User is attempting to add duplicate options using templates.
Timeout waiting for CICM EM response.	An HTTPS connection was made with the IIS web server but the EM response exceeded the provisioned timeout threshold.
Timeout contacting Primary & Secondary CICM EM server.	The proxy was unable to contact either the primary or secondary IIS servers. Check the state of the IIS servers provisioned CICM data on SESM.
End of option area expected. ’$’ not found.	SERVORD command syntax is invalid. The options string or command must be terminated by a ’$’ character.
Unable to parse the SONUMBER.	Self explanatory. SONumber should be ’$’.
Invalid DN entered.	The entered DN, or combination of SNPA and NXX is not valid.
Invalid LEN entered.	The proxy is unable to parse the entered CICM or other LEN.

Response Message	Cause
The NXX of the DN was not found in table TOFCNAME.	Self explanatory.
Rejected by CICM EM gateway. XML Gateway Message failure.	Fatal error. The request sent to the IIS web server was rejected. Collect details and report to the next level of support.
Rejected by CICM EM. EM terminal data corrupted.	The SERVORD request partially failed on the CICM EM and the proxy was unable to successfully undo the changes made.
Command FAILED. Rejected by CICM EM.	The SERVORD request failed on the CICM EM.

SERVORD+ commands for SIP Lines Provisioning

SIP lines can be provisioned using the SIP endpoint or the corresponding LEN. SIP client commands and data are distributed to the Session Server Element Manager (SS-EM) from SERVORD+ in the SESM.

The following commands trigger data distribution to the SS-EM from SESM SERVORD+ when SS vmgs/endpoints or associated LENs are in the command:

- NEW
- CHF
- OUT
- ADD
- EST
- DEL
- ADO (for MADN options)
- DEO (for MADN options)
- CDN
- CHF
- CLN/CTP
- QCUST
- QLEN/QTP
- QDNWRK
- QLENWRK
- QSIP

SIP provisioning information

SIP provisioning information includes the following:

- SIP_PACKAGE - Up to 30 characters.
- SIP_URI - Up to 60 characters.
- SIP_CLIENT_TYPE - The system come with a default value of ‘ONT’. This element can be created on the Prov client. This is used to provision whether the user (sip line) has a static client that it will use to register or not.
- SIP_PASSWD - The administrator defines the password rules. They include the following types or rules:
  - — Have a minimum password length between 4 and 10.
  - — Have a minimum of 0 to 10 numerical characters.
  - — Have a minimum of 0 to 10 non-numerical characters.
  - — Consist of ASCII characters 0x20 through 0x73 hexadecimal or 32 through 126 decimal
- SIP_LOCATION - The location name, a brief description of the location being added. This field cannot be null. Maximum length for a single location name is 30 characters. However, a fully qualified location name can consist of more than one dot-delimited location values.

  For example:
  Nortel.Research Triangle Park.NC0.column 1Z20

  In this example, no location value is greater than 30 characters, although the entire fully qualified location name is greater than 30 characters.
- SIP_SUBDOMAIN - Enter the name of the new local domain. The domain name must not be more then 60 characters in length and cannot contain the following symbols or characters:
- ‘ $ ( ) | ; ~ % [ ] / ! ^ { } ? @ & + “ , # * = <>

  Note: Values for suboptions SIP_CLIENT, SIP_PACKAGE, and SIP_LOCATION can be entered in either upper or lower case in the SERVORD+ commands. However, always enter these associated values in lower case when commissioning them on the Session Server Provisioning Manager. OSSGate automatically converts these values from the SERVORD+ commands into lower case before they are transmitted to the Session Server Provisioning Manager. Failure to commission these values in lower case in the Session Server Provisioning Manager results in command failures a the OSSGate SERVORD+.

NEW command

The following NEW command examples show SERVORD output reflecting SIP-data options:

-
  ```
  NEW $ 6195209998 1FR LATA1 0 SCOT 00 0 00 00 dgt
  SIP_DATA SIP_PACKAGE SIP Lines SIP_URI slynch@mordor.com
  SIP_CLIENT_TYPE SIP Line SIP_LOCATION Nortel.RTP.NC0
  SIP_PASSWD scott11 $ DPL Y 10 $
  ```

-
  ```
  NEW $ 6195209998 1FR LATA1 0 vmg1 SCOT/000/0/0000 dgt
  SIP_DATA SIP_PACKAGE SIP Lines SIP_URI slynch@mordor.com
  SIP_CLIENT_TYPE SIP Line SIP_LOCATION Nortel.RTP.NC0
  SIP_PASSWD scott11 $ DPL Y 10 $
  ```

CHF command

Command CHF supports SIP lines and SIP_DATA except for SIP_URI, which can be included in the command but must match the existing value assigned to the user. Command CHF also supports changing option SIP_SUBDOMAIN for the user.

This command also requires the use of gateway/termination names (or the equivalent LEN) to trigger data flowthrough for SS lines. Use of a DN instead of a primary LEN results in the CM only processing the command.

The following CHF command examples show SERVORD output reflecting SIP-data options:

-
  ```
  CHF $ SCOT 00 0 00 00 SIP_DATA SIP_PACKAGE siplines
  SIP_PASSWD scott11 SIP_CLIENT_TYPE IBN SIP_LOCATION
  IBM.RTP $ $
  ```

-
  ```
  CHF $ vmg1 SCOT/000/0/0000 SIP_DATA SIP_PACKAGE siplines
  SIP_PASSWD scott11 SIP_CLIENT_TYPE IBN SIP_LOCATION
  IBM.RTP $ $
  ```

Use command CHF to move a line or a user between domains and subdomains. The following example shows how a provisioned SIP_URI (slynch@mordor.com) in the root domain (mordor.com) can be moved to SIP_SUBDOMAIN (doom.mordor.com).

```
CHF $ SCOT 00 0 00 00 SIP_DATA SIP_SUBDOMAIN doom.mordor.com
$$
```

Specifying a sub-option SIP_SUBDOMAIN, the user associated with the LEN (SCOT 00 0 00 00) moves from its current domain to the value specified in suboption SIB_SUBDOMAIN. The use of the CHF command moves a user between the root domain and a subdomain, between subdomains, or between root domains.

ATTENTION

User moves between domains can affect user-based data in the SSEM. If packages, locations, or group information do not match in the target domains, the existing user data is overwritten by the default data in the target domain.

No warning displays if this situation occurs. Therefore, use caution when executing CHF-based user moves since important domain-specific data can be irretrievably lost. (Execute commands QLEN or QDN to locate any data mismatches or loss.)

OUT command

The following OUT command examples show SERVORD output reflecting SIP-data options:

- `OUT $ 6195209998 SCOT 00 0 00 00 BLDN`
- `OUT $ 6195209998 vmg1 SCOT/000/0/0000 BLDN`

ADO and DEO commands with MADN options

SIP lines are generally unaffected by the use of commands ADO and DEO. However, adding an MDN option to an existing line can change the DN of the line. Any existing user DN information is set to an empty value when the user is added to an MDN group. Adding option MDN can also trigger group creation on the session server if the line is the first session server line in the MADN group.

The following command examples show SERVORD output reflecting SIP-data options:

```
NEW $7215555 IBN BNR 0 0 619 NILLATA 0 C411 01 0 02 00
SIP_DATA SIP_PACKAGE SIP Lines SIP_URI user2@mordor.com
SIP_CLIENT_TYPE SIP Line SIP_LOCATION Nortel Networks.RTP.NC0
SIP_PASSWD user21 $ DPL Y 10 dgt $
```

```
ADO $ C411 01 0 02 00 MDN SCA Y Y 6197215555 TONE N $
```

C411 01 0 02 00 is the primary member of the MADN group 6197215555. MADN group 6197215555 is created on the SSEM. User user2@mordor.com is added to the group. The DN of user2@mordor.com is cleared out in the SSEM.

The following command shows how a second line is added with a different DN - 6197215554.

```
NEW $7215554 IBN BNR 0 0 619 NILLATA 0 C411 01 0 02
1SIP_DATA SIP_PACKAGE SIP Lines SIP_URI user3@mordor.com
SIP_CLIENT_TYPE SIP Line SIP_LOCATION Nortel Networks.RTP.NC0
SIP_PASSWD user21 $ DPL Y 10 dgt $
```

The following command shows how option MDN is added to the second line specifying a different DN (6197215555) and a treatment (BLDN) for the original DN (6197215554).

```
ADO $ C411 01 0 02 01 MDN SCA NY 6197215555 BLDN $
```

The result of the previous command is that the DN for C411 01 0 02 01 changes from 6197215554 to 6197215555 on the Core. The line becomes a member of MADN group 6197215555 on the SSEM.

The following command shows how option MDN is deleted from the newly added member and its new DN is specified.

```
DEO  $ C411 1 0 2 01 MDN SCA NY 6197215555 BLDN $
```

The result of deleting option MDN in the previous command is that the DN for C411 01 0 02 01 changes from 6197215555 to 6197215554 on the Core. The line is no longer a member of MADN group 6197215555 in the SSEM. User3@mordor.com is assigned DN 6197215554 on the SSEM.

NEW, OUT and CDN commands with MADN options

The following commands show how to create and populate MADN groups using the NEW command.

```
NEW $ 7215555 IBN BNR 0 0 619 NILLATA 0 C411 01 0 02 01 DGT
MDN SCA Y Y TONE N SIP_DATA SIP_PACKAGE SIP Lines SIP_URI
user1@mordor.com SIP_CLIENT_TYPE SIP Line SIP_LOCATION
Nortel Networks.RTP.NC0 SIP_PASSWD user11 SIP_SUBDOMAIN
doom.mordor.com $ $
```

During processing of this NEW command, a group with type MDN and name 6197215554 is created. User1@mordor.com also is added to the group.

```
NEW $ 7215555 IBN BNR 0 0 619 NILLATA 0 C411 01 0 02 2
DGT MDN SCA N Y SIP_DATA SIP_PACKAGE SIP Lines SIP_URI
user2@mordor.com SIP_CLIENT_TYPE SIP Line SIP_LOCATION
Nortel Networks.RTP.NC0 SIP_PASSWD user11 SIP_SUBDOMAIN
doom.mordor.com $ $
```

The second NEW command adds user2@mordor.com to the same group.

```
CDN $ 6197215554 6197215555 BLDN
```

The CDN command changes the DN for the MADN group in the Core for all members. This command also changes the group name from 6197215554 to 6197215555 in the SSEM.

```
OUT 7215555 C411 01 0 02 02 BLDN
```

```
OUT $ 7215555 C411 01 0 02 01 BLDN
```

The OUT command removes the line/user from the MADN group on the SSEM. If the user is the last member of the group, the group is removed also.

EST, ADD, DEL and CHDN commands for Hunt Groups

Hunt groups trigger flowthrough for hunt flavors only which assign service to hardware-assigned, software-unassigned (HASU) lines:

- Distributed Line Hunt (DLH)
- Multiline Hunt (MLH)
- Directory Number Hunt (DNH)

Multiple Position Hunt (MPH) also can assign service to an HASU line. However, SIP lines do not support MPH. Features, such as Call Pickup (CPU), CMG, SIMRING, and Bridged Night Number (BNN), which are assigned to lines with existing service, do not trigger flowthrough. No groups are created nor deleted on the SSEM.

> ***Note:*** Commands EST, ADD and DEL support only single SIP LENs/endpoints. Processing of one of these commands fails if it includes multiple SIP LENs/endpoints.

DLH The following commands show how DHL group 6195209998 is created on the Session Server.

```
EST $ DLH 6195209998 1FR LATA1 0 SCOT 00 0 00 00 $ dgt
SIP_DATA SIP_PACKAGE SIP Lines SIP_URI user1@mordor.com
SIP_CLIENT_TYPE SIP Line SIP_LOCATION Nortel Networks.RTP.NC0
SIP_PASSWD user11 SIP_SUBDOMAIN DOOM.MORDOR.COM $ DPL Y 10 $
```

In the previous command, EST creates the session server group, and user1@mordor.com is added to the group.

```
ADD $ DLH SCOT 00 0 00 00 SCOT 00 0 00 01 $ dgt SIP_DATA SIP
_PACKAGE SIP Lines SIP_URI user2@mordor.com SIP_CLIENT_TYPE
SIP Line SIP_LOCATION Nortel Networks.RTP.NC0 SIP_PASSWD
user21 SIP_SUBDOMAIN doom.mordor.com $ DPL Y 10 $
```

The ADD command does not trigger a group creation, but user2@mordor.com is added to group 6195209998 on the SSEM.

```
DEL $ DLH SCOT 00 0 00 01 $
```

The DEL command removes user2@mordor.com from the group and from the SSEM.

```
OUT $ 6195209998 SCOT 00 0 00 00 BLDN
```

The OUT command removes group 6195209998 from the SSEM and removes user1@mordor.com.

MLH The following commands show how MLH group 6195209998 is created on the Session Server.

```
EST $ MLH 6195209998 1FR LATA1 0 SCOT 00 0 00 00 $ dgt
SIP_DATA SIP_PACKAGE SIP Lines SIP_URI user1@mordor.com
SIP_CLIENT_TYPE SIP Line SIP_LOCATION Nortel Networks.RTP.NC0
SIP_PASSWD user11 SIP_SUBDOMAIN doom@mordor.com $ DPL Y 10 $
```

The EST command creates the session server group. User1@mordor.com is added to the group.

```
ADD $ MLH SCOT 00 0 00 00 SCOT 00 0 00 01 $ dgt SIP_DATA SIP
_PACKAGE SIP Lines SIP_URI user2@mordor.com SIP_CLIENT_TYPE
SIP Line SIP_LOCATION Nortel Networks.RTP.NC0 SIP_PASSWD
user21 SIP_SUBDOMAIN doom.mordor.com $ DPL Y 10 $
```

The ADD command does not trigger a group creation. User2@mordor.com is added to group 6195209998 on the SSEM.

```
CHDN $ SCOT 00 0 00 01 2145200999
```

The CHDN command changes the DN of member SCOT 00 0 00 00 TO 2145200999 in the Core. No change to the currently blank DN assigned to user2@mordor.com occurs in the SSEM. The user’s DN remains blank in the session server as long as it is a member of a group, regardless of DN changes in the Core.

DNH The following commands show how DNH group 6195209998 is created on the Session Server.

```
EST $ DNH 2145200998 1FR LATA1 0 SCOT 00 0 00 00 $ dgt
SIP_DATA SIP_PACKAGE SIP Lines SIP_URI user1@mordor.com
SIP_CLIENT_TYPE SIP Line SIP_LOCATION Nortel Networks.RTP.NC0
SIP_PASSWD user11 SIP_SUBDOMAIN doom@mordor.com $ DPL Y 10
$ 5
```

The EST command creates the session server group. User1@mordor.com is added to the group.

```
ADD $ DNH 2145200998 2145200997 SCOT 00 0 00 01 $ dgt
SIP_DATA SIP_PACKAGE SIP Lines SIP_URI user2@mordor.com
SIP_CLIENT_TYPE SIP Line SIP_LOCATION Nortel Networks.RTP.NC0
SIP_PASSWD user21 SIP_SUBDOMAIN doom.mordor.com $ DPL Y 10
$ 5
```

The EST and ADD commands support only one SIP LEN/GW-EP since only one SIP_DATA is allowed in the options area of the commands. The ADD or EST command fails if you enter multiple SIP LEN/GW-EP options.

```
DEL $ DNH 2145200997 SCOT 00 0 00 01 $ BLDN
```

The SERVORD+ format for the DEL command requires insertion of GW/EP or LEN for DNH groups, unlike the Core SERVORD.

CDN command

SIP lines support flowthrough of change DN requests. However, the CDN command is blocked for MLH/DHL pilots and members. The CDN command supports MADN and DNH group pilots, links, and members.

```
NEW $ 7215555 IBN BNR 0 0 619 NILLATA 0 C411 01 0 02 00
SIP_DATA SIP_PACKAGE SIP Lines SIP_URI user2@mordor.com
SIP_CLIENT_TYPE SIP Line SIP_LOCATION Nortel Networks.RTP.NC0
SIP_PASSWD user21 $ DPL Y 10 dgt $
```

```
CDN $ 6197215555 6197215554 BLDN
```

The CDN commands that modify the user’s DN do not autormatically update a SIP_URI containing DN information.

> ***Note:*** Nortel does not recommend the use of a DN in the value of SIP_URI (for example, 2145209999@nortel.com).

CLN command

The OSSGate SERVORD+ supports a limited flowthrough to process SIP lines using the CLN command. The flowthrough is restricted to changes between SIP lines only. The CLN command fails for processing SIP lines if the from_value and the to_value do not both contain SIP LENS/endpoints.

The following examples show processing SIP lines with the CLN command.

```
CLN $ C411 01 0 02 00 C411 01 0 02 01
```

```
CLN $ C411 00 00 00 C411 01 0 02 01
```

Query commands

Query commands QCUST, QLENWRK, and QDNWRK, which return formatted line and service information, include SIP_DATA options provisioned through commands NEW, EST, ADD, and CHF.

The following list includes SIP_DATA options that display at the end of the output for these query command:

- END POINT DATA: Data header without any information or value
- SIP_CLIENT_TYPE
- SIP_EP_NAME
- SIP_VMG_NAME
- SIP_DN
- SIP_LOCATION
- SIP_PACKAGE
- SIP_URI
- SIP_GROUP_DN

- SIP_GROUP_TYPE
- SIP_SUBDOMAIN

The following examples show SIP_DATA options at the end of this query command output.

```
> QLEN TEST 01 1 09 22

LEN: TEST 01 1 09 22
END POINT: vmg1 TEST/001/1/0922
TYPE: MEMBER OF MLH HUNT GROUP
SNPA: 214
HUNT GROUP: 13 HUNT MEMBER: 1
DIRECTORY NUMBER: 5209998
LINE CLASS CODE: 1FR
IBN TYPE: STATION
CUSTGRP: RES214 SUBGRP: 0 NCOS: 1
SIGNALLING TYPE: DIGITONE
LINE TREATMENT GROUP: 0
LINE ATTRIBUTE INDEX: 520_1FR
XLAPLAN KEY: 214_P520_1 RATEAREA KEY : RA214_2
CARDCODE: RDTLSG GND: N PADGRP: PKNIL BNV: NL MNO: Y
PM NODE NUMBER : 213
PM TERMINAL NUMBER : 923
OPTIONS:
DGT
RES OPTIONS: NONE
PILOT LEN: TEST 01 1 09 23
GROUP OPTIONS:
RCVD
END POINT DATA:
SIP_CLIENT_TYPE: sip line
SIP_EP_NAME: TEST/001/1/0922
SIP_VMG_NAME: vmg1
SIP_DN: 6195209998
SIP_LOCATION: nortel networks.rtp.nc0
SIP_PACKAGE: sip lines
SIP_URI: slynch@mordor.com
SIP_SUBDOMAIN: doom.mordor.com
SIP_GROUP_TYPE: MLH
SIP_GROUP_DN: 2145209999
```

```
> qdn 5209998

DN: 5209998
TYPE: SINGLE PARTY LINE
SNPA: 619 SIG: DT LNATTIDX: 0
LINE EQUIPMENT NUMBER: SCOT 00 0 00 00
END POINT: vmg1 SCOT/000/0/0000
LINE CLASS CODE: 1FR
LINE TREATMENT GROUP: 0
CARDCODE: RDTLSG GND: N PADGRP: PKNIL BNV: NL MNO: Y
PM NODE NUMBER : 87
PM TERMINAL NUMBER : 1
OPTIONS:
DGT DPL Y 10
OFFICE OPTIONS:
SRA
END POINT DATA:
SIP_CLIENT_TYPE: sip line
SIP_EP_NAME: SCOT/000/0/0000
SIP_VMG_NAME: vmg1
SIP_DN: 6195209998
SIP_LOCATION: nortel
networks.rtp.nc0
SIP_PACKAGE: sip lines
SIP_URI: slynch@mordor.com
SIP_SUBDOMAIN: doom.mordor.com
```

The following example shows SIP_DATA options for HASU LEN, reflecting associated endpoint information at the end of the query output.

```
>qlen TEST 00 1 01 00
```

```
-------------------------------------------------------------
-------------------------------------------------------
TYPE: HARDWARE ASSIGNED SOFTWARE UNASSIGNED
CARDCODE: RDTLSG GND: PADGRP: PKLNL BNV: NL MNO: N
PM NODE NUMBER : 136
PM TERMINAL NUMBER : 101
END POINT: ssvmg.42 TEST/000/1/0100
-------------------------------------------------------------
-------------------------------------------------------
```

The following example shows SIP_DATA options at the end of this output resulting from query command QCUST with record type 23.

```
>qcust bnr $
```

```
1 SS12 00 6 00 07 IBN PKLNL RDTLSG N
2 7207167 BNR 0 0 613
9 CWT DGT DPL Y 10 $
23 SIP_URI me@mydomain.com SIP_LOCATION my house
99
```

The following examples show SIP_DATA options at the end of the output resulting from bulk query commands QDNWRK and QLENWRK.

```
QDNWRK R 7215554 7215554 NLCC $ D
COMMAND AS ENTERED
QDNWRK R 7215554 7215554 ALL NLCC $ ALL D
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT
>REPORT ON WORKING DIRECTORY NUMBERS
FROM 6197215554 TO 61972165554
```

```
LCC ALL OPTION ALL
-------------------------------------------------------------
-----------------------
DN: 7215554
TYPE: SINGLE PARTY LINE
SNPA: 619 SIG: DT LNATTIDX: N/A
LINE EQUIPMENT NUMBER: C411 01 0 02 01
END POINT: vmg1 C411/001/0/0210
LINE CLASS CODE: IBN
IBN TYPE: STATION
CUSTGRP: BNR SUBGRP: 0 NCOS: 0
CARDCODE: RDTLSG GND: N PADGRP: PKLNL BNV: NL MNO: N
PM NODE NUMBER: 147
PM TERMINAL NUMBER: 50
OPTIONS:
DGT
END POINT DATA:
SIP_CLIENT_TYPE: sip line
SIP_EP_NAME: C411/001/0/0201
SIP_VMG_NAME: vmg1
SIP_DN: 6197215554
SIP_LOCATION: nortel networks.rtp.nc0
SIP_PACKAGE: sip lines
SIP_SUBDOMAIN: doom.mordor.com
SIP_URI: slynch@mordor.com
----------------------------------------------------------
---------------------
TOTAL COUNT OF WORKING DN FROM 6197215554 TO 61972155554:  1
>
COMMAND AS ENTERED
QLENWRK R C411 01 0 C411 01 0 ALL ALL NLCC $ D
ENTER Y TO CONFIRM, N TO REJECT OR E TO EDIT
>Y
WARNING: Queries of all LENS or a large range of LENs may run
for 30 minutes or more before producing any output.
Please confirm ("YES", "Y", "NO", "N"):
>Y
>REPORT ON WORKING LINE EQUIPMENT NUMBERS
FROM C411 01 0 TO C411 01 0 DRAWERS ALL
LCC ALL OPTION ALL
----------------------------------------------------------
---------
LEN: C411 01 0 02 01
TYPE: SINGLE PARTY LINE
DN 6197215554 LCC IBN SIG DT LNATTIDX N/A
LINE EQUIPMENT NUMBER: C411 01 0 02 01
IBN TYPE: STATION
CUSTGRP: BNR SUBGRP: 0 NCOS: 0
CARDCODE: RDTLSG GND: N PADGRP: PKLNL BNV: NL MNO: N
PM NODE NUMBER: 147
PM TERMINAL NUMBER: 50
OPTIONS:
DGT
END POINT DATA:
SIP_CLIENT_TYPE: sip line
SIP_EP_NAME: C411/001/0/0201
SIP_VMG_NAME: vmg1
SIP_DN: 6197215554
SIP_LOCATION: nortel networks.rtp.nc0
SIP_PACKAGE: sip lines
SIP_URI: slynch@mordor.com
SIP_SUBDOMAIN: doom.mordor.com
```

SIP SERVORD examples

See below for SIP SERVORD examples:

SERVORD NEW Example using SIP Endpoint

```
new $ 6136040004 ibn ibntst 0 0 lata1 0 sprintnlclocation1
+ sip_data sip_package basic package sip_uri
scott@nortelnetworks.com + sip_client_type ont
sip_location other2 sip_passwd scott11  + sip_subdomain
sm1.nortelnetworks.com $ dpl y 10 dgt cwt
```

SERVORD NEW Example using LEN

```
new $ 6136040004 ibn ibntst 0 0 lata1 0 ss 000 6 00
04 + sip_data + sip_package basic package sip_uri
scott@nortelnetworks.com + sip_client_type ont
sip_location other2 sip_passwd scott11 sip_subdomain
sm1.nortelnetworks.com $ dpl y 10 dgt cwt $
```

Limitations and restrictions

The following limitations and restrictions apply to provisioning SIP lines in SERVORD+:

- OSSGate has limited validation on SIP_DATA SIP_SUBDOMAIN. The Session Server Element Manager is responsible for validating the actual contents.
- Queries with SIP_URI information always include the domain formatted as <user>@<domain>. If a subdomain value is added in suboption SIP_URI in the provisioning command, subsequent queries display the SIP_URI with the domain (<user>@<domain>).
- Nortel discourages use of the DN in the SIP_URI value (for example, 2145559999@nortel.com) Command CDN, which modifies the DN assigned to the user, does not automatically update the SIP_URI value.
- Failure to adhere to the following rules results in immediate command failure or rejection:
  - — At least one suboption is required when specifying option SIP_DATA.

- — Suboption tags are reserved words. Do not use suboption tags as values for other suboptions or for non-related types (such as, CM customer groups, gateway names).

  Example: "SIP_PASSWORD sip_password" is an invalid option and value pair. Do not use "sip_password" as the value for suboption SIP_PASSWORD. Do not use "sip_password" as a part or whole value for any other suboption (such as, 'SIP_LOCATION sip_passwd").
- — SIP_DATA terminator $ is a reserved token. Do not use this terminator as a value for a SIP_DATA suboption. If a suboption exists in the SIP_DATA options list, it must include a value other than $.
- — The tag/value pair relationship for options and suboptions is strictly enforced. All suboptions must include associated values. The complete SIP_DATA option must terminate with a $.
- — OSSGate and SERVORD+ rejects multiple and extraneous SIP_DATA options or suboptions when found in a single command.
- — One is the maximum number of SIP LENs/endpoints allowed in an EST/ADD group command.
- — SIP_URI information must be in the format user@domain. The domain information is parsed from the URI value. A missing domain (such as a missing @ delimiter or domain information) results in a command failure. The Session Server Provisioning Manager rejects an invalid domain name entry.
- — The SIP_DATA option must include SIP_LOCATION when the command results in a new user is added on the Session Server.
- — Suboption SIP_SUBDOMAIN is optional. If it is not present, the domain parsed from the URI is used when a new user is added on the Session Server element manager.

Line Provisioning commands and authorization levels

In addition to users belonging to 'succssn' group to login to OSSGate, user need to be in application specific authorization groups to perform specific operations. Each Line provisioning command is associated with one or more authorization groups. In order to execute a command, a user must belong to at least one of the associated authorization groups. The groups associated with each OSSGate Node provisioning operation are specified in the table below.

In the table below, the Legacy column indicates if the command may be used on Legacy Lines via the OSSGate interface. The Carrier VoIP column indicates if the command may be used on Carrier VoIP Lines via the OSSGate interface.

Table legend

- (I)SN0x - indicates the release in which the command is first supported.
- N/S - indicates the command is not supported
- N/A - indicates the command is not applicable in the given column
- No termination point info - in output from query commands, the data will show LEN (logical group) info, but not the corresponding termination point name.
- ISDN - indicates the command is only supported for ISDN phone sets
- MMP - indicate the command is supported in International markets only

Number	Command	Legacy	Carrier VoIP	Authorization Group	Note
1	ABNN	(I)SN05	(I)SN03	Insprov	
2	ADA	(I)SN05	(I)SN05	Insprov	
3	ADD	(I)SN05	(I)SN03	Insprov	
4	ADDPH	(I)SN05	N/A	Insprov	ISDN
5	ADO	(I)SN05	(I)SN03	Insprov	
6	BULK	N/A	N/A	Insprov	
7	CAPSORD	(I)SN06	N/A	Insprov	ISDN
8	CBLKDN	(I)SN06	(I)SN06	Insprov	
9	CDN	(I)SN05	(I)SN05	Insprov	
10	CHDN	(I)SN05	(I)SN03	Insprov	
11	CHAPH	(I)SN06	N/A	Insprov	ISDN
12	CHF	(I)SN05	(I)SN03	Insprov	
13	CHG	(I)SN05	N/S	Insprov	
14	CHL	(I)SN05	(I)SN05	Insprov	
15	CICP	(I)SN05	(I)SN05	Insprov	
16	CISG	(I)SN05	N/A	Insprov	
17	CKLN	(I)SN05	(I)SN05	Insprov	
18	CLN	(I)SN05	(I)SN05	Insprov	
19	CLTG	(I)SN05	(I)SN05	Insprov	
20	COPYSET	(I)SN05	N/S	Insprov	
21	CTP	N/A	(I)SN05	Insprov	
22	CKTP	N/A	(I)SN05	Insprov	
23	DBNN	(I)SN05	(I)SN03	Insprov	
24	DEA	(I)SN05	(I)SN03	Insprov	

Numbe r	Command	Legacy	Carrier VoIP	Authorizati on Group	Note
25	DEL	(I)SN05	(I)SN03	Insprov	
26	DELCF	N/S	N/S	Insprov	MMP
27	DELPH	(I)SN06	N/A	Insprov	ISDN
28	DEO	(I)SN05	(I)SN03	Insprov	
29	DSP	(I)SN05	N/S	Insprov	
30	ECHO	N/S	N/A	Inro	
31	EST	(I)SN05	(I)SN03	Insprov	See restrictions
32	EXBADD	N/S	N/S	Insprov	
33	EXBADO	N/S	N/S	Insprov	
34	EXBCHG	N/S	N/S	Insprov	
35	EXBDEL	N/S	N/S	Insprov	
36	EXBDELG	N/S	N/S	Insprov	
37	EXBDELM	N/S	N/S	Insprov	
38	EXBEDO	N/S	N/S	Insprov	
39	EXBEST	N/S	N/S	Insprov	
40	MQDN	N/S	N/S	Insprov	Wireless only
41	NEW	(I)SN05	(I)SN03	Insprov	
42	NEWACD	(I)SN05	(I)SN06	Insprov	
43	NEWDN	(I)SN05	(I)SN05	Insprov	
44	OUT	(I)SN05	(I)SN03	Insprov	
45	OUTDN	(I)SN05	(I)SN05	Insprov	
46	PLP	(I)SN05	(I)SN05	Insprov	
47	QX75	(I)SN06	N/A	Inro	ISDN
48	QBB	(I)SN06	N/A	Inro	ISDN
49	QBERT	(I)SN05	N/S	Inro	
50	QCM	(I)SN05	(I)SN06	Inro	
51	QCOUNTS	(I)SN05	(I)SN05	Inro	
52	QCPUGNO	(I)SN05	(I)SN06	Inro	
53	QCUST	(I)SN06	(I)SN06	Inadm	
54	QDCH	(I)SN06	(I)SN06	Inro	ISDN
55	QDN	(I)SN05	(I)SN03	Inro	
56	QDNA	(I)SN05	(I)SN05	Inro	
57	QDNSU	(I)SN06	(I)SN06	Inadm	

Numbe r	Command	Legacy	Carrier VoIP	Authorizati on Group	Note
58	QDNWRK	(I)SN06	(I)SN06	Inadm	
59	QGRP	(I)SN05	(I)SN06	Inro	
60	QHA	(I)SN06	(I)SN06	Inadm	
61	QHASU	(I)SN06	(I)SN06	Inadm	
62	QHLR	N/S	N/S	Inro	Wireless only
63	QHU	(I)SN06	(I)SN06	Inadm	
64	QIT	(I)SN06	N/A	Inro	ISDN
65	QLEN	(I)SN03	(I)SN03	Inro	
66	QLENWRK	(I)SN06	(I)SN06	Inadm	
67	QLOAD	(I)SN06	(I)SN06	Inadm	
68	QLRN	(I)SN05	(I)SN05	Inro	
69	QLT	(I)SN05	N/A	Inro	
70	QMADN	(I)SN06	(I)SN06	Inadm	
71	QMODEL	(I)SN05	(I)SN05	Inro	
72	QMSB	(I)SN05	(I)SN05	Inro	
73	QNCOS	(I)SN06	(I)SN06	Inadm	
74	QPDN	(I)SN06	(I)SN06	Inadm	
75	QPHG	(I)SN05	N/A	Inro	
76	QPRIO	(I)SN05	(I)SN05	Inro	
77	QSCONN	(I)SN06	N/A	Inro	ISDN
78	QSCUGNO	(I)SN05	(I)SN06	Inro	
79	QSIMR	(I)SN05	(I)SN05	Inro	
80	QSI	(I)SN05	(I)SN06	Inro	
81	QTOPSPOS	(I)SN05	(I)SN05	Inro	
82	QTP	N/A	(I)SN04	Inro	
83	QWUCR	(I)SN05	(I)SN05	Inro	
84	RES	(I)SN05	(I)SN05	Insprov	
85	RESGRP	(I)SN05	(I)SN05	Insprov	
86	SDNA	(I)SN05	(I)SN05	Insprov	
87	SETPH	(I)SN06	N/A	Insprov	ISDN
88	SLT	(I)SN04	N/A	Insprov	
89	STOPECHO	N/S	N/A	Insprov	
90	SUS	(I)SN05	(I)SN05	Insprov	

Numbe r	Command	Legacy	Carrier VoIP	Authorizati on Group	Note
91	SUSGRP	(I)SN05	(I)SN05	Insprov	
92	SWAP	N/S	N/S	Insprov	
93	SWLT	(I)SN06	N/A	Insprov	ISDN

Line provisioning options

In the table below, the Legacy column indicates if the option may be used on Legacy Lines via the OSSGate interface. The Carrier VoIP column indicates if the option may be used on Carrier VoIP Lines via the OSSGate interface. Not all options are valid for all line class codes. For more details, refer to the appropriate SERVORD NTPs.

Table legend

- SN0x - indicates the release in which the command is first supported.
- N/S - indicates the command is not supported
- N/A - indicates the command is not applicable in the given column
- ISDN - indicates the command is only supported for ISDN phone sets
- MMP - indicate the command is supported in International markets only

Note: Any options not listed in the above table are considered not supported via the OSSGate Line Provisioning interface.

Number	Option	Legacy	Carrier VoIP	Notes
1	1MMS	(I)SN05	N/S	
2	3WC	(I)SN05	(I)SN05	
3	3WCPUB	(I)SN05	(I)SN05	
4	AAB	(I)SN05	(I)SN05	
5	AAK	(I)SN05	(I)SN05	
6	ABR	(I)SN05	(I)SN05	
7	ACB	(I)SN05	(I)SN05	
8	ACCB	(I)SN05	(I)SN05	
9	ACD	(I)SN05	(I)SN05	
10	ACDNR	(I)SN05	(I)SN05	
11	ACOU	(I)SN05	N/A	ISDN
12	ACR	(I)SN05	N/A	ISDN
13	ACRJ	(I)SN05	(I)SN05	
14	ADSI	(I)SN05	(I)SN05	

Number	Option	Legacy	Carrier VoIP	Notes
15	ADSL	(I)SN06	N/S	
16	AEMK	(I)SN05	(I)SN05	
17	AFC	(I)SN05	N/A	ISDN
18	AGA	(I)SN06	N/A	ISN (Using SLT command)
19	AIN	(I)SN05	(I)SN05	
20	AINDENY	(I)SN06	(I)SN06	
21	AINDN	(I)SN05	(I)SN05	
22	AINMWT	(I)SN05	(I)SN05	
23	AIOD	(I)SN05	(I)SN05	
24	ALI	(I)SN06	(I)SN06	
25	AMATEST	(I)SN05	(I)SN05	
26	AMSG	(I)SN05	(I)SN05	
27	AMSGDENY	(I)SN05	(I)SN05	
28	AOC	(I)SN06	(I)SN06	MMP
29	APS	(I)SN05	(I)SN05	
30	AR	(I)SN05	(I)SN05	
31	ARDDN	(I)SN05	(I)SN05	
32	ARLNA	(I)SN05	(I)SN05	
33	ARR	(I)SN06	N/A	ISDN
34	ASL	(I)SN05	(I)SN05 (see note)	
35	ASP	(I)SN05	(I)SN05	
36	ATC	(I)SN05	(I)SN05	
37	AUD	(I)SN05	(I)SN05	
38	AUL	(I)SN05	(I)SN05	
39	AUTODISP	(I)SN05	(I)SN05	
40	AUTOHF	N/S	N/S	ISDN MFT sets
41	AVT	N/S	N/S	
42	BBGI	(I)SN06	N/A	ISDN
43	BC	(I)SN05	(I)SN05	
44	BCLID	(I)SN05	N/S (See note)	
45	BLF	(I)SN05	(I)SN05	
46	BLOCKCDN	(I)SN06	N/A	ISDN

Number	Option	Legacy	Carrier VoIP	Notes
47	BLOCKCGN	(I)SN06	N/A	ISDN
48	BNN	(I)SN05	(I)SN05	See restrictions
49	BRICLID	(I)SN06	(I)SN06	MMP
50	CACH	(I)SN06	N/A	ISDN (using SLT command)
51	CAG	(I)SN05	(I)SN05	
52	CALLOG	(I)SN05	(I)SN05	
53	CARR	(I)SN06	(I)SN06	MMP
54	CARRG	(I)SN06	(I)SN06	MMP
55	CBE	(I)SN05	(I)SN05	
56	CBI	(I)SN05	(I)SN05	
57	CBU	(I)SN05	(I)SN05	
58	CCBS	(I)SN05	ISN05	Not used in NA. For Intl markets only
59	CCNR	(I)SN06	(I)SN06	
60	CCSA	(I)SN05	(I)SN05	
61	CCV	(I)SN05	(I)SN05	
62	CCW	(I)SN05	(I)SN05	
63	CCWB	(I)SN06	(I)SN06	MMP
64	CD	(I)SN06	(I)SN06	MP
65	CD0-CD9	(I)SN05	N/S	
66	CDC	(I)SN05	(I)SN05	
67	CDE	(I)SN05	(I)SN05	
68	CDF	(I)SN05	(I)SN05	LCC
69	CDI	(I)SN05	(I)SN05	
70	CDND	(I)SN06	(I)SN06	MMP
71	CDT	(I)SN05	(I)SN05	
72	CDTA	(I)SN06	(I)SN06	MMP
73	CDTO	(I)SN06	(I)SN06	MMP
74	CDU	(I)SN05	(I)SN05	
75	CEPT	(I)SN06	(I)SN06	MMP
76	CFB	(I)SN05	(I)SN05	
77	CFBL	(I)SN05	(I)SN05	
78	CFD	(I)SN05	(I)SN05	

Number	Option	Legacy	Carrier VoIP	Notes
79	CFDA	(I)SN05	(I)SN05	
80	CFDVT	(I)SN05	(I)SN05	
81	CFF	(I)SN05	(I)SN05	
82	CFFA	(I)SN06	(I)SN06	MMP
83	CFFPOVR	(I)SN05	(I)SN05	
84	CFGD	(I)SN05	(I)SN05	
85	CFGDA	(I)SN05	(I)SN05	
86	CFI	(I)SN05	(I)SN05	
87	CFIND	(I)SN05	(I)SN05	
88	CFK	(I)SN05	(I)SN05	
89	CFMDN	(I)SN05	(I)SN05	
90	CFO	(I)SN05	(I)SN05	
91	CFRA	(I)SN05	(I)SN05	
92	CFS	(I)SN05	(I)SN05	
93	CFTANN	(I)SN05	(I)SN05	
94	CFTB	(I)SN05	(I)SN05	
95	CFTD	(I)SN05	(I)SN05	
96	CFU	(I)SN05	(I)SN05	
97	CFW	(I)SN05	(I)SN05	
98	CFXDNCT	(I)SN06	N/A	ISDN
99	CFXVAL	(I)SN06	N/A	ISDN
100	CFWANN	(I)SN05	(I)SN05	
101	CHD	(I)SN05	(I)SN05	
102	CHG	(I)SN06	N/A	ISDN
103	CIDSDLV	(I)SN05	N/S	ISUP
104	CIDSSUP	(I)SN05	N/S	ISUP
105	CIF	(I)SN05	(I)SN05	
106	CIR	(I)SN05	(I)SN05	
107	CLF	(I)SN05	(I)SN05	
108	CLI	(I)SN05	(I)SN05	
109	CLNT900	(I)SN05	(I)SN05	
110	CLSUP	(I)SN05	(I)SN05	
111	CMCF	(I)SN05	(I)SN05	
112	CMD	(I)SN06	N/A	ISDN

Number	Option	Legacy	Carrier VoIP	Notes
113	CMG	(I)SN05	(I)SN05	
114	CNAB	(I)SN05	(I)SN05	
115	CNAMD	(I)SN05	(I)SN05	
116	CND	(I)SN05	(I)SN05	
117	CNDB	(I)SN05	(I)SN05	
118	CNDBO	(I)SN05	(I)SN05	
119	CNDDTMF	(I)SN06	(I)SN06	MMP
120	CNF	(I)SN05	(I)SN05	
121	COD	(I)SN05	(I)SN05	
122	COLP	(I)SN06	(I)SN06	
123	COLR	(I)SN06	(I)SN06	MMP
124	COLROVR	(I)SN06	(I)SN06	MMP
125	COT	(I)SN05	(I)SN05	
126	COTAMA	(I)SN05	(I)SN05	
127	CPC	(I)SN06	(I)SN06	MMP
128	CPH	(I)SN05	(I)SN05	
129	CPR	(I)SN05	N/S	
130	CPU	(I)SN05	(I)SN06	
131	CRA	(I)SN06	(I)SN06	MMP
132	CRBL	(I)SN06	N/A	ISDN
133	CRT	(I)SN05	(I)SN05	
134	CRTDENY	(I)SN05	(I)SN05	
135	CSMI	(I)SN05	(I)SN05	
136	CTD	(I)SN05	(I)SN05	
137	CTW	(I)SN05	(I)SN05	
138	CUG	(I)SN05	(I)SN06	
139	CUSD	(I)SN05	(I)SN05	
140	CW	(I)SN06	(I)SN06	MMP
141	CWD	(I)SN05	(I)SN05	
142	CWI	(I)SN05	(I)SN05	
143	CWO	(I)SN05	(I)SN05	
144	CWR	(I)SN05	(I)SN05	
145	CWT	(I)SN05	(I)SN05	
146	CWTACT	(I)SN05	(I)SN05	

Number	Option	Legacy	Carrier VoIP	Notes
147	CWTC	(I)SN05	(I)SN05	
148	CWX	(I)SN05	(I)SN05	
149	CXR	(I)SN05	(I)SN05	
150	DASK	(I)SN05	(I)SN05	
151	DBC	(I)SN06	N/A	ISDN
152	DCBI	(I)SN05	(I)SN05	
153	DCBX	(I)SN05	(I)SN05	
154	DCC	(I)SN06	N/A	ISDN
155	DCF	(I)SN05	(I)SN05	
156	DCPK	(I)SN05	(I)SN05	
157	DCPU	(I)SN05	(I)SN05	
158	DCPX	(I)SN05	(I)SN05	
159	DDN	(I)SN05	(I)SN05	
160	DDI	(I)SN06	(I)SN06	MMP
161	DELFTERM	(I)SN06	N/A	ISDN (using SLT command)
162	DENY	(I)SN05	(I)SN05	
163	DENYAR	(I)SN05	(I)SN05	
164	DENYCSMI	(I)SN05	(I)SN05	
165	DENYCTFP	(I)SN05	(I)SN05	
166	DENYCWTC	(I)SN05	(I)SN05	
167	DENYISA	(I)SN05	(I)SN05	
168	DENYSRA	(I)SN05	(I)SN05	
169	DENYU3WC	(I)SN05	(I)SN05	
170	DENYWUCR	(I)SN06	(I)SN06	MMP
171	DGT	(I)SN05	(I)SN05	
172	DIN	(I)SN05	(I)SN05	
173	DISCTO	(I)SN05	N/S	
174	DISP	(I)SN05	(I)SN05	
175	DLH	(I)SN05	(I)SN05	
176	DMCT	(I)SN06	(I)SN06	MMP
177	DNA	(I)SN06	N/A	ISDN (using SETPH, ADDPH, DELPH, and CHAPH commands)

Number	Option	Legacy	Carrier VoIP	Notes
178	DND	(I)SN05	(I)SN05	
179	DNH	(I)SN05	(I)SN05	
180	DOR	(I)SN05	(I)SN05	
181	DP	(I)SN05	N/S	
182	DPCAR	(I)SN06	N/A	ISDN
183	DPR	(I)SN05	N/S	
184	DQS	(I)SN05	(I)SN05	
185	DQT	(I)SN05	(I)SN05	
186	DRCW	(I)SN05	(I)SN05	
187	DRING	(I)SN05	(I)SN05	
188	DROP	(I)SN06	N/A	ISDN
189	DSCWID	(I)SN05	(I)SN05	
190	DTM	(I)SN05	(I)SN05	
191	EB0	(I)SN05	(I)SN05	
192	EBX	(I)SN05	(I)SN05	
193	ECM	(I)SN05	(I)SN05	
194	ECONF	(I)SN06	(I)SN06	MMP
195	ECT	(I)SN06	(I)SN06	MMP
196	ECTPTOP	(I)SN06	(I)SN06	MMP
197	EHLD	(I)SN05	N/A	ISDN
198	EKTS	(I)SN06	N/A	ISDN (using SLT command)
199	ELN	(I)SN05	(I)SN05	
200	EMK	(I)SN05	(I)SN05	
201	EMR	(I)SN06	(I)SN06	MMP
202	EMW	(I)SN05	(I)SN05	
203	ENQ	(I)SN06	(I)SN06	MMP
204	ESL	(I)SN05	(I)SN05	
205	EWAL	(I)SN05	(I)SN05	
206	EWALI	(I)SN05	(I)SN05	
207	EXT	(I)SN05	(I)SN05	
208	FAA	(I)SN05	(I)SN05	
209	FANI	(I)SN05	(I)SN05	
210	FC	(I)SN06	N/A	ISDN

Number	Option	Legacy	Carrier VoIP	Notes
211	FCTDINT	(I)SN05	(I)SN05	
212	FCTDNTER	(I)SN05	(I)SN05	
213	FCTDNTRA	(I)SN05	(I)SN05	
214	FDN	N/S	N/S	MMP
215	FGA	(I)SN05	(I)SN05	
216	FIG	(I)SN05	(I)SN05	
217	FNO	(I)SN05	(I)SN05	
218	FNT	(I)SN05	(I)SN05	
219	FPS	(I)SN06	(I)SN06	
220	FRO	(I)SN06	(I)SN06	
221	FRS	(I)SN06	(I)SN06	
222	FSR	(I)SN05	(I)SN05	
223	FTRGRP	(I)SN05	(I)SN05	
224	FTRKEYS	(I)SN05	(I)SN05	
225	FTS	(I)SN05	(I)SN05	
226	FXR	(I)SN05	(I)SN05	
227	GIAC	(I)SN05	(I)SN05	
228	GIC	(I)SN05	(I)SN05	
229	GLITE	N/S	N/S	
230	GLTC	(I)SN05	(I)SN05	
231	GND	(I)SN05	(I)SN05	
232	HF	N/S	N/S	ISDN MFT sets.
233	HFMUTE	N/S	N/S	ISDN MFT sets.
234	HLD	(I)SN05	(I)SN05	
235	HOLD	(I)SN06	(I)SN06	MMP
236	HOT	(I)SN05	(I)SN05	
237	HTL	N/S	N/S	MMP
238	I3WC	(I)SN06	(I)SN06	MMP
239	I6WC	(I)SN06	(I)SN06	MMP
240	ICM	(I)SN05	(I)SN05	
241	ICR	(I)SN06	(I)SN06	MMP
242	ICSDEACT	(I)SN05	(I)SN05	
243	ICT	(I)SN06	(I)SN06	MMP
244	ICWT	(I)SN06	(I)SN06	MMP

Number	Option	Legacy	Carrier VoIP	Notes
245	IECFB	(I)SN05	(I)SN05	
246	IECFD	(I)SN05	(I)SN05	
247	IICB	(I)SN06	(I)SN06	MMP
248	ILB	(I)SN05	(I)SN05	
249	ILDCHNL	N/S	N/A	
250	ILR	(I)SN06	(I)SN06	MMP
251	IMB	(I)SN05	(I)SN05	
252	INSPECT	(I)SN05	(I)SN05	
253	INT	(I)SN05	(I)SN05	
254	INTPIC	(I)SN05	(I)SN05	
255	IRR	(I)SN05	(I)SN05	
256	ISA	(I)SN05	(I)SN05	
257	ISADEACT	(I)SN05	(I)SN05	
258	ISDNAMA	(I)SN06	N/A	ISDN
259	IWUC	(I)SN06	(I)SN06	MMP
260	JOIN	(I)SN05	(I)SN05	
261	KSH	(I)SN05	(I)SN05	
262	KSMOH	(I)SN05	(I)SN05	
263	LAPB	(I)SN06	NA	ISDN (using SETPH, ADDPH, DELPH, and CHAPH commands)
264	LAPD	(I)SN06	N/A	ISDN (using SETPH, ADDPH, DELPH, and CHAPH commands)
265	LCDR	(I)SN05	(I)SN05	
266	LDA	(I)SN06	(I)SN06	MMP
267	LDSA	(I)SN05	(I)SN05	
268	LDSO	(I)SN05	(I)SN05	
269	LDSR	(I)SN05	(I)SN05	
270	LDST	(I)SN05	(I)SN05	
271	LDTPSAP	N/S	N/S	
272	LINEPSAP	(I)SN05	(I)SN05	
273	LMOH	(I)SN05	(I)SN05	
274	LNPTST	(I)SN05	(I)SN05	

Number	Option	Legacy	Carrier VoIP	Notes
275	LNR	(I)SN05	(I)SN05	
276	LNRA	(I)SN05	(I)SN05	
277	LOB	(I)SN05	(I)SN05	
278	LOD	(I)SN05	(I)SN05	
279	LOR	(I)SN05	(I)SN05	
280	LPIC	(I)SN05	(I)SN05	
281	LROA	N/S	N/S	MMP
282	LRN	(I)SN06	(I)SN06	MMP
283	LRS	N/S	N/S	MMP
284	LSPAO	(I)SN05	(I)SN05	
285	LSPSO	(I)SN05	(I)SN05	
286	LVM	(I)SN05	(I)SN05	
287	M518	(I)SN05	(I)SN05	
288	M522	(I)SN05	(I)SN05	
289	M536	(I)SN05	(I)SN05	
290	M622	(I)SN05	(I)SN05	
291	MAN	(I)SN05	N/S	
292	MBK	(I)SN05	(I)SN05	
293	MBSCAMP	(I)SN05	(I)SN05	
294	MCH	(I)SN05	(I)SN05	
295	MCID	(I)SN06	(I)SN06	MMP
296	MDN	(I)SN05	(I)SN05	
297	MDNNAME	(I)SN05	(I)SN05	
298	MEMDISP	(I)SN05	(I)SN05	
299	MLAMP	(I)SN05	(I)SN05	
300	MLH	(I)SN05	(I)SN05	
301	MMI	(I)SN06	(I)SN06	
302	MOG	(I)SN05	(I)SN05	
303	MPB	(I)SN05	(I)SN06	
304	MPH	(I)SN06	N/S	
305	MREL	(I)SN05	(I)SN05	
306	MRF	(I)SN05	(I)SN05	
307	MRFM	(I)SN05	(I)SN05	
308	MSB	(I)SN05	(I)SN05	

Number	Option	Legacy	Carrier VoIP	Notes
309	MSBI	(I)SN05	(I)SN05	
310	MSGDEACT	(I)SN05	(I)SN05	
311	MTR	(I)SN06	(I)SN06	MMP
312	MUTE	N/S	N/S	ISDN MFT sets.
313	MWI	(I)SN06	(I)SN06	MMP
314	MWIDC	(I)SN05	(I)SN05	
315	MWINK	(I)SN06	(I)SN06	
316	MWQRY	(I)SN05	(I)SN05	
317	MWT	(I)SN05	(I)SN05	
318	NAME	(I)SN05	(I)SN05	
319	NCCW	(I)SN05	(I)SN05	
320	NEMD	(I)SN06	(I)SN06	MMP
321	NDC	(I)SN05	(I)SN05	
322	NDNAP	(I)SN06	N/A	ISDN
323	NFA	(I)SN05	N/A	
324	NGTSRVCE	(I)SN05	(I)SN05	
325	NHT	(I)SN05	N/S	
326	NLT	(I)SN05	(I)SN05	
327	NOCOLL	(I)SN05	(I)SN05	
328	NODNY	N/A	N/A	Sub-option of SDN option.
329	NOH	(I)SN05	(I)SN05	
330	NOCOLPSCR	(I)SN06	(I)SN06	MMP
331	NPGD	(I)SN05	N/S	
332	NRS	(I)SN05	N/S	
333	NSDN	(I)SN06	N/S	
334	NTAIT	(I)SN06	(I)SN06	MMP
335	NUMC	(I)SN06	N/A	ISDN
336	OBS	(I)SN05	(I)SN05	
337	OCB	(I)SN06	(I)SN06	MMP
338	OCFA	(I)SN05	(I)SN05	
339	ODB	(I)SN06	N/A	ISDN
340	OFR	(I)SN06	(I)SN06	
341	OFS	(I)SN05	(I)SN05	

Number	Option	Legacy	Carrier VoIP	Notes
342	OLS	(I)SN05	(I)SN05	
343	ONI	(I)SN05	(I)SN05	
344	OUTWT	(I)SN05	(I)SN05	
345	PBL	(I)SN05	(I)SN05	
346	PCA	(I)SN06	(I)SN06	MMP
347	PCACIDS	(I)SN05	N/S	ISDN
348	PCI	(I)SN06	(I)SN06	MMP
349	PDO	(I)SN05	(I)SN05	
350	PF	(I)SN05	(I)SN05	
351	PIC	(I)SN05	(I)SN05	
352	PILOT	(I)SN05	(I)SN05	
353	PLP	(I)SN05	(I)SN05	
354	PMC	(I)SN05	(I)SN05	
355	PMD	(I)SN06	(I)SN06	
356	PORT	(I)SN05	(I)SN05	
357	PPH	(I)SN06	(I)SN06	MMP
358	PPL	(I)SN05	(I)SN05	
359	PREMTBL	N/S	N/S	Sub-option of SDN option.
360	PRH	(I)SN05	(I)SN05	
361	PRI	(I)SN05	(I)SN05	
362	PRK	(I)SN05	(I)SN05	
363	PRL	(I)SN05	(I)SN05	
364	PROVCDS	(I)SN06	N/A	ISDN
365	PROVCGS	(I)SN06	N/A	ISDN
366	PROVHLC	(I)SN06	N/A	ISDN
367	PROVLLC	(I)SN06	N/A	ISDN
368	PRV	(I)SN05	(I)SN05	
369	PTP	N/S	N/S	MMP
370	PVC	(I)SN06	N/A	ISDN (using SLT command)
371	QBS	(I)SN05	(I)SN05	
372	QCK	(I)SN05	(I)SN05	
373	QTD	(I)SN05	(I)SN05	
374	RAG	(I)SN05	(I)SN05	

Number	Option	Legacy	Carrier VoIP	Notes
375	RATEAREA	(I)SN05	(I)SN05	
376	RCD	(I)SN05	(I)SN05	
377	RCHD	(I)SN05	(I)SN05	
378	RCVD	(I)SN05	(I)SN05	
379	REASDSP	(I)SN05	(I)SN05	
380	REATTACH	N/S	N/S	
381	RESL	(I)SN06	(I)SN06	MMP
382	REV	(I)SN06	(I)SN06	MMP
383	RLS	(I)SN06	(I)SN06	
384	RND	(I)SN06	N/A	ISDN
385	RMB	(I)SN05	(I)SN05	
386	RMI	(I)SN05	(I)SN05	
387	RMP	(I)SN05	(I)SN05	
388	RMR	(I)SN05	(I)SN05	
389	RMS	(I)SN05	(I)SN05	
390	RMT	(I)SN05	(I)SN05	
391	RPA	(I)SN05	(I)SN05	
392	RSP	(I)SN05	(I)SN05	
393	RSUS	(I)SN05	(I)SN05	
394	SACB	(I)SN05	(I)SN05	
395	SBLF	(I)SN05	(I)SN05	
396	SC1	(I)SN05	(I)SN05	
397	SC2	(I)SN05	(I)SN05	
398	SC3	(I)SN05	(I)SN05	
399	SCA	(I)SN05	(I)SN05	
400	SCF	(I)SN05	(I)SN05	
401	SCI	(I)SN05	(I)SN05	
402	SCL	(I)SN05	(I)SN05	
403	SCMP	(I)SN05	(I)SN05	
404	SCR	(I)SN06	(I)SN06	MMP
405	SCRJ	(I)SN05	(I)SN05	
406	SCS	(I)SN05	(I)SN05	
407	SCU	(I)SN05	(I)SN05	
408	SCWID	(I)SN05	(I)SN05	

Number	Option	Legacy	Carrier VoIP	Notes
409	SDN	(I)SN05	(I)SN05	
410	SDS	(I)SN05	(I)SN05	
411	SDSDENY	(I)SN05	(I)SN05	
412	SDY	(I)SN05	(I)SN05	
413	SEC	(I)SN05	(I)SN05	
414	SETMODEL	(I)SN05	(I)SN05	
415	SHU	(I)SN06	(I)SN06	
416	SIMRING	(I)SN05	(I)SN05	
417	SKDISP	N/S	N/S	ISDN MFT set.
418	SL	(I)SN05	(I)SN05	
419	SLBRI	(I)SN06	N/A	ISDN (using SLT command)
420	SLC	(I)SN05	(I)SN05	
421	SLQ	(I)SN05	(I)SN05	
422	SLU	(I)SN05	(I)SN05	
423	SLVP	(I)SN05	(I)SN05	
424	SMDI	(I)SN06	(I)SN06	
425	SMDICND	N/S	N/S	
426	SMDR	(I)SN05	(I)SN05	
427	SND	N/S	N/S	Wireless
428	SNUMC	N/S	N/S	MMP
429	SOFTKEY	N/S	N/S	ISDN MFT sets.
430	SOR	(I)SN05	(I)SN05	
431	SORC	(I)SN05	(I)SN05	
432	SPB	(I)SN05	(I)SN05	
433	SPL	N/S	N/S	MMP
434	SPM	N/S	N/S	MMP
435	SRA	(I)SN05	(I)SN05	
436	SSAC	(I)SN05	(I)SN05	
437	STRD	(I)SN05	(I)SN05	
438	SURCOM	(I)SN06	(I)SN06	MMP
439	SUPPRESS	(I)SN05	(I)SN05	
440	SUPPRND	(I)SN06	N/A	ISDN
441	SUPR	(I)SN05	(I)SN05	

Number	Option	Legacy	Carrier VoIP	Notes
442	SUS	(I)SN05	(I)SN05	
443	SVCGRP	(I)SN05	(I)SN05	
444	TBO	(I)SN05	(I)SN05	
445	TCW	(I)SN06	(I)SN06	
446	TDN	(I)SN05	(I)SN05	
447	TDV	(I)SN05	(I)SN05	
448	TERM	(I)SN05	(I)SN05	
449	TERML	(I)SN06	N/A	ISDN (using SLT command)
450	TES	(I)SN05	(I)SN05	
451	TFO	(I)SN05	(I)SN05	
452	TLS	(I)SN05	(I)SN05	
453	TRANSFER	(I)SN06	N/A	ISDN
454	TRMBOPT	(I)SN05	(I)SN05	
455	TSPID	(I)SN06	N/A	ISDN (using SLT command)
456	UCD	(I)SN05	(I)SN05	
457	UCDLG	(I)SN05	(I)SN05	
458	UCDLI	(I)SN05	(I)SN05	
459	UCDSD	(I)SN05	(I)SN05	
460	UUSI	(I)SN06	(I)SN06	MMP
461	VAD	(I)SN06	(I)SN06	
462	VI	(I)SN06	N/A	ISDN
463	VIS	(I)SN06	(I)SN06	MMP
464	VMEADENY	(I)SN05	(I)SN05	
465	VMEADN	(I)SN05	(I)SN05	
466	VMI	(I)SN06	(I)SN06	MMP
467	VOW	(I)SN06	(I)SN06	MMP
468	VOWDN	(I)SN06	(I)SN06	MMP
469	WC	(I)SN05	(I)SN05	
470	WLN	N/S	N/S	MMP
471	WML	(I)SN05	(I)SN05	
472	WUCR	(I)SN05	(I)SN05	

Number	Option	Legacy	Carrier VoIP	Notes
473	XFER	(I)SN06	N/A	ISDN
474	XLAPLAN	(I)SN05	(I)SN05	

> ***Note 1:*** SERVORD+ query commands do not display termination point data for this option.
>
> ***Note 2:*** Manual work-around provisioning procedure exists.
>
> ***Note 3:*** Any options not listed in the above table are considered not supported through the OSSGate Line Provisioning interface.

Restricted queries

The following commands are restricted Queries. Only one restricted query is allowed to run via OSSGate on the CS 2000 management tool server at a given time.

For example: If one user issues a QDNWRK query via OSSGate and a second user issues a QLENWRK query via a second OSSGate session before the QDNWRK query has finished processing, the QLENWRK query request will be denied, with a message "The maximum number of concurrent instances of this command has been reached. Please try later".

- QCUST
- QDNSU
- QDNWRK
- QHA
- QHASU
- QHU
- QMADN
- QNCOS
- QLENWRK
- QLOAD
- QPDN

In some instances it may desirable to stop the processing of a restricted query. To stop the processing of a restricted query the KILL command may be issued either via OSSGate or directly from the Core.

> ***Note:*** The KILL command should be used with extreme caution. It has the potential to kill all of a certain type of process (e.g. all the QDNWRK commands running on the Core).

The following is a list of the restricted queries and their corresponding process names on the Core with example KILL command syntax.

Query	Process Name	Example Syntax to Stop Command

QCUST	QCUST	`kill QCUST only # <process ID>`
		`kill QCUST all`

QDNSU	SOQDNSU	`kill SOQDNSU only # <process ID>`
		`kill SOQDNSU all`

QDNWRK	SOQDWRK	`kill SOQDWRK only # <process ID>`
		`kill SOQDWRK all`

QHA	SOQHA	`kill SOQHA only # <process ID>`
		`kill SOQHA all`

QHASU	SOQHASU	`kill SOQHASU only # <process ID>`
		`kill SOQHASU all`

QHU	SOQHU	`kill SOQHU only # <process ID>`
		`kill SOQHU all`

QLENWRK	SOQLWRK	`kill SOQLWRK only # <process ID>`
		`kill SOQLWRK all`

QLOAD	SOQLOAD	`kill SOQLOAD only # <process ID>`
		`kill SOQLOAD all`

QMADN	MDNQUERY	`kill MDNQUERY only # <process ID>`
		`kill MDNQUERY all`

QNCOS	SOQNCOS	`kill SOQNCOS only # <process ID>`
		`kill SOQNCOS all`

QPDN	QPDNCI	`kill QPDNCI only # <process ID>`
		`kill QPDNCI all`

Stopping a restricted query

In some instances it may desirable to stop the processing of a restricted query. To stop the processing of a restricted query the KILL command may be issued either via OSSGate or directly from the Core.

> ***Note:*** The KILL command should be used with extreme caution. It has the potential to kill all of a certain type of process (for example, all the QDNWRK queries running on the Core).

Determining the process name of a command

At the Core, issue the query command to determine the process name of the restricted query.

```
Syntax:  query alias <command>

>query alias qdnwrk

SOQDWRK  EC=EF01  MODREF=28BA  PERPROCESS
SOQRYSUB  ORIGINAL
protected:  address=42CD6180  size=00C0 words
public size=0000 words extension size=0000 words
shared:     not allocated
private:    not allocated
scratch:    not allocated
entry: QUERY_DN  offset=0000  increment of CIPROC
alias: QDNWRK
SOQDWRK is the process name of the QDNWRK query.
```

Determining the process IDs of a process

At the core, issue the query command on a process name to determine the process IDs running.

```
Syntax:  query process <process name>

>query process SOQDWRK

496C02C0: 451F 208A SOQDWRK   class=BKG    slice=2
lock=3  unprot=0  queued on flag
451F 208A is the process ID of the SOQDWRK (i.e.  the QDNWRK
command) process running.
```

Stopping a process

To stop a specific process ID for a process name, issue the KILL command with the process ID either with OSSGate for the Core.

```
Syntax:  kill <process name> only #<process ID>

>kill SOQDWRK only #451F #208A

Process 451F 1068 killed dead.
```

To stop all the process IDs for a process name, issue the KILL command with the ALL option either with OSSGate or the Core.

```
Syntax:  kill <process name> all
>kill SOQDWRK all
Process 451 1068 killed dead.
```

Query	Process Name	Example Syntax to Stop Command

QCUST	QCUST	`kill QCUST only # <process ID>`
		`kill QCUST all`

QDNSU	SOQDNSU	`kill SOQDNSU only # <process ID>`
		`kill SOQDNSU all`

QDNWRK	SOQDWRK	`kill SOQDWRK only # <process ID>`
		`kill SOQDWRK all`

QHA	SOQHA	`kill SOQHA only # <process ID>`
		`kill SOQHA all`

QHASU	SOQHASU	`kill SOQHASU only # <process ID>`
		`kill SOQHASU all`

QHU	SOQHU	`kill SOQHU only # <process ID>`
		`kill SOQHU all`

QLENWRK	SOQLWRK	`kill SOQLWRK only # <process ID>`
		`kill SOQLWRK all`

QLOAD	SOQLOAD	`kill SOQLOAD only # <process ID>`
		`kill SOQLOAD all`

QMADN	MDNQUERY	`kill MDNQUERY only # <process ID>`
		`kill MDNQUERY all`

Query	Process Name	Example Syntax to Stop Command

QNCOS	SOQNCOS	`kill SOQNCOS only # <process ID>`
		`kill SOQNCOS all`

QPDN	QPDNCI	`kill QPDNCI only # <process ID>`
		`kill QPDNCI all`

Limitations and restrictions

- Any commands not listed are considered not supported via OSSGate

  Note: Use of unsupported commands on Carrier VoIP Lines via the DMS CallServer MAP (SERVORD) interface is not supported and may cause data mismatches between CS 2000 Management Tools server and the Call Server. This is only supported for MG 9000 lines with certain restrictions.

- Any options not listed are considered not supported via OSSGate.
- Setting the DMS CallServer table OFCVAR parameter XLAPLAN_RATEAREA_SERVORD_ENABLED to MANDATORY_PROMPTS is not supported for provisioning via the OSSGate Line Provisioning interface.
- Any commands and services that have not been officially tested by Nortel product verification, are considered unsupported and are not officially sanctioned for use.
- The PDO option can not be added on a DN on treatment. The PDO option can not be added or deleted via the table editor. It can only be added or deleted using SERVORD.
- The CHG command does not support changes of LCC (Line Class Code) on lines assigned to an SAA line card that involve changes from business sets to non-business set LCCs or vice-versa.
- There is a limitation on the number of characters that can be entered in one command. This number is approximately 400 characters. If this limit is exceeded, the command will not work. Some commands that allow multiple member adds or deletes can exceed the 400 character limit.
- There is an engineering limit of 16 MADN members in the same group per LGC or gateway. See MADN Engineering Rules (SEB 92-11-001), table OFCENG Office parm MAX_MADN_MEMBERS_PER_LSG.
- When the PDO option is applied to a VDN, the VDN can not be removed using the OUTDN command. The PDO option must first be removed

using the SERVORD DEO command. This is consistent with existing PDO behavior on DNs associated with POTS, IBN or KSET lines.

- The command name in an error message may not match the command entered when performing certain SERVORD+ commands from OSSGate. The failure message will reflect the actual XA-Core SERVORD command that was executed. Some Examples are: CKTP will be displayed as CKLN and QTP will be displayed as QLEN. Please note that this discrepancy only occurs in error messages.
- Due to performance reasons, it is recommended that system-wide queries be done during off-peak times when no other provisioning is in progress.

HUNT group restrictions

The following HUNT group types are supported when using the EST SERVORD+ command:

- MLH, DLH, MPH, DNH, CPU, PRH, CMG, SIMRING

  The following HUNT group types are not supported when using the EST SERVORD+ command:

- UA, BNN, (NOTE: BNN members may be added via the ABNN command and deleted via the DBNN command).

Meridian Business Group (MBG) provisioning limitations

Office parameter MAX_MBG_LINES (OFCOPT) controls the number of lines allowed to be added to a MBG. When the maximum number of lines is reached, SERVORD+ commands will be rejected by the XA-Core. However, the XA-Core SERVORD will not remove HASU LNINV entries which were created as part of the failed processing. These entries will need to be manually removed prior to deleting the associated gateway via the CS 2000 Management Tool, however they pose no problems for further line provisioning actions via OSSGate.

CICM line provisioning limitations

When executing commands (NEW, ADO, DEO, etc.) to add or delete multiple options from CICM (Centrex IP Client Manager) lines, some options may be rejected by the CM while the overall command itself is accepted and processed successfully. For example, when adding or modifying certain line options (e.g., FXR, CAG, DASK, etc.), incompatibilities with existing options can result in a core response message of the form "<feat> COULD NOT BE ADDED" or "...FAILED TO ASSIGN ONE OR MORE FEATURES..." or "<feat> did not pass updating" or "ATTEMPT TO ASSIGN MULTIPLE APPEARANCE". In this case, the command is not rejected in it’s entirety on the CM and is therefore not rolled back on the CICM Element Manager. Manual action must be taken at the CICM Element Manager to ensure that the option data accepted by the CM is mirrored on the CICM line.

- Hunt Group commands e.g., ABNN, ADD, CHDN, DEL, EST are not supported
- Centrex IP LCC’s cannot be changed (commands CHG, CKLN, CLN).
- COPYSET command is not supported.
- PF and FTRGRP features are not supported.

SIP line provisioning limitations

CHF SERVORD command requires the use of gateway/termination names (or the equivalent LEN) in order flow through to the Session Server Lines. Use of a DN will result in the command being processed by only the Core Manager and Session Server changes would be rejected.

Provisioning using XML

The greatest use for OSSGate is automation of provisioning and maintenance commands through an OSS connection. Some commands are sent in XML form and then parsed by OSSGate into the parameters required by the interfacing provisioning and maintenance applications. Note: Line Provisioning does not support an XML interface.

Once a connection is established by the OSS, the mode of OSSGate must be set to XML. This is accomplished by sending a CTRL-B and then sending the text string “mode XML”. Once in XML mode, the OSS can begin sending the XML data to OSSGate. OSSGate will forward incoming XML input to a parser, which parses the contents of the XML file and, after performing some validation of the parameters, passes the information to the appropriate provisioning or maintenance application. Currently there are two generic formats for a valid XML input file. Both of these are described below.

Supported syntax for XML provisioning

Rules for valid XML commands

- For every open tag there must be a close tag.
- Every attribute value must be quoted, either with single or double quotes, but they must be consistent.
- No attribute name may appear more than once in a single element.
- Elements must be nested (closed) properly.
- Only one root element is allowed.
- XML is case-sensitive.
- Support of XML Schemas (rather than DTDs) and XML Name Space is added.

Supported XML file format

XML command messages

The following changes have been made to the existing Node, Carrier, and V5.2 Carrier Provisioning XML command messages for all existing commands:

- The following XML tag is no longer supported
  - — MethodName
- The following XML tag has been added since (I)SN05

— the operation name - e.g., when the operation is delete Carrier, a delete Carrier tag exists (where the MethodName tag previously existed).

- standalone="yes" is removed from the first line of the command message.
- The xml version is added to the contents of the “specific method” tag.

Example format of a pre-(I)SN05 command message - generic operation/method:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<CommandList >
<Comand>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<MethodName usn="7">someMethod</MethodName>
<Parameters>
<fieldName>somevalue</fieldName>
<fieldName>somevalue</fieldName>
</Parameters>
</Methods>
</Command>
</CommandList>
```

Example format of a an (I)SN05 and later command message - generic operation/method

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList >
<Command>
<Interface>EndptGrpProvIfInterface>EndptGrpProvIf</Inte
rface>
<Methods>
<someMethod usn="7" version="1.0">
<Parameters>
<fieldName>somevalue</fieldname>
<fieldName>somevalue</fieldname>
</Parameters>
</someMethod>
</Methods>
</Command>
</CommandList
```

Example format of a pre-(I)SN05 command message, where deleteCarrier is the example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CommandList >
<Command>
<Interface>EndptGrpProvIf</Interface>
<Methods>
```

```
<MethodName usn=”7”>deleteCarrier</MethodName>
<Parameters>
<mgName>PVG8</mgName>
<carrierName>DS3_20.1</carrierName
</Parameters>
</Methods>
</Command>
</CommandList>
```

Example format of an (I)SN05 and later command message - in this example the deleteCarrier method is used:

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList >
<Command>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<deleteCarrier usn="7" version="1.0">
<Parameters>
<mgName>PVG8</mgName>
<carrierName>DS3_20.1</carrierName>
</Parameters>
</deleteCarrier>
</Methods>
</Command>
</CommandList>
```

Note: For other commands, substitute the command name for deleteCarrier tag. For example, for the associate MG command, substitute assocMG for deleteCarrier. Make sure to do so for both the open and close tags.

XML response messages

The following changes have been made to the existing Nodes, Carrier, and V5.2 Carrier Provisioning XML response messages, for all existing commands:

- The following XML tags have been removed:
  - — MethodName
  - — usn
- The following XML tags have been added:
  - — the operation name - e.g., when the operation is deleteCarrier, a deleteCarrier tag exists (where the MethodName tag previously existed). This tag also contains the usn.
  - — ReturnData

Also, note the following three additional changes to the response message:

- version and usn number are added to the contents of the “operation name” tag. The version refers to the version of the method or operation - e.g., the deleteCarrier operation may have version 1.0 and 2.0, where version 1.0 takes parms X and Y and version 2.0 takes parms X, Y, and Z
- The RC and MsgText XML tags are moved to a new location in the response message. In (I)SN04 they came before any operation-specific data returned in the response message. In (I)SN05 and later, they come after any operation-specific data returned in the response message. The deleteCarrier operation does not reveal the relocation of these tags/fields - there is no other data returned in the response message. However, in other operations, this “re-ordering” is evident.
- standalone="yes" is removed from the first line of the response message.

Example of general file format for a valid XML response file - generic (I)SN04 operation:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CommandList>
<Response>
<Interface>somevalue</Interface>
<Methods>
<MethodName>somevalue</MethodName>
<usn>1</usn>
<RC>somevalue</RC>
<MsgTxt>somevalue</MsgTxt>
<ReturnData>
<Row>
<fieldName>somevalue</fieldName>
</Row>
</ReturnData>
</Methods>
</Response>
</CommandList>
```

Example of general file format for a valid XML response file - generic (I)SN05 and later operation:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CommandList>
<Response>
<Interface>somevalue</Interface>
<Methods>
<someMethod usn="7" version="1.0">
<ReturnData>
<fieldname>somevalue</fieldname>
<RC>somevalue</RC
<MsgTxt>somevalue</MsgTxt>
</ReturnData>
</someMethod>
```

```
</Methods>
</Response>
</CommandList>
```

Example of format of an (I)SN04 deleteCarrier response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CommandList>
<Response>
<Interface>EndptGrpProvif</Interface>
<Methods>
<MethodName>deleteCarrier</MethodName>
<usn>7</usn>
<RC>0</RC>
<MsgTxt>Delete Carrier operation was successful.</MsgTxt>
</Methods>
</Response>
</CommandList>
```

Example of format of an (I)SN05 and later deleteCarrier response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CommandList>
<Response>
<Interface>EndptGrpProvif</Interface>
<Methods>
<deleteCarrier usn="7" version="1.0>
<ReturnData>
<RC>0</RC>
<MsgTxt>Delete Carrier operation was successful.</MsgTxt>
</ReturnData>
</deleteCarrier>
</Methods>
</Response>
</CommandList>
```

Nodes provisioning with OSSGate

H.323 provisioning changes

Prior to the (I)SN07 release, for H.323 GWs the End Point Groups (EPGs) were automatically created and added to the system. The size of the EPGs was fixed, dependent upon configuration (North American systems, 24 is used, International systems, 32). The number of EPGs provisioned was automatically calculated by the system so that the appropriate number of groups were added as dictated by the mandatory “Reserved terminations” field. Before (I)SN07 the location of the TIDs allocated was under system control.

In (I)SN07, this behavior was altered. EPGs are no longer automatically added. A separate step of adding the EPGs is required using the “Add carrier” operation. Functionality update in changeMG for capacity, IPaddress, port exist. Functionality updates to support H.323 gateways in addCarrier, queryEndpointGroup, listAllCarriers, disassociateMG, deleteCarrier has been added.

XML commands

The Cs2kCfgMgr is accessed by way of the cs2kCfgMgrIf interface. The following operations are available through that interface.

Step	Action
1	Add a GWC to the call server specified by XML element addGWCtoCS (not recommended for use by OSS)
2	Delete a GWC from the call server specified by XML element deleteGWCfrmCS (not recommended for use by OSS)
3	Query the list of GWCs on the call server or query the attributes of a specific GWC specified by XML element queryGWC.
4	Associate an MG with a GWC specified by XML element assocMG.
5	Disassociate an MG specified by XML element disAssocMG
6	Change MG attributes specified by XML element changeMG.
7	Query the list of MGs associated with a GWC or query the attributes of a specific MG specified by XML element queryMG.
8	Query the list of Sites populated in Call Server table Site specified by XML element querySiteInfo.

—End—

Add GWC

The following are the XML elements that are applicable to the add GWC operation. Unless indicated the element is required in the XML document to be valid.

addGWCtoCS

specifies the provisioning method. It has two attributes usn and version. The usn is a sequence number used by OSS to associate requests with responses. The usn value specified in the request is returned in the response. The version attribute specifies the version of the method being used. Currently only 1.0.

Parameters

specifies that the following elements are parameters for the method.

csUIName

delimits call server name value. This parameter element is optional or not required. type string.

gwcUIName

delimits GWC name value. Range 0 - 255. type string.

profileName

delimits GWC profile name value. type string.

gwcActvIp

delimits the active IP address value of the GWC. type decimal.

gwcSnmpPort

delimits the SNMP protocol port value to be used. This parameter element is optional, or not required. type integer.

msgRouterIp

delimits the CM message router IP address value. This parameter element is optional or not required starting (I)SN08. type decimal.

msRouterIpPort

delimits the CM message router IP port value. This parameter element is optional or not required starting (I)SN08. type integer.

externalIP

delimits the external IP address value for the GWC. This parameter element is optional or not required. type decimal.

externalPort

delimits the external IP port value for the GWC. This parameter element is optional or not required. type integer.

bearerNetworkName

delimits the bearer network name value to which the GWC is being added. This parameter element is optional or not required. type string.

bearerFabricType

delimits the bearer network fabric type value which the GWC must support. This parameter element is optional or not required. type string.

codecProfileName

delimits the code profile name value which is to be applied to the GWC. This parameter element is optional or not required. type string.

termType

delimits terminal type value to be supported by the GWC. This element may be repeated up to 8 times along with a corresponding exec, the following element. type string.

execLineup

delimits executive routine value that is to be associated with the corresponding terminal type. This element may be repeated up to 8 times along with a corresponding terminal type. type string.

gwDefaultDomainName

allows the user to input the default domain name if needed. type string.

Add GWC Response

The following XML elements are returned in response to an ADD GWC operation:

Response

delimits the response

Interface

delimits the interface specified in the request cs2kCfgMgrIf.

Methods

delimits the cs2kCfgMgrIf methods in the request.

addGWCtoCS

method requested. Has attributes usn and version. The user sequence number provided on the request. The version of the method accessed.

RC

return code. Delimits a value indicating the results of the request.

MsgTxt

delimits text describing the results of the request.

Delete GWC

The following are the XML elements that are applicable to the delete GWC operation. Unless indicated the element is required in the XML document to be valid.

deleteGWCfrmCS

specifies the provisioning method. It has two attributes usn and version. The usn is used by OSS to associate requests with responses. The usn value specified in the request is returned in the response. The version attribute specifies the version of the method being used. Currently only 1.0.

Parameters

specifies that the following elements are parameters for the method

csUIName

delimits call server name value. This parameter element is optional or not required. type string.

gwcUIName

delimits GWC name value. type string.

Delete GWC response

The following XML elements are returned in response to a Delete GWC operation:

Response

delimits the response.

Interface

delimits the interface specified in the request cs2kCfgMgrIf.

Methods

delimits the cs2kCfgMgrIf methods in the request.

deleteGWCfrmCS

method requested. Has attributes usn and version. The user sequence number provided on the request. The version of the method accessed.

RC

return code. Delimits a value indicating the results of the request.

MsgTxt

delimits text describing the results of the request.

Query GWC

The following are the XML elements that are applicable to the Query GWC operation.

queryGWC

specifies the provisioning method. It has two attributes usn and version. The usn is used by OSS to associate requests with responses. The usn value specified in the request is returned in the response. The version attribute specifies the version of the method being used. Currently only 1.0

Parameters

specifies that the following elements are parameters for the method

Note: A choice between each of the following parameters is required. One or the other may be specified.

csUIName

delimits call server name value. When this parameter element is specified, a list of all the GWC’s in the call server is returned. type string.

gwcUIName

delimits GWC name value. When this parameter element is specified the attributes of the GWC specified are returned. type string.

Query a list of all GWCs on a Call Server response

The following XML elements are returned in response to a Query list of all GWCs on a call server operation:

Response

delimits the response.

Interface
delimits the interface specified in the request cs2kCfgMgrIf.

Methods
delimits the cs2kCfgMgrIf methods in the request.

queryGWC
method requested. Has attributes usn and version. The user sequence number provided on the request. The version of the method accessed.

ReturnData
delimits the row of the query.

Row
delimits the query data

gwcUIList
delimits the GWC name value.

gwcIpList
delimits the GWC IP value.

RC
return code. Delimits a value indicating the results of the request.

MsgTxt
delimits text describing the results of the request.

Query attributes of a GWC response

The following XML elements are returned in response to a Query attributes of a GWC operation:

Response
delimits the response.

Interface
delimits the interface specified in the request cs2kCfgMgrIf.

Methods
delimits the cs2kCfgMgrIf methods in the request.

queryGWC

method requested. Has attributes usn and version. The user sequence number provided on the request. The version of the method accessed.

ReturnData

delimits the row of the query.

Row

delimits the query data

gwcUIList

delimits the GWC name value.

gwcIpList

delimits the GWC IP value.

callServerId

delimits the call server name value.

nodeName

delimits the GWC’s node name value

typeList

delimits a GWC capability type value.

typeList values are interpreted as follows

1. Lines
2. Trunks
3. Audio
4. APG
5. DPT
6. DQoS
7. Small MGs
8. Large MGs
9. Audio MGs

10. APG MG’s
11. SIPT
12. VRDN
13. RA
14. BCT
15. IPSec
16. Kerberos
17. V52
18. Conferences
19. Announcements
20. RMGC MG’s
21. H.323

xacNodeNumber
delimits the call server node number value for the GWC.

activIpAddress
delimits the active GWC IP address value.

snmp port
delimits the value of the port the GWC uses for receiving snmp messages.

mktTones
delimits the tone value for the market where the GWC is deployed.

termTypes
delimits the call server terminal type value.

pmExecs
delimits the call server peripheral module executive value. (termTypes and pmExecs are paired corresponding values).

capacity
delimits the value of the terminal capacity of the GWC.

gwDefaultDomainName

delimits the GWC default domain name

RC

return code. Delimits a value indicating the results of the request.

MsgTxt

delimits text describing the results of the request.

Associate MG

The following are the XML elements that are applicable to the Associate MG operation. Unless indicated the element is required in the XML document to be valid.

Prior to (I)SN07, associate MG for a H.323 gateways automatically provisioned the endpoint groups. From (I)SN07, EPGs are provisioned separately using the "Add Carrier" command.

assocMG

specifies the provisioning method. It has two attributes usn and version. The usn is used by OSS to associate requests with responses. The usn value specified in the request is returned in the response. The version attribute specifies the version of the method being used. Currently only 1.0.

Parameters

specifies that the following elements are parameters for the method.

mgUIName

delimits MG name value. type string.

mgProfileName

delimits the MG profile name value to be applied to the MG. type string.

mgIpAddr

delimits the MG IP address value. type decimal.

mgProtocolType

delimits the MG protocol type. type integer. Valid values are:

1. NCS

2. Aspen

3. DSMCC

4. MEGACO

5. MGCP

6. H.323

mgProtocolVersion
delimits the version value of the MG protocol to be used. type decimal.

mgProtocolPort
delimits the port value to be used by the protocol. type integer.

mgSiteName
delimits the site name value to used by a line type MG for its logical group. This parameter element is optional or not required. type string.

gwcUIName
delimits GWC name value. This parameter element is optional or not required. When not specified the call server selects the most appropriate GWC to associate with the MG. type string.

reservedTerminations
delimits the number of terminations value to be reserved for the MG. This parameter element is optional or not required and cannot exceed the engineered or maximum number of terminations supported by the type of MG being associated as defined by the profile. type integer.

Note: A choice between each of the following 3 parameters is required, one may be specified.

pepServerName
delimits policy enforcement point (PEP) server name value. Only valid for MG that support dynamic quality of service or DQoS. type string.

itransMiddleboxName
delimits the adjacent Middlebox name (either a network address translation server name or a VCAC limited bandwidth link name). type string.

rootMiddleboxName1(..5)

delimits the root Middlebox names. Up to 5 separate entries can be defined per MG, tags must be sequential e.g. if 3 root Middleboxes then 1,2 & 3 must be provided. type string.

RASLess

delimits the optional RAS-less device data. type boolean.

Associate MG response

The following XML elements are returned in response to an Associate MG operation:

Response

delimits the response.

Interface

delimits the interface specified in the request cs2kCfgMgrIf.

Methods

delimits the cs2kCfgMgrIf methods in the request.

assocMG

method requested. Has attributes usn and version. The user sequence number provided on the request. The version of the method accessed.

gwcUIName

delimits the name of the GWC that the MG is associated to.

gwcIpAddr

delimits the value of GWC active unit IP address.

mgNodeName

delimits the MGs node name value

RC

return code. Delimits a value indicating the results of the request.

MsgTxt

delimits text describing the results of the request.

Disassociate MG

The following are the XML elements that are applicable to the Disassociate MG operation.

disAssocMG

specifies the provisioning method. It has two attributes usn and version. The usn is used by OSS to associate requests with responses. The usn value specified in the request is returned in the response. The version attribute specifies the version of the method being used. Currently only 1.0.

Parameters

specifies that the following elements are parameters for the method.

mgUIName

delimits the gateway hostname and gateway full FQDN name of the MG to be disassociated. type string.

Disassociate MG response

The following XML elements are returned in response to an Disassociate MG operation:

Response

delimits the response.

Interface

delimits the interface specified in the request cs2kCfgMgrIf.

Methods

delimits the cs2kCfgMgrIf methods in the request.

disAssocMG

method requested. Has attributes usn and version. The user sequence number provided on the request. The version of the method accessed.

RC

return code. Delimits a value indicating the results of the request.

MsgTxt

delimits text describing the results of the request.

Change MG

Change MG for H.323 Gateway supports ability to change capacity, the IP address, and the Port. The capacity change is a stand alone operation while the IP address and Port changes can be combined.

The following are the XML elements that are applicable to the Change MG operation.

changeMG

specifies the provisioning method. It has two attributes usn and version. The usn is used by OSS to associate requests with responses. The usn value specified in the request is returned in the response. The version attribute specifies the version of the method being used. Currently only 1.0.

Parameters

specifies that the following elements are parameters for the method.

mgUIName

delimits the gateway hostname and gateway full FQDN name of the MG to be disassociated. type string.

Note 1: Only one parameter may be changed per operation. A choice between each set of the following parameters is required. One or the other may be specified.

- **reservedTerminations** - delimits the MG profile name value to be applied to the MG. type integer.
- **mgIpAddr** - delimits the new MG IP address value of an H.323 MG. Not applicable to any other type of MG. type decimal.
- **mgProtocolPort** - delimits the new protocol port value of an H.323 MG. Not applicable to any other type of MG. type integer.

Note 2: The user has the ability to include the reserved termination choice or the IP address and/or Port choice, but, cannot list both choices within a single transaction. Since the IP address and Port tags have been made optional, the user can include either or both within a single transaction. Note that failure to include any of the choice tags will result in a failed operation.

- **mgProfileName** - delimits the new profile name to which the existing gateway profile wants to change. type string. Change is permitted only if the "from" GWs profile is set as compatible in the "to" profiles list. In (I)SN09 the following Nortel Profiles are compatible:
  - — MGCP_IAD
    - – MGCP_IAD_40
  - — NUERA_BTX
    - – NUERA_BTX4K
  - — NUERA_GX
    - – NUERA_GX_ASPEN

  - NUERA_GX_MEGACO

- PVG_7K
  - PVG_7K_ASPEN
  - PVG_7K_MEGACO

- PVG15K
  - PVG15K_ASPEN
  - PVG15K_MEGACO

- PVG15K_1000
  - PVG15K_1000_ASPEN
  - PVG15K_1000_MEGACO

- PVG15K_PARTIAL
  - PVG15K_PARTIAL_ASPEN
  - PVG15K_PARTIAL_MEGACO

- PVG_VSP3
  - PVG_VSP3_ASPEN
  - PVG_VSP3_MEGACO

- PVG_APG_VSP3
  - PVG_APG_VSP3_ASPEN
  - PVG_APG_VSP3_MEGACO

- PVG_APG
  - PVG_APG_ASPEN
  - PVG_APG_MEGACO

RASLess

delimits the optional RAS-less device data. type boolean.

Change MG response

The following XML elements are returned in response to a Change MG operation:

Response

delimits the response.

Interface

delimits the interface specified in the request cs2kCfgMgrIf.

Methods

delimits the cs2kCfgMgrIf methods in the request.

changeMG

method requested. Has attributes usn and version. The user sequence number provided on the request. The version of the method accessed.

RC

return code. Delimits a value indicating the results of the request.

MsgTxt

delimits text describing the results of the request.

Query MG

The following are the XML elements that are applicable to the Query MG operation.

queryMG

specifies the provisioning method. It has two attributes usn and version. The usn is used by OSS to associate requests with responses. The usn value specified in the request is returned in the response. The version attribute specifies the version of the method being used. Currently only 1.0.

Parameters

specifies that the following elements are parameters for the method.

Note: A choice between each of the following parameters is required, one or the other may be specified.

gwcUIName

delimits the GWC name value. When this parameter element is specified a list of all the MGs associated with the GWC is returned. type string.

mgUIName

delimits MG name value. When this parameter element is specified the attributes of the MG specified are returned. type string.

Query a list of MGs on a GWC response

The following XML elements are returned in response to a Query list of all MGs on a GWC operation:

Response
delimits the response.

Interface
delimits the interface specified in the request cs2kCfgMgrIf.

Methods
delimits the cs2kCfgMgrIf methods in the request.

queryMG
method requested. Has attributes usn and version. The user sequence number provided on the request. The version of the method accessed.

ReturnData
delimits the row of the query

Row
delimits the query data

mgUIList
delimits the MG name value

mgIpList
delimits the MG IP value.

RC
return code. Delimits a value indicating the results of the request.

MsgTxt
delimits text describing the results of the request.

Query attributes of an MG response

The following XML elements are returned in response to a Query attributes of an MG operation:

Response
delimits the response.

Interface
delimits the interface specified in the request cs2kCfgMgrIf.

Methods
delimits the cs2kCfgMgrIf methods in the request.

queryMG

method requested. Has attributes usn and version. The user sequence number provided on the request. The version of the method accessed.

ReturnData

delimits the row of the query

Row

delimits the query data

mgUIList

delimits the MG name value

mgIpList

delimits the MG IP value.

callServerId

delimits the call server name value.

svcTypeList

delimits an MG server type value

svcTypeList values are interpreted as follows:

1. Line
2. Trunk
3. Audio
4. APG
5. DQoS
6. ITRANS
7. H.323
8. ITRANS ROAM

nodeName

delimits the MGs node name value

pepServerName

delimits a policy enforcement point (PEP) server name when applicable (only when the MG supports this attribute e.g. DQOS service type).

itransMiddleboxName

delimits the adjacent Middlebox name (either a network address translation server name or a VCAC limited bandwidth link name) when applicable (only when the MG supports this attribute e.g. ITRANS service type).

engrEndPoints

delimits the number of reserved endpoints value.

protType

delimits the protocol type value. Values mean:

1. NCS
2. Aspen
3. DSMCC
4. MEGACO
5. MGCP
6. H.323

protversion

delimits the protocol version value

protPort

delimits the protocol port value

profileName

delimits the profile name

rootMiddleboxName1(..5)

delimits the root Middlebox names up to 5 separate entries can be returned per MG when applicable (only when the MG supports this attribute e.g. ITRANS_ROAM service type).

RASLess

delimits the optional RAS-less device data

RC

return code. Delimits a value indicating the results of the request. See table for list of valid values.

MsgTxt

delimits text describing the results of the request.

> ***Note:*** The pepServerName, itransMiddleboxName and rootMiddleboxName1..5 tags are only returned to the client when the MG service type is compatible (see specific tags above). Also an empty tag can be returned to the client when there is no entry (this is displayed as <none> on the CS 2000 Config Manager GUI.). The format used for the empty tag is as follows:
> <exampleTag></exampleTag>

Get Site

The following are the XML elements that are applicable to the Get Site operation.

getSiteInfoMG

specifies the provisioning method. It has two attributes usn and version. The usn is used by OSS to associate requests with responses. The usn value specified in the request is returned in the response. The version attribute specifies the version of the method being used. Currently only 1.0.

Parameters

delimits the one parameter

csUIName

delimits the call server name value. type string.

Get Site response

The following XML elements are returned in response to a Get list of Site names operation:

Response

delimits the response.

Interface

delimits the interface specified in the request cs2kCfgMgrIf.

Methods

delimits the cs2kCfgMgrIf methods in the request.

getSiteInfo

method requested. Has attributes usn and version. The user sequence number provided on the request. The version of the method accessed.

ReturnData

delimits the row of the query

Row

delimits the query data

siteNameList

delimits a site name value read from Site table

RC

return code. Delimits a value indicating the results of the request.

MsgTxt

delimits text describing the results of the request.

The following are examples of XML commands and the corresponding response messages for each operation supported for Nodes provisioning. The XML code in these examples is formatted for ease of understanding.

> ***Note:*** Adding and deleting gateway controllers should be done from the Call Server 2000 GUI rather than via OSSGate. The syntax is provided for reference.

An example of an add GWC request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<addGWCtoCS usn="1" version="1.0">
<Parameters>
<csUIName>COMPACT6</csUIName>
<gwcUIName>GWC-12</gwcUIName>
<profileName>LARGE_LINENA</profileName>
<gwcActvIp>26.3.4.8</gwcActvIp>
<gwcSnmpPort>161</gwcSnmpPort>
<msgRouterIp>47.142.128.144</msgRouterIp>
<msgRouterIpPort>4684</msgRouterIpPort>
<bearerNetworkName>NET_IP</bearerNetworkName>
<bearerFabricType>IP</bearerFabricType>
<codecProfileName>Test</codecProfileName>
<termType>POTS</termType>
<termType>KEYSET</termType>
<execLineup>POTSEX</execLineup>
<execLineup>KSETEX</execLineup>
<gwDefaultDomainName>nortel.com.cn</
gwDefaultDomainName>
</Parameters>
</addGWCtoCS>
</Methods>
</Command>
</CommandList>
```

An example of an addGWC response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<addGWCtoCS usn="1"  version="1.0">
<ReturnData>
<RC>0</RC>
<MsgTxt>Add GWC operation was successful</MsgTxt>
</ReturnData>
</addGWCtoCS>
</Methods>
</Response>
</CommandList>
```

An example of assocMG request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<assocMG usn="1" version="1.0">
<Parameters>
<mgUIName>mg1.nortel.net</mgUIName>
<mgProfileName>ASKEY_LINE_GW_4</mgProfileName>
<mgIpAddr>47.49.23.43</mgIpAddr>
<mgProtocolType>5</mgProtocolType>
<mgProtocolVersion>2.0</mgProtocolVersion>
<mgProtocolPort>2427</mgProtocolPort>
<gwcUIName>GWC-0</gwcUIName>
<RASless>True</RASless>
</Parameters>
</assocMG>
</Methods>
</Command>
</CommandList>
```

Note: The parameters need to be specified in correct sequential order as listed in the request command

An example of assocMG response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<assocMG usn="1"  version="1.0">
<ReturnData>
<Row>
```

```
<gwcUIName>GWC-0</gwcUIName>
<gwcIpAddr>172.17.40.24</gwcIpAddr>
<mgNodeName>LG 00 0</mgNodeName>
</Row>
<RC>0</RC>
<MsgTxt>The MG was successfully associated with a
GWC</MsgTxt>
</ReturnData>
</assocMG>
</Methods>
</Response>
</CommandList>
```

H.323 gateway association

The process for associating an H.323 MG via XML is very similar to adding any other media gateway. The only difference are the additional MG profiles. The reserved terminations tag is a mandatory tag for H.323 GWs. The reserved endpoints must be specified as the last parameter tag Example: `<reservedTerminations>32</reservedTerminations>`

Associate H.323 MG request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<assocMG usn="1" version="1.0">
<Parameters>
<mgUIName>H323_M1</mgUIName>
<mgProfileName>SUCCESSION_1000</mgProfileName>
<mgIpAddr>10.10.10.43</mgIpAddr>
<mgProtocolType>6</mgProtocolType>
<mgProtocolVersion>4.0</mgProtocolVersion>
<mgProtocolPort>6000</mgProtocolPort>
<gwcUIName>GWC-0</gwcUIName>
<reservedTerminations>24</reservedTerminations>
<itransMiddleboxName>NAT1</itransMiddleboxName>
</Parameters>
</assocMG>
</Methods>
</Command>
</CommandList>
```

An example of disAssocMG request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<disAssocMG usn="1"  version="1.0">
```

```
<Parameters>
<mgUIName>mg1.nortel.net</mgUIName>
</Parameters>
</disAssocMG>
</Methods>
</Commands>
</CommandList>
```

An example of disAssocMG response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<disAssocMG usn="1"  version="1.0">
<ReturnData>
<RC>0</RC>
<MsgTxt>The MG was successfully disassociated from a
GWC</MsgTxt>
</ReturnData>
</disAssocMG>
</Methods>
</Response>
</CommandList>
```

Note: Before a gateway can be deleted, it is important to verify that it is not in use to avoid taking down active calls. All associated services (lines, carriers, trunks etc. depending on the type of gateway) must be deleted to ensure that no calls can originate during the gateway deletion process. To delete a H.323 gateway “BSY INB” the corresponding trunk group on the CS 2000 and delete the associated H.323 carrier. Failure to do this will result in the deletion being denied by the system.

An example of changeMG request for capacity

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<changeMG usn="1"  version="1.0">
<Parameters>
<mgUIName>mg1.nortel.net</mgUIName>
<reservedTerminations>3</reservedTerminations>
</Parameters>
</changeMG>
</Methods>
</Commands>
</CommandList>
```

An example of changeMG request for IP address and/or Port

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<changeMG usn="1"  version="1.0">
<Parameters>
<mgUIName>mg1.nortel.net</mgUIName>
<mgIpAddr>22.44.66.88</mgIpAddr>
<mgProtocolPort>88</mgProtocolPort>
</Parameters>
</changeMG>
</Methods>
</Commands>
</CommandList>
```

Note: The IP/Port change operation is only supported for H.323 GWs and they must have a NAT associated with them.

An example of changeMG request for Profile change

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<changeMG usn="1"  version="1.0">
<Parameters>
<mgUIName>MG1</mgUIName>
<mgProfileName>ARRIS_TOUCHTONE_NN03_4</mgProfileName>
</Parameters>
</changeMG>
</Methods>
</Commands>
</CommandList>
```

An example of changeMG response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<changeMG usn="1"  version="1.0">
<RC>0</RC>
<mgUIName>mg1.nortel.net</mgUIName>
<MsgTxt>mg1.nortel.net Change MG Successful
All Change MG requests have been processed.
</MsgTxt>
</changeMG>
</Methods>
```

```
</Response>
</CommandList>
```

An example of changeMG request that causes a failure response - a change attempted when reserved terminations exceed the maximum

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<changeMG usn="1"  version="1.0">
<Parameters>
<mgUIName>mg1.nortel.net</mgUIName>
<reservedTerminations>45</reservedTerminations>
</Parameters>
</changeMG>
</Methods>
</Command>
</CommandList>
```

An example of changeMG response - failure scenario

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<changeMG usn="1"  version="1.0">
<RC>50</RC>
<mgUIName>mg1.nortel.net</mgUIName>
<MsgTxt>mg1.nortel.net Change MG Failed - Reserved
Terminations Exceed
Maximum.  All Change MG requests have been processed.
</MsgTxt>
</changeMG>
</Methods>
</Response>
</CommandList>
```

An example of getSiteInfo request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<getSiteInfo usn="1"  version="1.0">
<Parameters>
<csUIName>RTPS</csUIName>
</Parameters>
</getSiteInfo>
</Methods>
```

```
</Command>
</CommandList>
```

An example of getSiteInfo response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<getSiteInfo usn="1"  version="1.0">
<ReturnData>
<Row>
<siteNameList>HOST</siteNameList>
<siteNameList>LG</siteNameList>
<siteNameList>UAIP</siteNameList>
</Row>
<RC>0</RC>
<MsgTxt>Request successful</MsgTxt>
</ReturnData>
</getSiteInfo>
</Methods>
</Response>
</CommandList>
```

An example of query all GWC on a call server request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryGWC usn="1"  version="1.0">
<Parameters>
<csUIName>RTPS</csUIName>
</Parameters>
</queryGWC>
</Methods>
</Command>
</CommandList>
```

Note: The query param in Query Gateway Controller on a Call Server can be either <csUIName>, which will list all GWCs in that CM CLLI space, or <gwcUIName>, which will list the details about specific GWCs.

An example of query all GWC on a call server response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryGWC usn="1"  version="1.0">
```

```
<ReturnData>
<Row>
<gwcUIList>GWC-0</gwcUIList>
<gwcIpList>172.17.40.24</gwcIPList>
</Row>
<Row>
<gwcUIList>GWC-1</gwcUIList>
<gwcIpList>172.17.40.28</gwcIPList>
</Row>
<Row>
<gwcUIList>GWC-2</gwcUIList>
<gwcIpList>172.17.40.32</gwcIPList>
</Row>
<Row>
<gwcUIList>GWC-3</gwcUIList>
<gwcIpList>172.17.40.36</gwcIPList>
</Row>
<RC>0</RC>
<MsgTxt>Query of all the GWC on a Call Server
was successful</MsgTxt>
</ReturnData>
</queryGWC>
</Methods>
</Response>
```

An example query attributes of specific GWC request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryGWC usn="1"  version="1.0">
<Parameters>
<gwcUIName>GWC-1</gwcUIName>
</Parameters>
</queryGWC>
</Methods>
</Command>
</CommandList>
```

Response to query attributes of a specific GWC

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryGWC usn="1"  version="1.0">
<ReturnData>
<Row>
<gwcUIList>GWC-0</gwcUIList>
```

```
<gwcIpList>172.17.40.24</gwcIPList>
<callserverId>RTPS</callServerId>
<nodeName>GWC1</nodeName>
<typeList>2</typeList>
<typeList>8</typeList>
<xacNodeNumber>13</xacNodeNumber>
<actvIpAddress>172.17.40.28</actvIpAddress>
<snmpPort>161</snmpPort>
<mktTones>NORTHAM</mktTones>
<termtypes>ABTRK</termtypes>
<termtypes>PRAB</termtypes>
<termtypes>null</termtypes>
<termtypes>null</termtypes>
<termtypes>null</termtypes>
<termtypes>null</termtypes>
<termtypes>null</termtypes>
<termtypes>null</termtypes>
<pmExecs>DTCEX</pmExecs>
<pmExecs>UTR250</pmExecs>
<pmExecs>null</pmExecs>
<pmExecs>null</pmExecs>
<pmExecs>null</pmExecs>
<pmExecs>null</pmExecs>
<pmExecs>null</pmExecs>
<pmExecs>null</pmExecs>
<capactity>4094</capacity>
</Row>
<RC>0</RC>
<MsgTxt>Query of a Single GWC was successful</MsgTxt>
</ReturnData>
</queryGWC>
</Methods>
</Response>
</CommandList>
```

An example of query attribute of a specific MG request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryMG usn="1"  version="1.0">
<Parameters>
<mgUIName>mg1.nortel.net</mgUIName>
</Parameters>
</queryMG>
</Methods>
</Command>
</CommandList>
```

Note: The GWC name in gwcUIName tag need not be provided. However, either gwcUIName or mgUIName tag needs to have a value provided. If mgUIName is empty, all the MGs for that GWC will be returned.

An example of query attribute of a specific MG response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryMG usn="1"  version="1.0">
<ReturnData>
<Row>
<mgUIList>mg1.nortel.net</mgUIList>
<mgIpList>47.49.23.43</mgIpList>
<callServerId>RTPS</callServerId>
<gwcUIName>GWC-0</gwcUIName>
<svcTypeList>0</svcTypeList>
<svcTypeList>1</svcTypeList>
<nodeName>LG 00 0</nodeName>
<itransMiddleboxName>
<engrEndPoints>2</engrEndPoints>
<protType>mgcp</protType>
<protVersion>2.0</protVersion>
<protPort>2427</protPort>
<profileName>ASKEY_LINE_GW_4</profileName>
<maxEngrEndPoints>4</maxEngrEndPoints>
</Row>
<RC>0</RC>
<MsgTxt>Query of a Single MG was successful</MsgTxt>
</ReturnData>
</queryMG>
</Methods>
</Response>
</CommandList>
```

An example of query List of MGs on a GWC request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryMG usn="1"  version="1.0">
<Parameters>
<gwcUIName>GWC-0</gwcUIName>
</Parameters>
</queryMG>
</Methods>
</Command>
```

```
</CommandList>
```

An example of query List of MGs on a GWC response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryMG usn="1"  version="1.0">
<ReturnData>
<Row>
<mgUIList>mg1.nortel.net</mgUIList>
<mgIpList>47.49.23.43</mgIpList>
</Row>
<Row>
<mgUIList>pp1.arris.net</mgUIList>
<mgIpList>10.9.1.201</mgIpList>
</Row>
<Row>
<mgUIList>pp10.arris.net</mgUIList>
<mgIpList>10.9.1.210</mgIpList>
</Row>
<Row>
<mgUIList>pp2.arris.net</mgUIList>
<mgIpList>10.9.1.202</mgIpList>
</Row>
<Row>
<mgUIList>pp3.arris.net</mgUIList>
<mgIpList>10.9.1.203</mgIpList>
</Row>
<Row>
<mgUIList>pp4.arris.net</mgUIList>
<mgIpList>10.9.1.204</mgIpList>
</Row>
<Row>
<mgUIList>pp5.arris.net</mgUIList>
<mgIpList>10.9.1.205</mgIpList>
</Row>
<Row>
<mgUIList>pp6.arris.net</mgUIList>
<mgIpList>10.9.1.206</mgIpList>
</Row>
<Row>
<mgUIList>pp7.arris.net</mgUIList>
<mgIpList>10.9.1.207</mgIpList>
</Row>
<Row>
<mgUIList>pp8.arris.net</mgUIList>
<mgIpList>10.9.1.208</mgIpList>
</Row>
```

```
<Row>
<mgUIList>pp9.arris.net</mgUIList>
<mgIpList>10.9.1.209</mgIpList>
</Row>
<Row>
<mgUIList>rtps1124a.nortel.com</mgUIList>
<mgIpList>10.9.1.50</mgIpList>
</Row>
<RC>0</RC>
<MsgTxt>Query of all the Media Gateway’s on the Gateway
Controller was successful</MsgTxt>
</ReturnData>
</queryMG>
</Methods>
</Response>
</CommandList>
```

An example of Error Response format for any operation

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryMG usn="1"  version="1.0">
<ReturnData>
<RC>2</RC>
<MsgTxt>Operation failed</MsgTxt>
</ReturnData>
</queryMG>
</Methods>
</Response>
</CommandList>
```

Policy Enforcement Point Provisioning for Cable

Configuration management of PacketCable dynamic quality of service (DQoS) elements are Policy Enforcement Points (PEP) servers, admission control policies, DQoS network level parameters, GWC to PEP Server connections and MG-to-PEP Server associations.

The primary purpose for implementing DQoS PEPs is to provide subscriber admission control mechanisms and to minimize abusive QoS usage. Abuse occurs when a subscriber fraudulently uses QoS or when denial-of-service attacks are launched.

Policy enforcement is performed by a PEP in the CMTS. Each PEP is associated with a particular multimedia terminal adapter (MTA) or MG that belongs to a subscriber. Each MTA subscribes to a service level or QoS which is granted to each session or call. The class of the session identifies the proper admission control policy or parameters to be applied for a call.

Data is configured by a system administrator to change the desired policy that is enforced by the PEP. A separate policy is defined for each level of service that is to be supported in a PacketCable network. Each policy has a DSCP, differentiated services code point and a flow specification.

At this time, a single network wide policy makes admission control the primary reason for this capability. The use of multiple policies, such as qualities of service, is not currently supported.

OSS-based DQoS Configuration management

Provisioning of the DQoS capability in the Call Server 2000 (CS 2000) is accomplished from the CS 2000 Management tool GUI. Initial provisioning of a Cable Network supporting DQoS and its elements PEP's and policies proceeds as follows:

Step	Action
1	Identify the policy enforcement point servers by adding PEP Servers to the management system. Note: Ensure that the PEP server has been added using the CS 2000 management tool GUI. Add, delete, and change of PEP servers is performed from the CS 2000 Management Tools GUI. Follow details in *GWC Configuration*, NN10112-511
2	Specify the admission control policy to be enforced.
3	Associate all MGs (MTAs) with a PEP Server defined in step 1 above.

—End—

Once these above steps are completed the network managed from the CS 2000 management tool is DQoS capable. It is now possible to do any of the following:

Step	Action
1	Association of additional MG's with GWCs and PEP Servers by using assocMG xml command.
2	Changing an MG's PEP association.
3	Monitoring the level of the number of PEP Server to GWC connections and GWC to PEP Server connections to avoid exceeding limits resulting in failures to associate MGs with GWC's by using the QueryPEPData xml command.

4 Querying of the admission control policy by using the QueryDQoSPolicy xml command.

5 Auditing PEP Servers specified in the network by using the QueryPEPInfo xml command.

6 Auditing MG associations with PEP’s by using the queryMG xml command.

—End—

An example of PEP server use with assocMG request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<assocMG usn="1" version="1.0">
<Parameters>
<mgUIName>mg5.nortel.net</mgUIName>
<mgProfileName>ARRIS_TOUCHTONE_NN02_4</mgProfileName>
<mgIpAddr>47.143.34.39</mgIpAddr>
<mgProtocolType>1</mgProtocolType>
<mgProtocolVersion>1.0</mgProtocolVersion>
<mgProtocolPort>2724</mgProtocolPort>
<mgSiteName>LG</mgSiteName>
<gwcUIName>GWC-0</gwcUIName>
<reservedTerminations>4</reservedTerminations>
<pepServerName>pep1</pepServerName>
</Parameters>
</assocMG>
</Methods>
</Command>
</CommandList>
```

An example of PEP server use with assocMG response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<assocMG usn="1" version="1.0">
<ReturnData>
<Row>
<gwcUIName>GWC-0</gwcUIName>
<gwcIpAddr>172.17.40.24</gwcIpAddr>
<mgNodeName>LG 00 0</mgNodeName>
</Row>
<RC>0</RC>
```

```
<MsgTxt>The MG was successfully associated with a
GWC.</MsgTxt>
</ReturnData>
</assocMG>
</Methods>
</Response>
</CommandList>
```

An example of query Single MG Request with PEP server

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryMG usn="1" version="1.0">
<Parameters>
<mgUIName>gateway1.nortel.net</mgUIName>
</Parameters>
</queryMG>
</Methods>
</Command>
</CommandList>
```

An example of Query Single MG Response with PEP server

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryMG usn="1" version="1.0">
<ReturnData>
<Row>
<mgUIList>gateway1.nortel.net</mgUIList>
<mgIpList>47.124.142.86</mgIpList>
<callServerId>COMPACT5</callServerId>
<gwcUIName>GWC-6</gwcUIName>
<svcTypeList>0</svcTypeList>
<svcTypeList>4</svcTypeList>
<nodeName>LG001</nodeName>
<pepServerName>pep1</pepServerName>
<algName>NOT_SET</algName>
<engrEndPoints>4</engrEndPoints>
<protType>ncsprotocol</protType>
<protVersion>1.0</protVersion>
<protPort>2427</protPort>
<profileName>ARRIS_TOUCHTONE_NN01_4</profileName>
<maxEngrEndPoints>4</maxEngrEndPoints>
<frameNumber>not_set</frameNumber>
<unitNumber>not_set</unitNumber>
<frameType>not_set</frameType>
```

```
<floorPos>not_set</floorPos>
<rowPos>not_set</rowPos>
<framePos>not_set</framePos>
<unitPos>not_set</unitPos>
</Row>
<RC>0</RC>
<MsgTxt>Query of a Single MG was successful</MsgTxt>
</ReturnData>
</queryMG>
</Methods>
</Response>
</CommandList>
```

An example request to change assigned PEP server to a media gateway

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>DQoSOSSIf</Interface>
<Methods>
<changeMGGWCEMData usn="135" version="1.0">
<Parameters>
<mgUIName>mg5.nortel.net</mgUIName>
<PEPServerName>pep2</PEPServerName>
</Parameters>
</changeMGGWCEMData>
</Methods>
</Command>
</CommandList>
```

Note: The pep server must exist in the system before the change is attempted. In the example mg5.nortel.net assigned to pep1 is changed to pep2. Both pep1 and pep2 are available before the change.

An example of response to change assigned PEP server to a media gateway

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>DQoSOSSIf</Interface>
<Methods>
<changeMGGWCEMData usn="135" version="1.0">
<ReturnCode value="0" text="Successful result"/>
</changeMGGWCEMData>
</Methods>
</Response>
</CommandList>
```

An example to get one PEP Server Data request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
```

```
<Command>
<Interface>DQoSOSSIf</Interface>
<Methods>
<getPEPServersData usn="135" version="1.0">
<Parameters>
<PEPServerName>pep1</PEPServerName>
</Parameters>
</getPEPServersData>
</Methods>
</Command>
</CommandList>
```

An example to get one PEP Server Data response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>DQoSOSSIf</Interface>
<Methods>
<getPEPServersData usn="135" version="1.0">
<ReturnCode value="0" text="Successful result"/>
<PEPServerName>pep1</PEPServerName>
<Type>dqosmb</Type>
<IPAddress>47.143.34.39</IPAddress>
<MaxConnections>10</MaxConnections>
<ProtocolVersion>DQOS I04</ProtocolVersion>
</getPEPServersData>
</Methods>
</Response>
</CommandList>
```

An example QueryGWCPEPConn request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>DQoSOSSIf</Interface>
<Methods>
<getPepNamesbyGWCId usn="1" version="1.0">
<Parameters>
<GWC>GWC-4</GWC>
</Parameters>
</getPepNamesbyGWCId>
</Methods>
</Command>
</CommandList>
```

An example of Query GWC to PEP connections response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>DQoSOSSIf</Interface>
```

```
<Methods>
<getPepNamesbyGWCId usn="1" version="1.0">
<ReturnCode value="0" text="Successful result"/>
<GWC>GWC-6</GWC>
<PEPServer>pep1</PEPServer>
</getPepNamesbyGWCId>
</Methods>
</Response>
</CommandList>
```

An example GetDQoSPoliciesData request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>DQoSOSSIf</Interface>
<Methods>
<getDQoSPoliciesData usn="379" version="1.0">
<Parameters>
<Polcy>All</Policy>
</Parameters>
</getDQoSPoliciesData>
</Methods>
</Command>
</CommandList>
```

An example of GetDQoSPoliciesData response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>DQoSOSSIf</Interface>
<Methods>
<getDQoSPoliciesData usn="379" version="1.0">
<RC>0</RC>
<MsgTxt>Data Available</MsgTxt>
<DSCP>101110</DSCP>
<BucketDepth>200</BucketDepth>
<BucketRate>150</BucketRate>
<PeakRate>1000</PeakRate>
<MinimumUnit>16</MinimumUnit>
<MaxDataGram>1024</MaxDataGram>
<ReservedRate>4096</ReservedRate>
<SlackTerm>200</SlackTerm>
</getDQoSPoliciesData>
</Methods>
</Response>
</CommandList>
```

An example Delete GWC request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
```

```
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<deleteGWCfrmCS usn="1" version="1.0">
<Parameters>
<csUIName>COMPACT6</csUIName>
<gwcUIName>GWC-9</gwcUIName>
</Parameters>
</deleteGWCfrmCS>
</Methods>
</Command>
</CommandList>
```

An example of Delete GWC response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<deleteGWCfrmCS usn="1" version="1.0">
<ReturnData>
<RC>0</RC>
<MsgTxt>Delete GWC operation was successful</MsgText>
</ReturnData>
</deleteGWCfrmCS>
</Methods>
</Response>
</CommandList>
```

PEP provisioning limitations and restrictions

Certain provisioning rules apply when provisioning PEP servers. These include:

- AssocMG
  - — The profile applied to the MG must have a service type of DQoS when a PEP Server is specified.
  - — The GWC specified must have the DQoS capability when a PEP Server is specified.
  - — The GWC specified cannot have more than 20 PEP Server connections when this limit is exceeded the command is rejected.
  - — The PEP Server specified cannot have more than the maximum number of connections specified when it was defined. When this limit is exceeded the command is rejected.
- QueryPEPInfo
  - — The specified PEP name must be defined.

- QueryGWCPEPData
  — The specified GWC must be defined.

Event Messaging support for Cable gateways

To meet compliancy to PacketCable lab specification, any event message sent from the call server need to contain the gateway and termination information.

To facilitate the above a new table LNENDPT was added in the CS 2000 in (I)SN08. As part of adding of gateways this table is datafilled for each termination. Lens are thus pre-provisioned. The pre-provisioned lens are in HASU state when the gateways are provisioned. The state changes to WORKING when services are assigned.

Verifying for HASU status

When in CI mode in OSSGate, the QLEN commands response provides an indication of the line status.

QLEN query and response for a HASU Len

```
qlen LG 00 0 00 01

-------------------------------------
TYPE: HARDWARE ASSIGNED SOFTWARE UNASSIGNED
CARDCODE: RDTLSG GND: N PADGRP: PKLNL BNV:NL MNO: N
PM NODE NUMBER : 33
PM TERMINAL NUMBER : 2
-------------------------------------
```

AssociateMG, DisassociateMG and ChangeMG operations facilitate the management of data in the new table.

Upgrade to (I)SN09

The full support for event messaging occurs only after all elements in the network are upgraded. The CS 2000 is the last in the upgrade order. Refer to upgrade doc NN10061-461 (Upgrading the Communications Server 2000). On upgrade of CS2M to (I)SN09 node provisioning operations will continue to work with the CS 2000 in a previous supported release. On upgrade of CS 2000 a data upgrade process gets initiated, in order to populate the new table LNENDPT. This data transfer process can take some time depending on the switch data fill size. This takes place on a per GWC basis. Any failure of the upgrade process will send an alarm to the fault management systems. Refer to fault management doc. for alarm details. This upgrade alarm details include:

```
Alias Name:  EventMessagingUpgrade
Category:  Processing Error
Alarm Time:  hh:mm:ss DD-MM-YYYY <TimeZone Info>
Severity:  Major
Probable Cause:  "Upgrade Failure"
```

```
Specific Problem:  The Event Messaging Upgrade failed or has
been interrupted
```

During the upgrade period

- A minor alarm is raised. This alarm indicates the data specific to a GWC being upgraded. The alarm will clear when the specific GWC has been upgraded.
- Node provisioning and line provisioning operations will fail gracefully with the message as shown below.
- The CS2M Line audit operation from CS 2000 Management Tools GUI is disabled during the upgrade period. Running a audit can potentially cause incorrect results to be reported.

An example of upgrade-in-progress response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</interface>
<Methods>
<changeMG usn="1" version="1.0">
<RC>20</RC>
<mgUIName>mg1.nortel.net</mgUIName>
<MsgTxt>mg1.nortel.net Change MG Failed.
Data upgrade to SN09 in progress.  Please try again
later.</MsgTxt>
</changeMG>
</Methods>
</Response>
</CommandList>
```

User authorization for Node Provisioning operations

In addition to users belonging to “succssn” group to login to OSSGate, user need to be in application specific groups to perform specific operations. Each operation is associated with one or more user groups. In order to execute a command, a user must belong to at least one of the associated user groups. The user group’s associated with each OSSGate Node provisioning operation are specified in the table below.

Command	**User Group**				
	mgcadm	**mgciprov**	**mgcmtc**	**mgcsprov**	**mgcro**
disAssocMg	X	X			
assocMG	X	X			
changeMG	X	X			

Command	User Group				
	mgcadm	mgciprov	mgcmtc	mgcsprov	mgcro
querySiteInfo	X	X	X	X	X
queryQWC	X	X	X	X	X
queryMG	X	X	X	X	X
changeMGGWCEMData	X	X			
getPEPServerData	X	X	X	X	X
queryGWCPEPConn	X	X	X	X	X
getDQosPoliciesData	X	X	X	X	X
queryEndpointGroups	X	X	X	X	X

Return Codes

The table below shows the Nodes Provisioning return codes.

Return Code	Meaning
0	Successful operation
1	More data coming
2	Request Rejected due to Resource Limits Reached
3	Application was Commanded to Abort
4	Invalid input from client
5	Failed SERVINV update or could not get node name and number
6	Failed to Add to GWCEM
7	Failed to update NetworkView database
8	Failed Rollback of ADD GWC
9	Query GWC Operation Failed to get Data from GWC EM
10	Query GWC Operation Failed to Read GWC List from the Network View
11	Query GWC Operation Failed to Get Data from XA Core
12	Delete GWC Operation failed GWC Data in Network View could not be read
13	Delete GWC Operation aborted, the GWCEM failed to delete the GWC
14	Delete GWC Operation aborted, GWCEM rejected delete, GWC has associated MG's or Endpoints, invalid operations
15	Delete GWC Operation encountered an error after deleting the GWC from the GWCEM
16	Associate MG Operation Invalid input from client interface
17	Associate MG Operation failed to access DB when reading gateway table

Return Code	Meaning
18	Associate MG Operation failed to assign a GWC. GWC with sufficient capacity could not be selected
19	Associate MG Operation failed to read the GWC Data frames Network View
20	Associate MG Operation failed to assign a LGRP node name for the MG
21	Associate MG Operation failed to associate the MG with a GWC
22	Associate MG Operation failed to update the SB and the Network View with data about the new MG
23	Associate MG Operation failed roll back of a transaction in progress that encountered an error
24	Query MG Operation invalid input from client
25	Query MG Operation failed to read the Gateway Data from the GWCEM
26	Disassociate MG Operation invalid input from client
27	Disassociate MG Operation failed to Read Gateway Data
28	Disassociate MG Operation failed to read Gateway Controller Data
29	Disassociate MG Operation aborted, GWCEM failed to delete the MG
30	Disassociate MG Operation aborted, GWCEM rejected delete, MG has provisioned Endpoints, this is an invalid operation
31	Disassociate MG Operation failed to de-assign Lgrp Node Name
32	Disassociate MG Operation encountered an error after deleting the MG from the GWCEM
33	Audit of XA-Core data used by Cs2kCfgMgr failed to complete successfully
34	Query XA-Core GWC data used by CS2KCfgMgr failed to complete successfully
35	Query XA-Core GWC data used by CS2KCfgMgr failed to complete IP not found
36	Query XA-Core GWC data not found in XAC
37	Query XA-Core GWC data failed
38	Associate MG Operation MG is already provisioned
39	One of the mandatory parameters required for assigning a MG is not present
40	One of the parameters used when assigning a MG does not have the correct format
41	The query of XA-Core table site failed
42	The query of XA-Core table site returned an empty list
43	The MG name is not known
44	One of the parameters used does not have the correct format
45	One of the mandatory parameters required is not present
46	Other input errors
47	Change MG Operation failed to access NV when reading gateway table

Return Code	Meaning
48	Change MG Operation failed to access GWC-EM when reading GWC data
49	Change MG Operation failed to read the Profile Data from the Network View
50	Change MG Operation the number of reserved terminations exceeds maximum value.
51	Change MG Operation the number of reserved terminations less than 1.
52	Change MG Operation the number of reserved terminations less than
53	Change MG Operation the GWC does not have capacity
54	Change MG Operation failed to update the SB and the NetworkView with data about the new MG
55	Change MG Operation failed roll back of a transaction in progress that encountered an error
56	Failed SERVRINV update or could not get node name and number
57	Failed change to GWCEM registration
58	Failed to update NV
59	Failed RollBack of Change GWC
60	Associate MG Operation Preprovisioning Failed
61	Associate MG Operation Preprovisioning is in progress
62	Failed MG 9000 VMG line conversion
63	Upgrade MG Operation progress message indication
64	Change MG Operation Preprovisioning Filed (i.e. failed to add/delete line card(s))
65	The GWC is not deletable from GWCEM (i.e. provisioned data associated with it
66	The MG is not deletable from GWCEM (i.e. provisioned data associated with it
67	The MG cannot be associated because the number of gateways in the call server is at the limit that a redirecting GWC can support.
68	Associate MG operation failed and was rolled back
69	The MG and GWC TN’s are provisioned, LEN provisioning is incomplete retry to continue and complete LEN provisioning
70	The query of XA-Core table BEARNETS failed
71	The query of XA-Core table BEARNETS returned an empty list
72	An event was received by a state which it was able to process. This is a software error

Nodes Provisioning support for Internet Transparency

Provisioning Media Gateways in IP address spaces that are different to those in the Telecom Service Providers office is supported by the Internet Transparency capabilities of the OSSGate Nodes provisioning interface.

This provides the ability to associate Media Gateways with NAT middleboxes as part of the Media Gateway provisioning requests. It also provides the ability to query the NAT middleboxes and Media Proxies that have been provisioned for a CS 2000.

Provisioning Media Gateways that are behind a limited bandwidth link (LBL) is supported by the VCAC capabilities of the Nodes provisioning interface. This provides the ability to associate Media Gateways with LBL middleboxes as part of the Media Gateway provisioning requests.

Media Gateways which support service type ’ITRANS_ROAM’ may be associated with up to five ’Root’ middleboxes as part of the Media Gateway provisioning request. This allows terminals belonging to the Media Gateway to operate from behind NATs and limited bandwidth links.

Beginning with the (I)SN08 release and introduction of support for Network Virtual Call Admission Control (VCAC) counting using the Policy Controller (PC) platform, new network zone interfaces are introduced for use in situations where a Policy Controller is part of the deployment. These new interfaces are replacements for the current NAT and LBL provisioning interfaces, although the later remain in place and supported. Note that any topology (Network Zone) provisioning commands (i.e. add, change, delete) MUST be carried out in both the CS 2000 Management system AND the Policy Controller. Each command should be sent first to the CS 2000 Management system and, if this succeeds, to the Policy Controller (which supports the same XML schema).

The interface also provides the ability to provision the following network devices, and topological links between middleboxes.

- NAT and LBL middleboxes
- NAT, LBL, and Composite NAT plus LBL network zones
- Resource Usage profile (used by VCAC LBL middleboxes)
- Media Proxies

In order to support the functionality which allows SIP-T trunks to be configured as ’intra-domain’ SIP-T trunks, provisioning of a unique identifier for each CS 2000 the ’Call Agent Identifier’ is provided. It should be noted that the (I)SN08 version of the PC has no intra-domain support and hence this functionality should not be used where external VCAC counting is enabled for use with the PC.

The OSSGate interface provides the following:

- Querying the Call Agent Identifier
- Setting the Call Agent Identifier
- Adding a NAT or LBL Middle box
- Changing a NAT or LBL middlebox attribute

- Deleting a NAT or LBL Middle box
- Adding a RU Profile
- Changing an RU profile attribute
- Deleting an RU profile
- Association of a MG to an adjacent NAT / LBL middlebox OR a set of root Middleboxes as part of the MG to Gateway Controller association request
- Changing of the adjacent NAT / LBL middlebox OR set of rootMiddleboxes associated with a Media Gateway
- Listing of the adjacent NAT / LBL middlebox OR rootMiddleboxes assigned to a MG as part of the query of the MG data
- The ability to specify that the MG is outside Telecom Service Provider/TSP domain, but not behind a NAT or LBL, as part of the MG to Gateway Controller association request.
- Listing of all NAT or LBL middleboxes
- Listing of all NAT or LBL middleboxes associated with a specified Gateway Controller
- Listing of parameters for a specified NAT or LBL middlebox
- Add a new Media Proxy Group and the Media Proxies associated with that group (new for (I)SN09).
- Query Media Proxy Group (new for (I)SN09).
  - — List all the Media Proxies within a Media Proxy Group.
  - — List all Media Proxy Groups assigned against a Gateway Controller.
  - — List all the Media Proxy Groups that a Media Proxy belongs to.
- Change Media Proxy Group - Modify the list of Media Proxies assigned to a group (new for (I)SN09).
- Delete Media Proxy Group - Delete a Media Proxy group (new for (I)SN09).
- Listing of all Media Proxies
- Listing of all Media Proxies associated with a specified Gateway Controller
- Listing of parameters for a specified Media Proxy
- Add a NAT, LBL or Composite NAT plus LBL network zone
- Delete a NAT, LBL or Composite NAT plus LBL network zone

- Change the parent network zone associated with a NAT, LBL or Composite NAT plus LBL network zone.
- Query an individual network zone.
- Query all network zones.
- Add a new VPN with the specified name (new for (I)SN09).
- Delete a VPN from the list of VPNs (new for (I)SN09).

Note: All the above functions are also supported by the CS 2000 Management Tools Graphical User Interface.

In addition to the above described interfaces, the following PC interfaces utilized for external VCAC counting are known to OSSGate even though the functionality they describe is unsupported. The interfaces are known to OSSGate due to a desire to support a common XSD between the PC and the CS 2000 management system. Any attempts to use these interface will result in error 306 being returned to indicate this.

- Add a link layer type
- Delete a link layer type.
- Change a link layer type
- Query a link layer type.

Terminology and Description of Method Parameters

version

The version refers to the version of the method or operation - e.g., the addNAT operation may have version 1.0 and 2.0, where version 1.0 takes parameters X and Y and version 2.0 takes parms X, Y, and Z. TYPE decimal.

usn

Unique Sequence Number. This parameter is introduced to facilitate sending multiple XML commands in one file. This parameter value will be used to match the responses with the requests. TYPE integer.

CallAgentId

The unique integer that has been assigned by the user to this Call Agent (e.g. this CS 2000) This must be a value between 1 and 255. TYPE integer.

CounterGWC

The VCAC counter GWC IP address. TYPE IPAddressType

GWCname

The name of the GWC. String of form GWC- followed by an integer 0-255 range.

IPAddress

The Media Proxy IP address.

ListNATid

Indication of whether the NAT id should be included in the response to this command. Set to ‘true’ if the NATid is required and ‘false’ if not.

itransMiddleboxName

The name of the adjacent Middlebox (either NAT or LBL) for the gateway command. 32 character token without spaces.

LBLname

The name of the LBL that this command applies to. 64 character token without spaces. Type NoNameSpaceType.

LBLid

The unique middlebox id for this LBL. This must be an integer between 2 and 16777215. NB the allowed range for LBLid is smaller that the allowed range for the data type MiddleBoxIndexType.

MaxCount

The VCAC count maximum limit. TYPE RUMaxCountType.

MPname

The name of the Media Proxy that this command applies to. TYPE NoSpaceNameType (32 character token without spaces)

NATname

The name of the NAT that this command applies to. TYPE NoSpaceNameType.

NATid

The unique middlebox id for this NAT. This must be an integer between 2 and 16777215. TYPE MiddleBoxIndexType. The allowed range for NATid is smaller that the allowed range for the data type MiddleBoxIndexType.

NATType

Indicates the type of NAT, in particular, NATType = 1 indicates an uncontrolled NAT. No other values are used in this release. TYPE string.

ParentMB

The name of the middlebox that is the parent of the NAT. 64 character token without spaces. Type NoSpaceNameType.

Protocol

The Media Proxy communication protocol (GWC to MP). This must be either MGCP+ or MPCP. TYPE ProtocolType.

ProtocolVersion

The Media Proxy communication protocol version. Version is restricted to a value of 2.0 for MGCP+ or a value of 3.0 for MPCP. TYPE ProtocolVersionType.

rootMiddleboxName1(1..5)

The Middlebox name(s) of the root node middlebox(es) that encompasses the enterprise network that the CICM gateway operates with 64 character token without spaces

ReturnCode value

A numeric value indicating the result of the attempted operation. TYPE string.

Return code text

Appropriate text that describes the result of the operation. TYPE string.

RUDescription

a text description for the RU Profile (maximum length of 64 characters). TYPE string.

RUValue

a composite of the following parameters. Codec, PacketRate, Value. TYPE RUValueParamsType

Service

Describes the service type supported by a network zone that is not an LBL. Type NZServiceType.

ID

Indicates the ID for a given network zone. Type NZIDType (unsigned integer minimum value = 2). If it is not specified it will be generated.

Name

Describes the name for a given network zone. Type DNSNameType (string conforming to DNS naming conventions, maximum length = 32).

Parent

describes a network zone parent. Defined as a choice allowing the parent to be described by ParentID or ParentName. Type ParentType.

ParentID

choice member of ParentType. Describes a parent network zone based on its ID. Type unsigned Int.

ParentName

choice member of ParentType. Describes a parent network zone based on its ID. Type unsigned Int.

IntraZoneGWInfo

Composite of BWCIR and NNSC. These fields are only relevant to the PC and will be ignored by OSSGate but must be correctly populated in accordance with the XSD interface if present. Type BWInfoType (addNetworkZone) or IntroZoneBWInfoType (changeNetworkZone).

LogicalNetworkLink

Composite of BandwidthInfo, LayerInfo and delete (changeNetworkZone only). These fields are only relevant to the PC and will be ignored by OSSGate, other than to indicate that the network zone possess LBL characteristics. When present it must be correctly populated. Type AddLogicalLinkType (addNetworkZone) or ChangeLogicalLinkType (changeNetworkZone).

IDMin

filter describing the minimum network zone ID to be returned as part of a query operation. Type unsigned Int.

IDMax

filter describing the maximum network zone ID to be returned as part of a query operation. Type unsigned Int.

MaxZones

filter describing the maximum number of network zones to be returned as part of a query operation. Type unsigned Int.

NetworkZone

Composite of ID, Name, and ParentID

Service

Identifies Network Zone as being a NAT or Composite type Network Zone. For a Composite Network Zone the LogicalNetworkLink tag should also be present.

algName

The name of the ALG node middlebox. TYPE NoSpaceNameType (32 character token without spaces)

ALGipaddress

The ALG IP address

ALGPort

The Port used by ALG Integer data type 0-65535

ALGProtocol

The protocol used. Only valid value is NCS.

Description of Method Parameters

Method parameters are defined below. Input data is mandatory except where indicated otherwise.

Step Action

1 queryCallAgentId - Method that gets the current value of the Call Agent Id for this Call Agent (CS 2000). This method has the following parameters:

- Input data
  - usn
  - version
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - CallAgentId - The value of the Call Agent Id for this Call Agent.

- ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information

2 setCallAgentId - Method that sets the Call Agent Id for this Call Agent (CS 2000) to the specified value. This method has the following parameters:

- Input data
  - usn
  - version
  - CallAgentId - the desired value for the Call Agent Id for this Call Agent (CS 2000)
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.

3 queryNAT - Method that gets the information regarding a NAT or set of NATs, depending on the parameters specified. If neither a NATname nor a GWCname is included, then information is returned for ALL NATs provisioned in this CS 2000 Mgr. This method has the following parameters:

- Input data
  - usn
  - version
  - NATname (optional, cannot be included as well as GWCname) - indicates the NAT for which information should be returned
  - GWCname (optional, cannot be included as well as NATname) - returns information for all NATs associated with this GWC
  - ListNATid (optional) - if set to true, this returns the NAT id as part information for each NAT, if set to false or if the parameter is omitted the NAT id is not returned.
- Output data
  - usn (value should be the same as the input)

  - version (value should be the same as the input)
  - NATname
  - NATType (returned only when the NATname is specified)
  - NATid (returned only when ListNATid is set to ‘true’ in the input data)
  - ParentMB (if the NAT has no parent middlebox, the parent middlebox tag is empty. For example, <ParentMB></ParentMB>). returned only when the NATname is specified OR ListNATid is set to ’true’

4 addNAT - Method that provision a NAT into the CS 2000 Mgr. This method has the following parameters: If the NAT id is not specified then the CS 2000 Mgr generates a NAT middlebox id for the NAT.

- Input data
  - usn
  - version
  - NATname - unique name for the NAT to be provisioned
  - NATid (optional) - indicates the middlebox id that should be allocated to this NAT.
  - ParentMB (optional) - indicates that the NAT has a parent middlebox. The parent middlebox specified here must already have been provisioned into the CS 2000 Mgr.
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - ReturnCode

5 deleteNAT - Method that removes a NAT from the CS 2000 Mgr. This method has the following parameters:

- Input data
  - usn
  - version
  - NATname - unique name for the NAT to be removed
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)

— ReturnCode

6 changeNAT - Method that modifies the provisioned details of a NAT in the CS 2000 Mgr. This method has the following parameters:

- Input data
  - — usn
  - — version
  - — NATname - unique name identifying the NAT to be changed
  - — ParentMB (optional) - indicates that the NAT has a parent middlebox. The parent middlebox specified here must already have been provisioned into the CS 2000 Mgr.
- Output data
  - — usn (value should be the same as the input)
  - — version (value should be the same as the input)
  - — ReturnCode

7 queryLBL - Method that gets the information regarding a LBL or set of LBLs, depending on the parameters specified. If neither a LBLname nor a GWCname is included, then information is returned for ALL LBLs provisioned in this CS 2000 Mgr. This method has the following parameters:

- Input data
  - — usn
  - — version
  - — LBLname (optional, cannot be included as well as GWCname) - indicates the LBL for which information should be returned
  - — GWCname (optional, cannot be included as well as LBLname) - returns information for all LBLs associated with this GWC
- Output data
  - — usn (value should be the same as the input)
  - — version (value should be the same as the input)
  - — LBLname
  - — LBLType (returned only when the LBL name is specified)
  - — The Counter GWC IPAddress

- The RU Description
- Max Count Value
- ParentMB (an empty value is returned if there is no parent Middlebox. For example <ParentMB></ParentMB>

8 addLBL - Method that provision a LBL into the CS 2000 Mgr. This method has the following parameters:

- Input data
  - usn
  - version
  - LBLname - unique name for the LBL to be provisioned
  - LBLid (optional) - indicates the middlebox id that should be allocated to this LBL.
  - The Counter GWC IPAddress (optional).
  - The RU Description - must match a description of an RU Profile already provisioned into the CS 2000 Mgr.
  - Max Count Value
  - ParentMB (optional) - indicates that the LBL has a parent middlebox. The parent middlebox specified here must already have been provisioned into the CS 2000 Mgr.
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - ReturnCode

9 deleteLBL - Method that removes a LBL from the CS 2000 Mgr. This method has the following parameters:

- Input data
  - usn
  - version
  - LBLname - unique name for the LBL to be removed
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - ReturnCode

10 changeLBL - Method that modifies the provisioned details of a LBL in the CS 2000 Mgr. This method has the following parameters:

- Input data
  - usn
  - version
  - LBLname - unique name identifying the LBL to be changed.
  - The Counter GWC IPAddress (optional).
  - The RU Description (optional) - must match a description of an RU Profile already provisioned into the CS 2000 Mgr.
  - Max Count Value (optional).
  - ParentMB (optional) - indicates that the NAT has a parent middlebox. The parent middlebox specified here must already have been provisioned into the CS 2000 Mgr.
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - ReturnCode

11 queryMP - Method that gets the information regarding a MP or set of MPs, depending on the parameters specified. If neither a MPname nor a GWCname is included, then information is returned for ALL MPs provisioned in this CS 2000 Mgr. This method has the following parameters:

- Input data
  - usn
  - version
  - MPname (optional, cannot be included as well as GWCname) - indicates the MP for which information should be returned
  - GWCname (optional, cannot be included as well as MPname) - returns information for all MPs associated with this GWC
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - MPname
  - IPAddress

- Protocol
- Protocol version

12 addMP - Method that provision a MP into the CS 2000 Mgr. This method has the following parameters:

- Input data
  - usn
  - version
  - MPname - unique name for the NAT to be provisioned
  - IPAddress - the address must not be assigned already to another MP in the CS 2000 Mgr
  - Protocol
  - Protocol version
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - ReturnCode

13 deleteMP - Method that removes a MP from the CS 2000 Mgr. This method has the following parameters:

- Input data
  - usn
  - version
  - MPname - unique name for the MP to be removed
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - ReturnCode

14 changeMP - Method that modifies the provisioned details of a NAT in the CS 2000 Mgr. This method has the following parameters:

- Input data
  - usn
  - version

  - MPname - unique name identifying the MP to be changed
  - IPAddress - the new address must not be assigned already to another MP in the CS 2000 Mgr
  - Protocol
  - Protocol version
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - ReturnCode

15 changeRootMiddleboxes - Method that changes the rootMiddleboxes for a given gateway. This method has the following parameters:

- Input data
  - usn
  - version
  - MGname - indicates the gateway that this change command is for.
  - rootMiddleboxName1 - defines the 1st RootMiddlebox.
  - rootMiddleboxName2 - (Optional) defines the 2nd RootMiddlebox.
  - rootMiddleboxName3 - (Optional) defines the 3rd Root Middlebox.
  - rootMiddleboxName4 - (Optional) defines the 4th Root Middlebox.
  - rootMiddleboxName5 - (Optional) defines the 5th Root Middlebox.
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.

16 addNetworkZone - Method that adds a new network zone of type NAT, LBL or Composite (containing the attributes of both a NAT and LBL simultaneously)

- Input data
  - usn
  - version
  - ID - (Optional) defines the ID for a new network zone. The Call Agent ID must be defined for this parameter to be passed as valid.
  - Name - defines the name of a new network zone.
  - Service - (Optional) defines the service type supported by this network zone. Only NAT is supported. The presence of Service defines the network zone as being either a NAT or Composite type.
  - Parent - (Optional) defines the parent network zone associated with the new network zone. Parent network zone may be described by either the ParentName (network zone Name) or ParentID (network zone ID) tags.
  - IntraZoneBWInfo - (Optional) the attributes contained within IntraZoneBWInfo are only of relevance to the PC and will be ignored by OSSGate if supplied.
  - LogicalNetworkLink - (Optional) the presence of this tag is interpreted as indicating that the network zone is of type LBL or Composite (in conjunction with Service). The actual parameters within the LogicalNetworkLink are only of relevance to the PC and will be ignored by OSSGate if supplied.
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.

17 changeNetworkZone - Method that modifies and existing network zone.

> ***Note:*** Only the associated parent network zone may be modified by this interface. This is a subset of the functionality supported by the PC and the XSD itself. Operations that are unsupported by OSSGate but legal for the PC will be rejected.

- Input data
  - usn

- — version
- — ID - (Optional) describes the ID of the network zone to be changed. Request rejected if specified in conjunction with Name as modification of a network zone name is currently unsupported by OSSGate.
- — Name - (Optional) describes the name of the network zone to be modified.
- — Parent - (Optional) defines the parent network zone associated with the new network zone. Parent network zone may be described by either the ParentName (network zone Name) or ParentID (network zone ID) tags.
- — IntraZoneBWInfo - (Optional) the attributes contained within IntraZoneBWInfo are only of relevance to the PC and will be ignored by OSSGate if supplied.
- — LogicalNetworkLink - (Optional) the attributes contained within the LogicalNetworkLink are only of relevance to the PC and will have no impact on OSSGate. However checking is imposed to detect a change of type to/from NAT and Composite for this network zone as this is unsupported functionality for OSSGate. Attempts to change type will be rejected to prevent OSSGate / PC mismatch.

- • Output data
  - — usn (value should be the same as the input)
  - — version (value should be the same as the input)
  - — ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.

18 deleteNetworkZone - Method that deletes an existing network zone.

- • Input data
  - — usn
  - — version
  - — ID - (Optional) describes the ID of the network zone to be deleted (ID or Name must be specified).
  - — Name - (Optional) describes the name of the network zone to be deleted (ID or Name must be specified).
- • Output data
  - — usn (value should be the same as the input)

- version (value should be the same as the input)
- ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.

19 queryNetworkZone - Method that query one or more network zones.

- Input data
  - usn
  - version
  - ID - (Optional) describes the ID of the network zone to be queried. Mutually exclusive with Name. Restricts query to a single unique network zone.
  - Name - (Optional) describes the name of the network zone to be queried. Mutually exclusive with ID. Restricts query to a single unique network zone.
  - IDMin - (Optional) acts as a filter to restrict return data from a query. Specifies the minimum ID for a network zone to be returned.
  - IDMax - (Optional) acts as a filter to restrict the return data from a query. Specifies the maximum ID for a network zone to be returned.
  - MaxZones - (Optional) acts as a filter to restrict the return data from a query. Specifies an upper limit on the number of network zones that may be returned.
- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.
  - ID - the ID for the returned network zone. Contained within a NetworkZone tag where query targets multiple network zones (ID / Name tags not present in input data).
  - Name - the name of the returned network zone. Contained within a NetworkZone tag where query targets multiple network zones.
  - ParentID - the ID of the parent network zone (0 if no parent). Contained within a NetworkZone tag where query targets multiple network zones.

— Service - (Optional) returned where the network zone is of type NAT / Composite *and* the query targets an individual network zone by specifying the Name / ID tags.

— LogicalNetworkLink - (Optional) returned where the network zone is of type LBL / Composite *and* the query targets an individual network zone. The LogicalNetworkLink is data filled with dummy data as OSSGate has no knowledge of PC specific data.

—End—

PC supported interfaces

The PC platform supports a subset of the Internet Transparency interfaces that are supported via OSSGate for the CS 2000 management system. Those interfaces only supported by the CS 2000 management system should not be sent to the PC. The following interfaces are supported by both the PC and OSSGate:

- addNetworkZone
- changeNetworkZone
- deleteNetworkZone
- queryNetworkZone

In addition the following interfaces are supported by the PC but not from OSSGate for the CS 2000 management system:

- addLinkLayerType
- changeLinkLayerType
- deleteLinkLayerType
- queryLinkLayerType

Query Call Agent identifier command

XML command queryCallAgentId

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryCallAgentId usn="1" version="1.0"/>
</Methods>
</Command>
</CommandList>
```

XML Response for queryCallAgentId

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryCallAgentId usn="1" version="1.0">
<ReturnData>
<CallAgentId>3</CallAgentId>
<ReturnCode value="0" text="Successful result" />
</ReturnData>
</queryCallAgentId>
</Methods>
</Response>
</CommandList>
```

Set Call Agent identifier command

XML command setCallAgentId

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<setCallAgentId usn="1" version="1.0">
<Parameters>
<CallAgentId>3</CallAgentId>
</Parameters>
</setCallAgentId>
</Methods>
</Command>
</CommandList>
```

XML Response for setCallAgentId

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<setCallAgentId usn="1" version="1.0">
<ReturnData>
<ReturnCode value="0" text="Successful result"/>
</ReturnData>
</setCallAgentId>
</Methods>
</Response>
</CommandList>
```

Associate Gateway with adjacent middlebox command

The association MG to adjacent MB is only allowed if the gateway supports the Internet Transparency service type (ITRANS). The OSS can associate a Media Gateway with a Middle Box and specify if the gateway is inside or outside the Telecoms Service Providers (IN/OUT TSP) domain. This is used to determine if the Media Proxy NAT services are required or the VCAC counting service should be enabled for a call flow. A specific Middlebox "outtsp" is reserved to indicate that the MG is outside the TSP/Carrier VoIP Site but NOT behind a NAT. An additional field is added to the existing assocMG method XML query. The new element <itransMiddleboxName> is optional. When used, it must contain the name of an existing NAT or LBL middle box.

An example of assocMG command with adjacent Middlebox addition

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<assocMG usn="1" version="1.0">
<Parameters>
<mgUIName>mg4.nortel.net</mgUIName>
<mgProfileName>ASKEY_LINE_GW_4</mgProfileName>
<mgIpAddr>47.49.23.48</mgIpAddr>
<mgProtocolType>5</mgProtocolType>
<mgProtocolVersion>2.0</mgProtocolVersion>
<mgProtocolPort>2427</mgProtocolPort>
<gwcUIName>GWC-0</gwcUIName>
<itransMiddleboxName>myNat1</itransMiddleboxName>
</Parameters>
</assocMG>
</Methods>
</Command>
</CommandList>
```

An example of assocMG response with adjacent Middlebox addition

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<assocMG usn="1" version="1.0">
<ReturnData>
<Row>
<gwcUIName>GWC-0</gwcUIName>
<gwcIpAddr>172.17.40.24</gwcIpAddr>
<mgNodeName>LG 00 0</mgNodeName>
</Row>
<RC>0</RC>
```

```
<MsgTxt>The MG was successfully associated with a
GWC</MsgTxt>
</ReturnData>
</assocMG>
</Methods>
</Response>
</CommandList>
```

If the Media Gateway to be added is outside the Carrier VoIP VPN and not behind a NAT or LBL, the itransMiddleboxName XML tag must be used with a value of “outtsp”.

Query MG command

An additional field is added to the existing QueryMG method XML response. The new element <itransMiddleBoxName> is returned when a NAT or LBL middle box is associated with the gateway. When no NAT or LBL is associated to the gateway, an empty tag is returned.

An example of queryMG command response with NAT

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryMG usn="1" version="1.0">
<ReturnData>
<Row>
<mgUIList>mg4.nortel.net</mgUIList>
<mgIpList>47.49.23.48</mgIpList>
<callServerId>RTPS</callServerId>
<gwcUIName>GWC-0</gwcUIName>
<svcTypeList>0</svcTypeList>
<nodeName>LG    00 0</nodeName>
<itransMiddleboxName>myNat1i</transMiddleboxName>
<engrEndPoints>4</engrEndPoints>
<protType>mgcp</protType>
<protVersion>2.0</protVersion>
<protPort>2427</protPort>
<profileName>ASKEY_LINE_GW_4</profileName>
</Row>
<RC>0</RC>
<MsgTxt>Query of a single MG was successful</MsgTxt>
</ReturnData>
</queryMG>
</Methods>
</Response>
</CommandList>
```

Note: When the MG is outside the Carrier VoIP VPN and not behind a NAT, the itransMiddleboxName value is set to “outtsp” (‘outside telecom service provider VPN’).

Associate Gateway with Root Middleboxes command

The association MG to root MBs is only allowed if the gateway supports the Internet Transparency service type (ITRANS_ROAM). This is used to determine if the Media Proxy NAT services are required or the VCAC counting service should be enabled for a call flow. Additional fields are added to the existing assocMG method XML query, these are defined in section "Associate MG" (page 120). The new elements <rootMiddleboxName1>, <rootMiddleboxName2>, <rootMiddleboxName3>, <rootMiddleboxName4> and <rootMiddleboxName5> are optional. When used, they must contain the name of an existing NAT or LBL middle box.

An example of assocMG command with root Middleboxes addition

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<assocMG usn="1" version="1.0">
<Parameters>
mgUIName>CICM-022</mgUIName>
<mgProfileName>CICM</mgProfileName>
<mgIpAddr>11.13.45.58</mgIpAddr>
<mgProtocolType>4</mgProtocolType>
<mgProtocolVersion>1.0</mgProtocolVersion>
<mgProtocolPort>2944</mgProtocolPort>
<gwcUIName>GWC-107</gwcUIName>
<reservedTerminations>1023</reservedTerminations>
<rootMiddleboxName1>nat1</rootMiddleboxName1>
<rootMiddleboxName2>nat2</rootMiddleboxName2>
<rootMiddleboxName3>nat3</rootMiddleboxName3>
<rootMiddleboxName4>nat4</rootMiddleboxName4>
<rootMiddleboxName5>lbl1</rootMiddleboxName5>
</Parameters>
</assocMG>
</Methods>
</Command>
</CommandList>
```

QueryMG command (Root Middleboxes returned)

Additional fields are added to the existing QueryMG method XML response. The new root Middlebox Names are returned when root middleboxes are associated with the gateway. When no root Middleboxes are associated to the gateway, empty tags are returned.

An example of queryMG command response with root MBs

```
<?xml version="1.0" encoding"UTF-8" ?>
```

```
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryMG usn="1" version="1.0">
<ReturnData>
<Row>
<mgUIList>CICM-322</mgUIList>
<mgIpList>11.13.45.58</mgIpList>
<callServerId>telco</callServerId>
<gwcUIName>GWC-222</gwcUIName>
<svcTypeList>0</svcTypeList>
<svcTypeList>7</svcTypeList>
<nodeName>LG</nodeName>
<engrEndPoints>1023</engrEndPoints>
<protType>megaco</protType>
<protVersion>1.0</protVersion>
<protPort>2944</protPort>
<profileName>CICM</profileName>
<maxEngrEndPoints>3069</maxEngrEndPoints>
<rootMiddleboxName1>lbl1</rootMiddleBoxName1>
<rootMiddleboxName2>nat1</rootMiddleBoxName2>
<rootMiddleboxName3>nat2</rootMiddleBoxName3>
<rootMiddleboxName4/>
<rootMiddleboxName5/>
</Row>
<RC>0</RC>
<MsgTxt>Query of a single MG was successful</MsgTxt>
</ReturnData>
</queryMG>
</Methods>
</Response>
</CommandList>
```

Add NAT command

A NAT can be added to the CS 2000 Management System by using a unique name.

An example of AddNat command request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransLf</Interface>
<Methods>
<addNat usn="1" version="1.0">
<Parameters>
<NATname>Nat-B</NATname>
<NATid>12345</NATid>
<ParentMB>Nat-A</ParentMB>
<PreferredMPGroup>MPG1</PreferredMPGroup>
```

```
</Parameters>
</addNAT>
</Methods>
</Command>
</CommandList>
```

An example of AddNat response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<addNAT usn="1" version="1.0">
<ReturnData>
<Return Code value ="0" text="Successful result"/>
</ReturnData>
</addNAT>
</Methods>
</Response>
</CommandList>
```

Note: Attempting to add a NAT name already added to the system will fail with a message "Failed to add the Middlebox because Name specified is already in use: Nat-B."

Depending on the parameters used in the XML query the response returns one of the following:

- Success result
- NAT name already in use.
- NAT id already in use
- Unknown Parent Middlebox

Delete NAT command

A provisioned NAT can be deleted from the CS 2000 management system by passing the NAT name. A existing NAT will be deleted only if the NAT has no association to a media gateway.

An example of deleteNAT command request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransLf</Interface>
<Methods>
<deleteNAT usn="1" version="1.0">
<Parameters>
<NATname>NAT-A</NATname>
</Parameters>
```

```
</deleteNAT>
</Methods>
</Command>
</CommandList>
```

An example of deleteNAT response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<deleteNAT usn="1" version="1.0">
<ReturnData>
<Return Code value ="0" text="Successful result"/>
</ReturnData>
</deleteNAT>
</Methods>
</Response>
</CommandList>
```

Depending on the parameters used in the XML query the response returns:

- Success result
- MiddleBox name not found

> ***Note:*** If a NAT is provisioned into multiple CS 2000s care should be taken when deleting the NAT. If the NAT middlebox ID used for the shared NAT has been automatically generated by one CS 2000 Mgr, the NAT should not be deleted from that CS 2000 Mgr while it is still provisioned in other CS 2000s. It must be deleted from all other CS 2000 Mgrs first.

Change NAT command

A provisioned NAT can be changed on the CS 2000 management system by passing the NAT name and a new parent middlebox name. The MediaProxyGroup or VPN associated with the NAT can be changed as well.

XML command to Change specified NAT parent middlebox

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransLf</Interface>
<Methods>
<changeNAT usn="1" version="1.0">
<Parameters>
<NATname>NAT-A</NATname>
<ParentMB>VPN-Eastenders</ParentMB>
<PreferredMPGroup>MPG2</PreferredMPGroup>
</Parameters>
</changeNAT>
</Methods>
```

```
</Command>
</CommandList>
```

XML response to Change specified NAT

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<changeNAT usn="1" version="1.0">
<ReturnData>
<Return Code value ="0" text="Successful result"/>
</ReturnData>
</changeNAT>
</Methods>
</Response>
</CommandList>
```

Depending on the parameters used in the XML query the response returns:

- Success result
- MiddleBox name not found

Query NAT command

The queryNAT command returns

- information concerning a given NAT
- names of the NAT used on a given GWC
- names of all NATs provisioned
- names of all NATs having a specified Media Proxy group

The information regarding each NAT returned is as follows:

- NAT name
- NAT type - only in queryNAT request in which a single NAT is specified
- NAT parent middlebox. If the NAT has no parent middlebox, the parent middlebox tag is empty (for example, <ParentMB></ParentMB>) - only in queryNAT request in which a single NAT is specified OR if ListNATid tag is "true"

An additional optional tag may be specified in the queryNAT command, the ’ListNATid’ tag. If this tag is present and has a value true, the NAT middlebox identifier is also listed for each NAT returned. This tag may be present in all the forms of the queryNAT command, for example, for a specified NAT, on a given GWC and for all provisioned NATs. If the tag is not present, middlebox identifiers are not listed.

An example of query one NAT request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransLf</Interface>
<Methods>
<queryNAT usn="1" version="1.0">
<Parameters>
<NATname>myNat1</NATname>
</Parameters>
</queryNAT>
</Methods>
</Command>
</CommandList>
```

An example of query one NAT response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryNAT usn="1" version="1.0">
<ReturnData>
<NATname>myNat1</NATname>
<NATType>1<NATType>
<ParentMB>parentNat</ParentMB>
<Return Code value ="0" text="Successful result"/>
</ReturnData>
</queryNAT>
</Methods>
</Response>
</CommandList>
```

An example of query one NAT request, with NAT id

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryNAT usn="1" version="1.0">
<Parameters>
<NATname>myNat1</NATname>
<ListNATid>true</ListNATid>
</Parameters>
</queryNAT>
</Methods>
</Command>
</CommandList>
```

An example of query one NAT response, including NAT id

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryNAT usn="1" version="1.0">
<ReturnData>
<NATname>myNat1</NATname>
<NATType>1</NATType>
<ParentMB>parentNat</ParentMB>
<NATid>65538</NATid>
<Return Code value ="0" text="Successful result"/>
</ReturnData>
</queryNAT>
</Methods>
</Response>
</CommandList>
```

An example of query all NATs on a given GWC request, including NAT id

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryNAT usn="1" version="1.0">
<Parameters>
<GWCname>GWC-0>/GWCname>
<ListNATid>true</ListNATid>
</Parameters>
</queryNAT>
</Methods>
</Command>
</CommandList>
```

An example of query all NATs on a given GWC response, including NAT id

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryNAT usn="1" version="1.0">
<ReturnData>
<NATname>myNat1</NATname>
<NATid>234</NATid>
<ParentMB/>
<NATname>myNat2</NATname>
<NATid>234</NATid>
<ParentMB>aParentmb<ParentMB>
<Return Code value ="0" text="Successful result"/>
```

```
</ReturnData>
</queryNAT>
</Methods>
</Response>
</CommandList>
```

An example of query all NATs on a specified Media Proxy Group request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryNAT usn="1" version="1.0">
<Parameters>
<NATname>Nat1</NATname>
<preferredMPGroup>MPG1</preferredMPGroup>
</Parameters>
</queryNAT>
</Methods>
</Command>
</CommandList>
```

An example of query all NATs request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryNAT usn="1" version="1.0">
<Parameters>
</Parameters>
</queryNAT>
</Methods>
</Command>
</CommandList>
```

An example of query all NATs on a given GWC response, including NAT id

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryNAT usn="1" version="1.0">
<ReturnData>
<NATname>NAT1</NATname>
<NATname>NAT2</NATname>
<NATname>NAT3</NATname>
<NATname>NAT4</NATname>
<NATname>NAT5</NATname>
<Return Code value ="0" text="Successful result"/>
```

```
</ReturnData>
</queryNAT>
</Methods>
</Response>
</CommandList>
```

Add LBL command

An LBL can be added to the CS 2000 management system by using a unique name.

XML command to Create an LBL device

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<addLBL usn="1" version="1.0">
<Parameters>
<LBLname>LBL1</LBLname>
<CounterGWC>47.132.132.12</CounterGWC>
<RUDescription>AAL1/ip40</RUDescription>
<MaxCount>1000</MaxCount>
<ParentMB>VPN-A</ParentMB>
<PreferredMPGroup>MPG1</PreferredMPGroup>
</Parameters>
</addLBL>
</Methods>
</Command>
</CommandList>
```

XML response to Creating an LBL device

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<addLBL usn="1" version="1.0">
<ReturnData>
<Return Code value ="0" text="Successful result"/>
</ReturnData>
</addLBL>
</Methods>
</Response>
</CommandList>
```

Depending on the parameters used in the XML query the response returns one of the following:

- Success result
- LBL already in use

- LBL id already in use
- Unknown Parent Middlebox

Delete LBL command

A provisioned LBL can be deleted from the CS 2000 management system by passing the LBL name. The existing LBL will be deleted only if the LBL has no association to a media gateway.

XML command to Delete an LBL device

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<deleteLBL usn="1" version="1.0">
<Parameters>
<LBLname>LBL1</LBLname>
</Parameters>
</deleteLBL>
</Methods>
</Command>
</CommandList>
```

XML response to Deleting an LBL device

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<deleteLBL usn="1" version="1.0">
<ReturnData>
<Return Code value ="0" text="Successful result"/>
</ReturnData>
</deleteLBL>
</Methods>
</Response>
</CommandList>
```

Depending on the parameters used in the XML query the response returns one of the following:

- Success result
- Middlebox name not found

Change LBL command

A provisioned LBL can be changed on the CS 2000 management system by passing the LBL name. The Media Proxy group can be changed as well.

XML command to Change an LBL device RU profile

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<changeLBL usn="1" version="1.0">
<Parameters>
<LBLname>LBL1</LBLname>
<RUDescription>AAL2/ip40</RUDescription>
</Parameters>
</changeLBL>
</Methods>
</Command>
</CommandList>
```

XML command to Change an LBL device Media Proxy Group

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<changeLBL usn="1" version="1.0">
<Parameters>
<LBLname>LBL1</LBLname>
<PreferredMPGroup>MPG2</PreferredMPGroup>
</Parameters>
</changeLBL>
</Methods>
</Command>
</CommandList>
```

XML response to Changing an LBL device

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<changeLBL usn="1" version="1.0">
<ReturnData>
<Return Code value ="0" text="Successful result"/>
</ReturnData>
</changeLBL>
</Methods>
</Response>
</CommandList>
```

Depending on the parameters used in the XML query the response returns one of the following:

- Success result
- Middlebox name not found

Query LBL command

The query LBL command returns:

- LBL name
- the Counter GWC IPAddress
- the RU description
- Max Count Value
- Parent MB name

XML command to Query an LBL, Query single LBL

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIfInterface>ITransIf>
<Methods>
<queryLBL usn="1" version="1.0">
<Parameters>
<LBLname>LBL1LBLname>LBL1>
</Parameters>
</queryLBL>
</Methods>
</Command>
</CommandList>
```

XML command to Query an LBL, Query all LBLs

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf>/Interface>
<Methods>
<queryLBL usn="1" version="1.0">
<Parameters>
</Parameters>
</queryLBL>
</Methods>
</Command>
</CommandList>
```

XML command to Query an LBL, MPG-based Query

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
```

```
<Interface>ITransIf>/Interface>
<Methods>
<queryLBL usn="1" version="1.0">
<Parameters>
<PreferredMPGroup>MPG1PreferredMPGroup>
</Parameters>
</queryLBL>
</Methods>
</Command>
</CommandList>
```

XML response to Query LBL

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryLBL usn="1" version="1.0">
<ReturnData>
<LBLname>LBL1</LBLname>
<CounterGWC>47.132.132.12</CounterGWC>
<RUDescription>AAL1/ip40</RUDescription>
<MaxCount>1000</MaxCount>
<ParentMB>VPN-A</ParentMB>
<Return Code value ="0" text="Successful result"/>
</ReturnData>
</queryLBL>
</Methods>
</Response>
</CommandList>
```

Add RU Profile command

An RU can be added to the CS 2000 management system by using a unique RU Description, and RUValue which includes the Codec, PacketRate and Value.

XML command to Create an RU profile

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<addRU usn="1" version="1.0">
<Parameters>
<RUDescription>AAL1/IP40</RUDescription>
<RUValue>
<Codec>G.711</Codec>
<PacketRate>10 ms</PacketRate>
<Value>103</Value>
</RUValue>
```

```
<RUValue>
<Codec>G.711</Codec>
<PacketRate>20 ms</PacketRate>
<Value>133</Value>
</RUValue>
<RUValue>
<Codec>G.729</Codec>
<PacketRate>10 ms</PacketRate>
<Value>132</Value>
</RUValue>
<RUValue>
<Codec>G.729</Codec>
<PacketRate>20 ms</PacketRate>
<Value>103</Value>
</RUValue>
</Parameters>
</addRU>
</Methods>
</Command>
</CommandList>
```

XML response to Creating an RU profile

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<addRU usn="1" version="1.0">
<ReturnData>
<Return Code value ="0" text="Successful result"/>
</ReturnData>
</addRU>
</Methods>
</Response>
</CommandList>
```

Depending on the parameters used in the XML query the response returns on of the following:

- Success result
- RU already in use

Delete RU Profile command

A provisioned RU can be deleted from the CS 2000 management system by passing the RU name. A existing RU will be deleted only if the RU has no association to an LBL.

XML command to Delete an RU profile

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
```

```
<Command>
<Interface>ITransIf</Interface>
<Methods>
<deleteRU usn="1" version="1.0">
<Parameters>
<RUDescription>AAL1/IP40</RUDescription>
</Parameters>
</deleteRU>
</Methods>
</Command>
</CommandList>
```

XML response to Deleting an RU profile

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<deleteRU usn="1" version="1.0">
<ReturnData>
<Return Code value ="0" text="Successful result"/>
</ReturnData>
</deleteRU>
</Methods>
</Response>
</CommandList>
```

Depending on the parameters used in the XML query the response returns one of the following:

- Success result
- RU name not found
- RU in use by an LBL

Change RU Profile command

A provisioned RU can be changed on the CS 2000 management system by passing the RU name.

XML command to Change RU profile values

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<changeRU usn="1" version="1.0">
<Parameters>
<RUDescription>AAL1/IP40</RUDescription>
<RUValue>
<Codec>G.711</Codec>
```

```
<PacketRate>10 ms</PacketRate>
<Value>123</Value>
</RUValue>
<RUValue>
<Codec>G.711</Codec>
<PacketRate>20 ms</PacketRate>
<Value>133</Value>
</RUValue>
<RUValue>
<Codec>G.729</Codec>
<PacketRate>10 ms</PacketRate>
<Value>152</Value>
</RUValue>
<RUValue>
<Codec>G.729</Codec>
<PacketRate>20 ms</PacketRate>
<Value>163</Value>
</RUValue>
</Parameters>
</changeRU>
</Methods>
</Command>
</CommandList>
```

XML response to Changing an RU profile

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<changeRU usn="1" version="1.0">
<ReturnData>
<Return Code value ="0" text="Successful result"/>
</ReturnData>
</changeRU>
</Methods>
</Response>
</CommandList>
```

Depending on the parameters used in the XML query the response returns:

- Success result
- RU name not found

Query RU Profile command

The query RU command returns:

- RU Description
- Codec

- PacketRate
- Value

XML command to Query RU profiles

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryRU usn="1" version="1.0">
<Parameters>
</Parameters>
</queryRU>
</Methods>
</Command>
</CommandList>
```

XML response to Query all RU profiles

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryRU usn="1" version="1.0">
<ReturnData>
<RUDescription>AAL1/IP40</RUDescription>
<RUDescription>AAL2/IP40</RUDescription>
<RUDescription>AAL1/IP40-B</RUDescription>
<RUDescription>AAL2/IP40-B</RUDescription>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</queryRU>
</Methods>
</Response>
</CommandList>
```

XML command to Query RU profile data

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryRU usn="1" version="1.0">
<Parameters>
<RUDescription>AAL1/IP40</RUDescription>
</Parameters>
</queryRU>
</Methods>
</Command>
</CommandList>
```

XML response to Query an RU profile

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryRU usn="1" version="1.0">
<ReturnData>
<RUDescription>AAL1/IP40</RUDescription>
<RUValue>
<Codec>G.711</Codec>
<PacketRate>10 ms</PacketRate>
<Value>123</Value>
</RUValue>
<RUValue>
<Codec>G.711</Codec>
<PacketRate>20 ms</PacketRate>
<Value>133</Value>
</RUValue>
<RUValue>
<Codec>G.729</Codec>
<PacketRate>10 ms</PacketRate>
<Value>152</Value>
</RUValue>
<RUValue>
<Codec>G.729</Codec>
<PacketRate>20 ms</PacketRate>
<Value>163</Value>
</RUValue>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</queryRU>
</Methods>
</Response>
</CommandList>
```

Add MP Group

Add Media Proxy Group is a new command in SN09. The command defines a new group and adds between one and five Media Proxies to that group.

XML command to Add Media Proxy Group

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<addMPGroup usn="1" version="1.0">
<Parameters>
<MPGroupName>aGroup</MPGroupName>
<MPname>mp1</MPname>
```

```
<MPname>mp11</MPname>
<MPname>mp111</MPname>
<MPname>mp1111</MPname>
<MPname>mp11111</MPname>
</Parameters>
</addMPGroup>
</Methods>
</Command>
</CommandList>
```

XML response to Add an Media Proxy Group

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<addMPGroup usn="1" version="1.0">
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</addMPGroup>
</Methods>
</Response>
</CommandList>
```

Query MP Group

Query Media Proxy Group is a new command in (I)SN09. Three versions of the command provide queries to return the following information:

- The list of Media Proxies contained in a Media Proxy Group
- List all Media Proxy Groups
- The list of Media Proxy Groups to which a Media Proxy belongs
- The list of Media Proxy Groups associated with a Gateway Controller

The three versions of XML for this command are shown in below.

XML command to Query MP Group

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryMPGroup usn="1" version="1.0">
<Parameters>
<MPGroupName>aGroup</MPGroupName>
</Parameters>
</queryMPGroup>
</Methods>
</Command>
```

```
</CommandList>
```

Get all MPG entries

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryMPGroup usn="1" version="1.0">
<Parameters>
</Parameters>
</queryMPGroup>
</Methods>
</Command>
</CommandList>
```

Get MPGs to which the MP belongs

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryMPGroup usn="1" version="1.0">
<Parameters>
<MPname>mp1</MPname>
</Parameters>
</queryMPGroup>
</Methods>
</Command>
</CommandList>
```

Get MPGs that are on a GWC

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryMPGroup usn="1" version="1.0">
<Parameters>
<GWCname>GWC-1</GWCname>
</Parameters>
</queryMPGroup>
</Methods>
</Command>
</CommandList>
```

Two types of response are provided, the detailed single MPG response, and the multiple MPG name response. The single response is only provided when a single MPG is requested.

Single MPG response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryMPGroup usn="1" version="1.0">
<ReturnData>
<MPGGroupName>MPG-ABC</MPGGroupName>
<MPname>MP-ABC</MPname>
<MPname/>
<MPname/>
<MPname/>
<MPname/>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</queryMPGroup>
</Methods>
</Response>
</CommandList>
```

Multiple MPG response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryMPGroup usn="1" version="1.0">
<ReturnData>
<MPGGroupName>MPG-ABC</MPGGroupName>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</addMPGroup>
</Methods>
</Response>
</CommandList>
```

Change MP Group

Change Media Proxy Group is a new command in (I)SN09. The command redefines the list of Media Proxies contained within a Media Proxy Group. The new list will replace the previous list.

XML command to Change MP Group

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<changeMPGroup usn="1" version="1.0">
<Parameters>
```

```
<MPGroupName>aGroup</MPGroupName>
<MPname>mp1</MPname>
<MPname>mp2</MPname>
</Parameters>
</changeMPGroup>
</Methods>
</Command>
</CommandList>
```

XML response to Change MP Group

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<changeMPGroup usn="1" version="1.0">
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</changeMPGroup>
</Methods>
</Response>
</CommandList>
```

Delete MP Group

Delete Media Proxy Group is a new command in (I)SN09. The command deletes the definition of a Media Proxy Group.

XML command to Delete MP Group

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<deleteMPGroup usn="1" version="1.0">
<Parameters>
<MPGroupName>aGroup</MPGroupName>
</Parameters>
</deleteMPGroup>
</Methods>
</Command>
</CommandList>
```

XML response to Delete MP Group

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<deleteMPGroup usn="1" version="1.0">
```

```
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</deleteMPGroup>
</Methods>
</Response>
</CommandList>
```

Add MP command

A MP can be added to the CS 2000 management system by using a unique name.

XML command to Create a new Media Proxy

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<addMP usn="1" version="1.0">
<Parameters>
<MPname>MP-A</MPname>
<IPAddress>47.165.32.23</IPAddress>
<Protocol>MGCP+</Protocol>
<ProtocolVersion>2.0</ProtocolVersion>
</Parameters>
</addMP>
</Methods>
</Command>
</CommandList>
```

XML response to Creating a new Media Proxy command

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<addMP usn="1" version="1.0">
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</addMP>
</Methods>
</Response>
</CommandList>
```

Depending on the parameters used in the XML query the response returns one of the following:

- Success result
- MP already in use

Delete MP command

A provisioned MP can be deleted from the CS 2000 management system by passing the MP name. A existing MP will be deleted only if the MP has no association to a GWC gateway.

XML command to Delete a Media Proxy

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<deleteMP usn="1" version="1.0">
<Parameters>
<MPname>MP-A</MPname>
</Parameters>
</deleteMP>
</Methods>
</Command>
</CommandList>
```

XML response to Deleting a Media Proxy

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<deleteMP usn="1" version="1.0">
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</deleteMP>
</Methods>
</Response>
</CommandList>
```

Depending on the parameters used in the XML query the response returns:

- Success result
- MP name not found

Change MP command

A provisioned MP can be changed on the CS 2000 management system by passing the MP name.

XML command to Change the IP data of a Media Proxy

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
```

```
<changeMP usn="1" version="1.0">
<Parameters>
<MPname>MP-A</MPname>
<IPAddress>47.165.32.22</IPAddress>
<Protocol>MGCP+</Protocol>
<ProtVersion>2.0</ProtVersion>
</Parameters>
</changeMP>
</Methods>
</Command>
</CommandList>
```

XML response to Change Media Proxy IP data

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<changeMP usn="1" version="1.0">
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</changeMP>
</Methods>
</Response>
</CommandList>
```

Depending on the parameters used in the XML query the response returns:

- Success result
- MP name not found

Query MP command

The queryMP command returns:

- complete information concerning a given MP
- names of the Media Proxies used on a given GWC
- names of all Media Proxies provisioned

An example of query one Media Proxy request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryMP usn="1" version="1.0">
<Parameters>
<MPname>churchill</MPname>
</Parameters>
```

```
</queryMP>
</Methods>
</Command>
</CommandList>
```

An example of query one Media Proxy response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryMP usn="1" version="1.0">
<ReturnData>
<MPname>churchill</MPname>
<DeviceIP>1.1.1.1</DeviceIP>
<Protocol>MGCP+</Protocol>
<ProtocolVersion>2.0</ProtocolVersion>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</queryMP>
</Methods>
</Response>
</CommandList>
```

An example of query Media Proxies on a given GWC request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryMP usn="1" version="1.0">
<Parameters>
<GWCname>GWC-102</GWCname>
</Parameters>
</queryMP>
</Methods>
</Command>
</CommandList>
```

An example of query Media Proxies on a given GWC response

```
<?xml version="1.0" ?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryMP usn="1" version="1.0">
<ReturnData>
<MPname>admiral</MPname>
<MPname>churchill</MPname>
<ReturnCode value ="0" text="Successful result"/>
```

```
</ReturnData>
</queryMP>
</Methods>
</Response>
</CommandList>
```

An example of query all Media Proxies request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryMP usn="1" version="1.0">
<Parameters>
</Parameters>
</queryMP>
</Methods>
</Command>
</CommandList>
```

An example of query all Media Proxies response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryMP usn="1" version="1.0">
<ReturnData>
<MPname>admiral</MPname>
<MPname>churchill</MPname>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</queryMP>
</Methods>
</Response>
</CommandList>
```

Change Gateways associated NAT or LBL command

This method provides ability to change the adjacent NAT or LBL middle box associated with a gateway.

An example of Change middlebox associated with a GW request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<changeAssocMB usn="1" version="1.0">
<Parameters>
<MGname>NorthBrook1105.com</MGname>
```

```
<itransMiddleboxName>myNAT1</itransMiddleboxName>
</Parameters>
</changeAssocMB>
</Methods>
</Command>
</CommandList>
```

An example of query all Media Proxies response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<changeAssocMB usn="1" version="1.0">
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</changeAssocMB>
</Methods>
</Response>
</CommandList>
```

To disassociate a gateway from a NAT or LBL, omit the itransMiddleboxName XML tag. This way, the gateway will not be associated to any NAT.

To change the status of a gateway as being outside the Carrier VoIP VPN and not behind a NAT or LBL, set the itransMiddleboxName XML tag to “outtsp“.

Change Gateways associated root Middleboxes command

This method provides ability to change the root middle boxes associated with a gateway.

An example of Change root Middleboxes associated with a GW request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<changeRootMiddleboxes usn="1" version="1.0">
<Parameters>
<MGname>CICM-001</MGname>
<rootMiddleboxName1>nat1</rootMiddleboxName1>
<rootMiddleboxName2>nat2</rootMiddleboxName2>
<rootMiddleboxName3>nat3</rootMiddleboxName3>
</Parameters>
</changeRootMiddleboxes>
</Methods>
</Command>
</CommandList>
```

An example of Change root Middleboxes associated with a GW response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<changeRootMiddleboxes usn="1" version="1.0">
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</changeRootMiddleboxes>
</Methods>
</Response>
</CommandList>
```

To disassociate a gateway from root Middleboxes, omit the rootMiddleboxName1..5 XML tags. This way, the gateway will not be associated to any root Middleboxes.

Add Network Zone command

A Network zone can be added to the CS 2000 management system by using a unique name and optionally an ID.

Network Zones should added to the system in situations there is a Policy Controller deployed and commissioned and where Network VCAC has been enabled.

In order for a Network Zones ID to be specified at time of creation, the Call Agent ID should be setup within the CS200 management system. As the PC platform requires the ID to be specified it is recommended that this be done prior to the creation of network zones.

Network zones may be of type NAT, LBL and Composite (both NAT and LBL characteristics). XML examples are given for all variations:

An example of Add Network Zone (NAT type) request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<addNetworkZone usn="1" version="1.0">
<Parameters>
<Name>NZ1</Name>
<Service>NAT</Service>
<PreferredMPGroup>MPG1</PreferredMPGroup>
</Parameters>
</addNetworkZone>
</Methods>
</Command>
</CommandList>
```

An example of Add Network Zone (LBL type) request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<addNetworkZone usn="1" version="1.0">
<Parameters>
<ID>235</ID>
<Name>router2</Name>
<Parent>
<ParentName>router1</ParentName>
</Parent>
<IntraZoneBWInfo>
<BWCIR>321</BWCIR>
<NNSC>
<Premium>12</Premium>
<Platinum>23</Platinum>
<Gold>25</Gold>
<Silver>3</Silver>
<Bronze>5</Bronze>
</NNSC>
</IntraZoneBWInfo>
<LogicalNetworkLink>
<BandwidthInfo>
<Symmetry>false</Symmetry>
<UpBWInfo>
<BWCIR>456</BWCIR>
</UpBWInfo>
<DownBWInfo>
<BWCIR>997</BWCIR>
</DownBWInfo>
<LayerInfo>
<Layer>
<LinkLayerType>IP</LinkLayerType>
</Layer>
<Layer>
<Overhead>34</Overhead>
</Layer>
<Layer>
<Overhead>16</Overhead>
<UnitSize>53</UnitSize>
<UnitOverhead>5</UnitOverHead>
</Layer>
<Layer>
<LinkLayerType>Sonet</LinkLayerType>
</Layer>
</LayerInfo>
</LogicalNetworkLink>
<PreferredMPGroup>MPG1</PreferredMPGroup>
```

```
</Parameters>
</addNetworkZone>
</Methods>
</Command>
</CommandList>
```

An example of Add Network Zone (Composite type) request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<addNetworkZone usn="1" version="1.0">
<Parameters>
<ID>236</ID>
<Name>router3</Name>
<Service>NAT</Service>
<Parent>
<ParentID>234</ParentID>
</Parent>
<IntraZoneBWInfo>
<BWCIR>321</BWCIR>
<NNSC>
<Premium>12</Premium>
<Platinum>23</Platinum>
<Gold>25</Gold>
<Silver>3</Silver>
<Bronze>5</Bronze>
</NNSC>
</IntraZoneBWInfo>
<LogicalNetworkLink>
<BandwidthInfo>
<Symmetry>false</Symmetry>
<UpBWInfo>
<BWCIR>456</BWCIR>
</UpBWInfo>
<DownBWInfo>
<BWCIR>997</BWCIR>
</DownBWInfo>
<LayerInfo>
<Layer>
<LinkLayerType>IP</LinkLayerType>
</Layer>
<Layer>
<Overhead>34</Overhead>
</Layer>
<Layer>
<Overhead>16</Overhead>
<UnitSize>53</UnitSize>
<UnitOverhead>5</UnitOverHead>
```

```
</Layer>
<Layer>
<LinkLayerType>Sonet</LinkLayerType>
</Layer>
</LayerInfo>
</LogicalNetworkLink>
</Parameters>
</addNetworkZone>
</Methods>
</Command>
</CommandList>
```

An example of Add Network Zone response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<addNetworkZone usn="1" version="1.0">
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</addNetworkZone>
</Methods>
</Response>
</CommandList>
```

Note 1: The node provisioning application will ignore the details of the Logical Network Link information since it does not need it. This information is required by the Policy Controller. If details of Logical Network Link information are included, Network VCAC must be on for the command to succeed.

Note 2: At least one of the tags Service and LogicalNetworkLink must be present. If both Service and LogicalNetworkLink tags are present, the Zone is a composite NAT and LBL zone.

Change Network Zone command

A Network zone provisioned within the CS 2000 management system can be modified such that a new parent network zone, Media Proxy group, or VPN name is selected.

This is a subset of the functionality supported by the interface and available to the PC platform. Attempts to modify parameters other than the parent network zone will either be ignored - or will result in an error being returned. An error indicates that the change request would result in a mismatch between the CS 2000 management platform and the PC.

Two examples of a Change Network Zone request are shown, the first is valid for NAT, LBL and Composite type network zones, the second is only valid for LBL and Composite type network zones (if issued against a NAT it implies an unsupported type change from NAT to Composite). Both examples perform the same task as only the parent network zone change is of interest.

An example of Change Network Zone (Any Network Zone type) request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<changeNetworkZone usn="1" version="1.0">
<Parameters>
<Name>router1</Name>
<PreferredMPGroup>MPG2</PreferredMPGroup>
<Parent>
<ParentID>247</ParentID>
</Parent>
</Parameters>
</changeNetworkZone>
</Methods>
</Command>
</CommandList>
```

An example of Change Network Zone (LBL or Composite) request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<changeNetworkZone usn="1" version="1.0">
<Parameters>
<Name>router1</Name>
<Parent>
<ParentID>247</ParentID>
</Parent>
<IntraZoneBWInfo>
<BWCIR>362</BWCIR>
<NNSC>
<Premium>12</Premium>
<Platinum>23</Platinum>
<Gold>25</Gold>
<Silver>3</Silver>
<Bronze>5</Bronze>
</NNSC>
</IntraZoneBWInfo>
<LogicalNetworkLink>
<BandWidthInfo>
```

```
<Symmetry>false</Symmetry>
<UpBWInfo>
<BWCIR>456</BWCIR>
</UpBWInfo>
<DownBWInfo>
<BWCIR>997</BWCIR>
</DownBWInfo>
<LayerInfo>
<Layer>
<LinkLayerType>IP</LinkLayerType>
</Layer>
<Layer>
<Overhead>34</Overhead>
</Layer>
<Layer>
<Overhead>16</Overhead>
<UnitSize>53</UnitSize>
<UnitOverhead>5</UnitOverHead>
</Layer>
<Layer>
<LinkLayerType>Sonet</LinkLayerType>
</Layer>
</LayerInfo>
</LogicalNetworkLink>
</Parameters>
</changeNetworkZone>
</Methods>
</Command>
</CommandList>
```

An example of Change Network Zone response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<changeNetworkZone usn="1" version="1.0">
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</changeNetworkZone>
</Methods>
</Response>
</CommandList>
```

Delete Network Zone command

Delete network zone removes a provisioned network zone from the CS 2000 management system. The network zone to be removed may be identified by either its name or ID.

An example of Delete Network Zone request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<deleteNetworkZone usn="1" version="1.0">
<Parameters>
<ID>352</ID>
</Parameters>
</deleteNetworkZone>
</Methods>
</Command>
</CommandList>
```

An example of Delete Network Zone response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<deleteNetworkZone usn="1" version="1.0">
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</deleteNetworkZone>
</Methods>
</Response>
</CommandList>
```

Query Network Zone command

Query network zone provides a mechanism to retrieve a single network zone, all network zones or a range of network zones restricted by ID filters.

An example of Query Network Zone request (All zones)

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryNetworkZone usn="1" version="1.0">
<Parameters>
</Parameters>
</queryNetworkZone>
</Methods>
</Command>
</CommandList>
```

An example of Query Network Zone request (range of zones)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryNetworkZone usn="1" version="1.0">
<Parameters>
<IDMin>500</IDMin>
<MaxZones>200</MaxZones>
</Parameters>
</queryNetworkZone>
</Methods>
</Command>
</CommandList>
```

An example of Query Network Zone request (specific zone)

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryNetworkZone usn="1" version="1.0">
<Parameters>
<Name>router12</Name>
</Parameters>
</queryNetworkZone>
</Methods>
</Command>
</CommandList>
```

An example of Query Network Zone request (specific Media Proxy group)

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<queryNetworkZone usn="1" version="1.0">
<Parameters>
<PreferredMPGroup>MPG1</PreferredMPGroup>
</Parameters>
</queryNetworkZone>
</Methods>
</Command>
</CommandList>
```

An example of Query Network Zone (range/all zones) response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
```

```
<queryNetworkZone usn="1" version="1.0">
<ReturnData>
<NetworkZone>
<ID>235</ID>
<Name>NZ1</Name>
<ParentID>0</ParentID>
</NetworkZone>
<NetworkZone>
<ID>236</ID>
<Name>Peter8</Name>
<ParentID>0</ParentID>
</NetworkZone>
<NetworkZone>
<ID>566</ID>
<Name>nat2</Name>
<ParentID>65546</ParentID>
</NetworkZone>
<NetworkZone>
<ID>891</ID>
<Name>NZ2</Name>
<ParentID>235</ParentID>
</NetworkZone>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</queryNetworkZone>
</Methods>
</Response>
</CommandList>
```

An example of Query Network Zone (individual Composite network zone) response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryNetworkZone usn="1" version="1.0">
<ReturnData>
<ID>236</ID>
<Name>Peter8</Name>
<Service>NAT</Service>
<Parent>
<ParentID>0</ParentID>
</Parent>
<LogicalNetworkLink>
<LayerInfo>
<Layer>
<LinkLayerType>UNKNOWN</LinkLayerType>
</Layer>
</LayerInfo>
```

```
</LogicalNetworkLink>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</queryNetworkZone>
</Methods>
</Response>
</CommandList>
```

Add VPN command

Add VPN is a new command in SN09. The command adds the definition of a new VPN.

An example of Add VPN request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ITransIf</Interface>
<Methods>
<addVPN usn="1" version="1.0">
<Parameters>
<vpnName>VPN11</vpnName>
</Parameters>
</addVPN>
</Methods>
</Command>
</CommandList>
```

An example of Add VPN response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<addVPN usn="1" version="1.0">
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</addVPN>
</Methods>
</Response>
</CommandList>
```

Delete VPN command

Delete VPN is a new command in SN09. The command deletes the definition of a new VPN.

An example of Delete VPN request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
```

```
<Interface>ITransIf</Interface>
<Methods>
<deleteVPN usn="1" version="1.0">
<Parameters>
<vpnName>VPN11</vpnName>
</Parameters>
</deleteVPN>
</Methods>
</Command>
</CommandList>
```

An example of Add VPN response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<deleteVPN usn="1" version="1.0">
<ReturnData>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</deleteVPN>
</Methods>
</Response>
</CommandList>
```

Internet Transparency unsuccessful response messages

An example of Internet transparency error response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>ITransIf</Interface>
<Methods>
<queryNAT usn="1" version="1.0">
<ReturnData>
<NATname>nat1</NATname>
<ReturnCode value ="0" text="Successful result"/>
</ReturnData>
</queryNAT>
</Methods>
</Response>
</CommandList>
```

Limations and restrictions

- assocMG (Associate Media Gateway)
  - — The profile applied to the MG must have a service type of ITRANS when a NAT or LBL middle box is specified.

- — The profile applied to the MG must have a service type of ITRANS_ROAM when root middleboxes are specified.
- — The GWC specified must have the Internet Transparency capability (ability to handle NAT/LBL and MPs) when a NAT or LBL middle box is specified.
- — The GWC specified cannot have more than 2000 middleboxes associated.
- — The NAT, LBL, or Root Middleboxes names passed to assocMG must exist.
- — ’outtsp’ cannot be specified as a Root Middlebox.
- — The Pep server name tag cannot be used in a XML query containing the itransMiddlebox name tag.
- — The PepServer name tag or the itransMiddlebox name tag cannot be used in a XML query containing the rootMiddlebox name tags.

- queryMP (Query Media Proxy)
  - — the MP name and GWC name must not be specified together in the same query.

- queryNAT
  - — the NAT name and GWC name must not be specified together in the same query

- changeAssocMB (Change NAT/LBL associated to a Media Gateway)
  - — the specified MG must be defined
  - — the specified NAT/LBL middlebox must be defined.

- changeRootMiddleboxes (Change the root Middleboxes associated to a Media Gateway)
  - — the specified MG must be defined
  - — the specified root middleboxes must be defined
  - — ’outtsp’ cannot be a specified middlebox
  - — the specified MG must be defined

- addLBL
  - — ’outtsp’ cannot be specified as the parent of an LBL

- addNAT
  - — ’outtsp’ cannot be specified as the parent of a NAT

- addNetworkZone
  - specification of network zone ID requires Call Agent ID to be setup
  - Network zones should not be added without the presence of the PC platform and external VCAC counting being enabled.
- changeNetworkZone
  - changing network zone name is unsupported.
  - changing LogicalNetworkLink BandwidthInfo for a network zone of type NAT is unsupported (implies change of type from NAT to Composite).
  - deleting LogicalNetworkLink for a network zone is unsupported as implies potential change of type.
  - Removal of network zone parent must be done by setting ParentID = 0. Cannot perform using an empty ParentName tag.
- queryNetworkZone
  - LBL / Composite type network zone indicated by LogicalNetworkLink populated by dummy data.
- addLinkLayerType
  - Interface unsupported by CS 2000 management system.
- changeLinkLayerType
  - Interface unsupported by CS 2000 management system
- deleteLinkLayerType
  - Interface unsupported by CS 2000 management system
- queryLinkLayerType
  - Interface unsupported by CS 2000 management system

Policy Controller (PC) provisioning

The Policy Controller (PC) provide capability to apply policies during call-processing, such as policy-based codec selection, connection admission control etc. In general the policy may be implemented on the PC itself.

Beginning in SN08, the PC will support exactly one policy, that of Network Resource Reservation, using Virtual Connection Admission Control (VCAC) as the enforcement mechanism. Hence counting of available resource across Limited Bandwidth Links (LBLs) and connection admission decisions will be carried out on the PC, instead of being done on the GWCs as in SN07. The GWCs communicate with the PC to determine whether a call can be set-up. This will allow LBLs to be shared across different GWCs.

For provisioning, new XML interfaces are added for use in network installations where the CS2M platform is required to operate in conjunction with an PC. These interfaces align with the provisioning model used by the PC for NAT and VCAC functionality and is described in terms of Network Zones and Zone characteristics. The XML interfaces are identical to those supported by the PC.

For XML commands supported from PC interface refer Policy Controller Configuration Management NTP NN10432-511. In order to provision any entity, the information must first be provisioned using OSSGate node provisioning and then, if successful, the same information must be provisioned into the PC. Node provisioning via OSSGate should be done first because all constraints (e.g. whether the zone is currently associated with a gateway) are verified. Thus if any constraints fail, the provisioning request will fail providing the proper response and will avoid PC provisioning failures. It is important that the information is provisioned in the same order into each system. In particular, adding/deleting NATs & LBLs should be carried out identically in both systems. For example this could be achieved by adding an LBL to OSSGate node provisioning and then to PC, followed by adding the next LBL to OSSGate node provisioning and then to the PC etc.

The (I)SN07 ITRANS VCAC interfaces for NAT and LBL provisioning are still available. However, when a PC is deployed in the network configuration, the PC consistent provisioning model is provided via use of the zone provisioning interfaces and the (I)SN07 ITRANS VCAC interfaces for NAT and LBL provisioning are not required. A mapping is available in table below.

> ***Note:*** When adding a Network Zone, Resource Usage Factor and Max Count values are not required as long as Network VCAC is on. However if Network VCAC is ‘off’, resource usage factor and max count details ARE required and hence the XML command ‘addLBL’ should be used to add LBL Network Zones. However it is not recommend to switch Network VCAC off once it has been switched on. Also, if it is switched off, basic VCAC will only work correctly if the topology still meets the basic VCAC/’internal counting’ topology constraints.

Add, delete, change of PC are done from the CS 2000 Management Tool GUI. For details refer GWC Configuration NTP NN10112-511.pdf

Limitations and restrictions

- Only one PC per CS 2000 is supported

- Topology data transfer from CS2M node provisioning to PC when PC is commissioned has to be done manually using existing GUI/OSS interfaces.
- Once Network VCAC is switched on, CS2M node provisioning will remove any topology provisioning constraint checks which apply to Basic VCAC (internal counting). If Network VCAC is subsequently switched off, the internal counting checks will NOT be restored. It is not recommended to switch Network VCAC off once it has been switched on. Also, if it is switched off, basic VCAC will only work correctly if the topology still meets the basic VCAC/'internal counting' topology constraints. In normal use it is expected that Network VCAC will be switched on as soon as the upgrade and soak process is complete and PC commissioned and will not be switched off.

Application Layer Gateway (ALG) provisioning

Application Layer gateways are logical interfaces that provide a way for network operators to separate the Cable hosted network IP address space from the service provider operator IP address space. It will allow the connection of two different IP address space domains without having to understand the exact IP address within the other domain.

ALGs are NATs that understand the application (in this case, NCS protocol) and enable the application to continue to work even though it is “behind” the NAT. The ALG provides a single IP address interface to the CS 2000 via a GWC while interfacing to multiple small gateways each with its own unique IP address. The IP address of the gateways which subtend the ALG are not known to the CS 2000 and thus will be set to 0.0.0.0 during provisioning. The ALG is positioned in the path of packets flowing from media gateways and media gateway controllers. In this level the ALG provides a level of packet translation without directly participating in the media flow, that is to say the ALG does not directly establish media path like a gateway, instead it serves as a proxy for the media through a request from the Call Server.

The OSS interface supports AssocMG, queryMG, ChangeMGGWCEMData, GetALGData and GetAlgNamesByGWCid.

AssocMG supports associating an ALG to a GW. The queryMG returns the associated ALG. The ChangeMGGWCEMData operation helps to change the ALG associated to a gateway to a different one. GetALGData returns the provisioning parameters associated with an ALG. GetAlgNamesByGWCid returns all ALG’s provisioned on the specific GWC

Description of Method Parameters

Method parameters are defined below. Input data is mandatory except where indicated otherwise.

Step **Action**

1 AssocMG - Method to associate a gateway to the CS 2000. This method has the following NEW parameters: (only NEW ones described, see standard documentation for the other definitions).

a. Input data

    i. usn

    ii. version

    iii. algName (Optional, Cannot be included as well as pepServerName) - defines the ALG (optional) for this gateway

b. Output data

    i. usn (value should be the same as the input)

    ii. version (value should be the same as the input)

    iii. ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information

2 GetALGData - Method that retrieves. This method has the following parameters:

a. Input data

    i. usn

    ii. version

    iii. algName - defines the new ALG name

b. Output data

    i. usn (value should be the same as the input)

    ii. version (value should be the same as the input)

    iii. ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.

    iv. Returns algname, IP address, Port and Protocol for the given ALG name

3 ChangeMGGWCEMData- Method that changes the ALG for a given gateway. This method has the following parameters:

a. Input data

    i. usn

    ii. version

iii. MGname indicates the gateway that this change command affects

b. Output data

i. usn (value should be the same as the input)

ii. version (value should be the same as the input)

iii. ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information

4 QueryMG- Method that retrieves ALGs which are associated with input Gateway provided as parameter. This method has the following parameters

a. Input data

i. usn

ii. version

iii. MGname indicates the gateway that this change command affects

b. Output data

i. ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.

ii. Returns ALG middlebox name associated with MGname provided

5 GetAlgNamesByGWCid- Method that retrieves. This method has the following parameters:

a. Input data

i. usn

ii. version

iii. Gateway controller name where data will be searched

b. Output data

i. usn (value should be the same as the input)

ii. version (value should be the same as the input)

iii. ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.

iv. Provides all of the ALGs which are provisioned on the given Gateway Controller

—End—

An example of Associate MG with ALG request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<assocMG usn="1" version="1.0">
<Parameters>
<mgUIName>MGXML.5</mgUIName>
<mgProfileName>MOTOROLAMTA_2</mgProfileName>
<mgIpAddr>0.0.0.0</mgIpAddr>
<mgProtocolType>4</mgProtocolType>
<mgProtocolVersion>1.0</mgProtocolVersion>
<mgProtocolPort>2427</mgProtocolPort>
<gwcUIName>GWC-113</gwcUIName>
<reservedTerminations>2</reservedTerminations>
<algName>ALG</algName>
</Parameters>
</assocMG>
</Methods>
</Command>
</CommandList>
```

An example of queryMG with ALG response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<queryMG usn="1"  version="1.0">
<ReturnData>
<Row>
<mgUIList>billrgw.1</mgUIList>
<mgIpList>0.0.0.0</mgIpList>
<callServerId>RTP7</callServerId>
<gwcUIName>GWC-113</gwcUIName>
<svcTypeList>0</svcTypeList>
<nodeName>C411 00 1</nodeName>
<engrEndPoints>2</engrEndPoints>
<protType>ncsprotocol</protType>
<protVersion>1.0</protVersion>
<protPort>2427</protPort>
<profileName>MOTOROLAMTA_2</profileName>
<algName>ALG1</algName>
</Row>
<RC>0</RC>
```

```
<MsgTxt>Query of a Single MG was successful</MsgTxt>
</ReturnData>
</queryMG>
</Methods>
</Response>
</CommandList>
```

An example of Change Associated ALG with an MG request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ALGOSSIf<Interface>
<Methods>
<changeMGGWCEMData usn="1" version="1.0">
<Parameters>
<MGUIName>MG1</MGUIName>
<algName>billr.alg1</algName>
</Parameters>
</changeMGGWCEMData>
</Methods>
</Command>
</CommandList>
```

An example of Change Associated ALG with an MG response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>ALGOSSIf</Interface>
<Methods>
<changeMGGWCEMData usn="135"  version="1.0">
<ReturnCode value="0" text="Successful result"/>
<MsgTxt>MG ALG association changed successfully</MsgTxt>
</changeMGGWCEMData>
</Methods>
</Response>
</CommandList>
```

An example of GetALG data request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ALGOSSIf<Interface>
<Methods>
<getALGData usn="1" version="1.0">
<Parameters>
<algName>billr.alg1</algName>
</Parameters>
</getALGData>
</Methods>
</Command>
```

```
</CommandList>
```

An example of GetALG data response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>ALGOSSIf</Interface>
<Methods>
<getALGData usn="123"  version="1.0">
<ReturnCode value="0" text="Successful result"/>
<MsgTxt>Data Available</MsgTxt>
<algName>billr.alg1</algName>
<ServerType>DQoS</ServerType>
<IPAddress>47.192.234.88</IPAddress>
<Port>2427></Port>
</getALGData>
<getALGData usn="124" version="1.0>
<ReturnCode Value="0" text="Successful result"/>
<MsgTxt>Data Available</MsgTxt>
<algName>billr.alg1</algName>
<ServerType>DQoS</ServerType>
<IPAddress>47.192.234.100</IPAddress>
<Port>2427</Port>
</getALGData>
</Methods>
</Response>
</CommandList>
```

An example of Query GWC to ALG Connections request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>ALGOSSIf<Interface>
<Methods>
<getAlgNamesbyGWCId usn="1" version="1.0">
<Parameters>
<GWC>GWC-1</GWC>
</Parameters>
</getAlgNamesbyGWCId>
<getAlgNamesbyGWCId usn="1" version ="1.0">
<Parameters>
<GWC>GWC-5</GWC>
</getAlgNamesbyGWCId>
</Methods>
</Command>
</CommandList>
```

An example of GetALG data response

```
<?xml version="1.0" encoding="UTF-8">
<CommandList>
```

```
<Response>
<Interface>ALGOSSIf</Interface>
<Methods>
<getAlgNamesbyGWCId usn="1"  version="1.0">
<ReturnCode value="0" text="Successful result"/>
<MsgTxt>Data Available</MsgTxt>
<algName>billr.alg1</algName>
<algName>billr.alg2</algName>
<algName>billr.alg3</algName>
<algName>billr.alg4</algName>
<algName>billr.alg5</algName>
</getAlgNamesbyGWCId>
<getAlgNamesbyGWCId usn="2" version="1.0>
<ReturnCode Value="0" text="Successful result"/>
<MsgTxt>Data Available</MsgTxt>
<algName>billr.alg6</algName>
<algName>billr.alg7</algName>
<algName>billr.alg8</algName>
</getAlgNamesbyGWCId>
</Methods>
</Response>
</CommandList>
```

Limitations and Restrictions

- The CS 2000 has a limit of 2000 “middleboxes” which is the number of ALGs, PEP servers, NATs and media Proxies combined.
- The CS 2000 has a limit of 200 ALG middlebox.
- A single GWC on the CS 2000 has a limit of 20 ALG middleboxes.
- ALGs can only be provision against MTA gateways.
- ALG only supports the NCS protocol.
- If the ALG does not transmit Audit Endpoint (AUEP) messages to the MTA before the MTA sends its first message (typically an RSIP) then the lines in the CS 2000 will appear SYSB because the Audit Endpoint is rejected or not responded to. This could happen in the event that the ALG loses all state information and requires the first MTA originated message to re-establish its bindings. To correct this problem, the MTA needs to send any message, e.g. an RSIP or an off hook and this will re-establish the binding and clear the SYSB state on the line. This problem will not occur if the ALG vendor supports discovery through the use of DNS of the MTA IP Address before the MTA sends its first message.
- DQoS is not supported on an MTA behind an ALG. Nortel is not aware of any ALG on the market that supports DQOS.

- IPSEC is not supported since a NAT device will cause end to end (GWC to MTA) security to fail because of the IP address and message content manipulation done by the ALG.
- If a Redirecting Media Gateway Controller (RMGC) is configured, the DNS servers must be configured correctly. For example the ALG upon receiving the response from the RMGC on the CS 2000 must be able allow the gateway to resolve the Fully Qualified Domain Name(FQDN) in the message through DNS queries.
- DNS is not supported by the CS 2000 to discover the ALG IP address. We only support static configuration of the IP address of the ALG since it is not expected that there will be many ALGs.
- When an ALG is associated to a gateway a DQOS PEP Servers, Low Band Width(LBL), IP-VPN Nats will not be allowed on the same gateway.
- The IP address of the gateway being associated with an ALG will not be modified. If the IP address field is left blank, the default value of 0.0.0.0 will be provision against the gateway.
- CS 2000 QoS collector will collect statistics as reported by the ALG. There is no separation of statistics between the ALG boundaries.

Authorization for ALG commands

The security for the GUI Client and OSSgate interface includes support for permission or ‘Authorization’ levels for commands. Each command is associated with one or more user groups. In order to execute a command, a user must belong to at least one of the associated user groups. The user groups associated with the new /modified commands associated with this feature are specified in the table below.

Command	**User Group**				
	mgcadm	**mgciprov**	**mgcmtc**	**mgcsprov**	**mgcro**
addALG	X	X			
deleteALG	X	X			
changeALG	X	X			
AssocMG	X	X			
queryMG	X	X	X	X	X
GetALGData	X	X	X	X	X
changeMGGWCEMData	X	X			
getAlgNamesByGWCid	X	X	X	X	X

ALG return codes

Return code	Meaning
0	Successful operation
2	Associate MG operation invalid input from client interface
3	Incorrect command version
4	The GWC has no ALG connections
11	Gateway does not exist in Database

User authorization for Internet Transparency operations

The security for the OSSGate interface includes support for permission or ‘Authorization’ levels for commands. Each operation is associated with one or more authorization groups. To execute a command, a user must belong to at least one of the associated authorization groups. The authorization groups associated with each Internet Transparency operation is listed in the table below. (For assocMG and queryMG authorization levels, see Nodes Provisioning permissions).

Command	User Group				
	mgcadm	mgcrw	mgcmtc	mgcsprov	mgcro
queryCallAgentId	X	X	X	X	X
setCallAgentId	X	X			
addNAT	X	X			
deleteNAT	X	X			
changeNAT	X	X			
queryNAT	X	X	X	X	X
addLBL	X	X			
deleteLBL	X	X			
changeLBL	X	X			
queryLBL	X	X	X	X	X
addRU	X	X			
deleteRU	X	X			
changeRU	X	X			
queryRU	X	X	X	X	X
addMP	X	X			
deleteMP	X	X			
changeMP	X	X			
queryMP	X	X	X	X	X

Command	User Group				
	mgcadm	mgcrw	mgcmtc	mgcsprov	mgcro
changeAssocMB	X	X			
changeRootMiddleboxes	X	X			
addNetworkZone	X	X			
changeNetworkZone	X	X			
deleteNetworkZone	X	X			
queryNetworkZone	X	X	X	X	X

Internet Transparency return codes

Return code	Meaning
0	**Successful operation**
16	Associate MG Operation invalid input from client interface. Example: A service type of ITRANS is required.
21	Associate MG Operation failed to associate the MG with a GWC. Example: The number of distinct NAT's used exceeds the limit of 400 per GWC.
301	Incorrect command version in Internet Transparency change/query request
302	Platform applications internal error: OSS Interface applications cannot connect to required server application
303	Error in the input data
304	GWC-Element Manger error during ITRANS operation
305	No Data available. Example: No NAT/MP found matching this query or query of a non-existing GWC>
306	PC Interface Unsupported By Platform Interface is valid according to XSD but unsupported by the CS 2000 management system. Returned in response to addLinkLayerType, changeLinkLayerType, deleteLinkLayerType and queryLinkLayerType interfaces being called.
501	User is not a member of any of the user groups allowed to perform this action.

H.323 limitations and restrictions

The following are current restrictions with respect to H.323 gateway provisioning.

Change gateway operation

The following modifications are not allowed:

- Changing H.323 version. (This is set to "4.0".)

- Changing protocol from H.323 to another protocol.
- Changing from another protocol to H.323.

Change gateway controller operation

GWCs can not be changed from being non-H.323 to being H.323. Likewise, they can not be changed from H.323 to non-H.323.

Querying Gateway profiles

Starting in (I)SN08, OSSGate supports query functionality to enable the OSS to query aspects of a GW profiles (profile certificates). The following commands are supported:

- Get Certificate List - This command returns all the valid profiles names that is currently known to the CS 2000 management tool system. Each profile name is listed tag-limited with <certificate>
- Get Certificate Data - This command returns the details pertaining to the provisioned profile. The user inputs in the the name of the Profile Name with which the certificate was initially created.

An example of Get Certificate List request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>CertificateOSSIf</Interface>
<Methods>
<getCertificateList usn="1" version="1.0">
<Parameters>
</Parameters>
</getCertificateList>
</Methods>
</Command>
</CommandList>
```

An example of Get Certificate List response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>CertificateOSSIf</Interface>
<Methods>
<getCertificateList usn="1"  version="1.0">
<ReturnCode value="13.0" text="Successful result"/>
<certificate>ARRIS_TOUCHTONE_NN01_4</certificate>
<certificate>ARRIS_TOUCHTONE_NN02_4</certificate>
<certificate>ASKEY_LINE_GW_12</certificate>
<certificate>ASKEY_LINE_GW_30</certificate>
<certificate>ASKEY_LINE_GW_4</certificate>
<certificate>AUDIOCODES</certificate>
<certificate>CISCO_2600</certificate>
```

```
<certificate>CISCO_AS5300</certificate>
<certificate>CVX1800_2688</certificate>
<certificate>DOMAIN_NAME</certificate>
<certificate>MEDIATRIX_LINE_GWC_24</certificate>
<certificate>MEDIATRIX_LINE_GWC_4</certificate>
<certificate>MGCP_LINE_GW_1</certificate>
<certificate>MOTOROLAMTA_1</certificate>
<certificate>MOTOROLAMTA_2</certificate>
<certificate>MOTOROLAMTA_4</certificate>
<certificate>NORTEL_BCM</certificate>
<certificate>NUERA_GX_ASPEN</certificate>
<certificate>NUERA_GX_MEGACO</certificate>
<certificate>PVG15K_1000_ASPEN</certificate>
<certificate>PVG15K_1000_MEGACO</certificate>
<certificate>PVG15K_ASPEN</certificate>
<certificate>PVG15K_MEGACO</certificate>
<certificate>PVG15K_PARTIAL_ASPEN</certificate>
<certificate>PVG15K_PARTIAL_MEGACO</certificate>
<certificate>AMBIT_LINE_GW_16</certificate>
<certificate>PVG7K_ASPEN</certificate>
<certificate>PVG7K_MEGACO</certificate>
<certificate>PVG_APG_ASPEN</certificate>
<certificate>PVG_APG_MEGACO</certificate>
<certificate>PVG_APG_VSP3_ASPEN</certificate>
<certificate>PVG_APG_VSP3_MEGACO</certificate>
<certificate>PVG_VSP3_ASPEN</certificate>
<certificate>PVG_VSP3_MEGACO</certificate>
<certificate>(I)SN06_CICM</certificate>
<certificate>SUCCESSION_10000</certificate>
<certificate>TGCP</certificate>
<certificate>TOUCHTONE_NN01_1</certificate>
<certificate>TOUCHTONE_NN01_2</certificate>
<certificate>TOUCHTONE_NN01_3</certificate>
<certificate>TOUCHTONE_NN01_4</certificate>
<certificate>UAS</certificate>
<certificate>UE9000MG</certificate>
<certificate>UE9000MG_ABI</certificate>
<certificate>UE9000MG_ABI_IP</certificate>
<certificate>UE9000MG_IP</certificate>
<certificate>WESTELL</certificate>
<getCertificateList>
</Methods>
</Response>
</CommandList>
```

An example of Get Certificate Data request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>CertificateOSSIf</Interface>
```

```
<Methods>
<getCertificateData usn="1" version="1.0">
<Parameters>
<CertificateName>CISCO_2600</CertificateName>
</Parameters>
</getCertificateData>
</Methods>
</Command>
</CommandList>
```

An example of Get Certificate Data response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>CertificateOSSIf</Interface>
<Methods>
<getCertificateList usn="1"  version="1.0">
<ReturnCode value="13.0" text="Successful result"/>
<certificate>
<MaxEndPoints>1032</MaxEndPoints>
<Category>LARGE</Category>
<EndpointType>nocarrierTrkGwcNN</EndpointType>
<GenerateLGRP>true</GenerateLGRP>
<ResvTermMandatory>true</ResvTermMandatory>
<ChangeIPAvailable>true</ChangeIPAvailable>
<DispPhyLocation>false</DispPhyLocation>
<InventoryType>H.323 Gateway</InventoryType>
<InventoryRole>Media Gateway</Inventory Role>
<Supported Protocol>
<protocolName>h323</protocolName>
<version>4.0</version>
</Supported Protocol>
<GWCProfileNumber>63</GWCProfileNumber>
<EPIDGenDesc>auto1</EPIDGenDesc>
<ServiceTypeList>H323</ServiceTypeList>
<ServiceTypeList>ITRANS</ServiceTypeList>
<CompatibleGWProfileList>CISCO_2600</
CompatibleGWProfileList>
<GatewayNameFormatList>
<nameFormat formatKey="GATEWAY.H323"
enabled="true" applyToPattern=".*">
<name>H323 Gateway</name>
<delimiters>‘</delimiters>
<maxTokens>1</maxTokens>
<tokenFormats>
<tokenFormat tokenKey="GW">
<name>Name</name>
<pattern><!CDATA[[^"‘’<>=]*]]></pattern>
</tokenFormat>
</tokenFormats>
```

```
</nameFormat>
<nameFormat formatKey="GATEWAY.ALL"
enabled="true" applyToPattern".*">
<name>Generic For All Gateways</name>
<template>gatewayName</template>
<maxLength>32</maxLength>
<minLength>1</minLength>
<delimiters>none</delimeters>
<tokenFormats>
<tokenFormat tokenKey="GW">
<pattern><![CDATA[[^`'<>=,;]*]]></pattern>
<name>Name</name>
</tokenFormat>
</tokenFormats>
</nameFormat>
</GatewayNameFormatList>
<EndpointNameFormatList>
<nameFormat formatKey="CARRIER.H323.EPG"
enabled="true" applyToPatter=".*">
<name>EPG</name>
<template><!CDATA[EPG_<nnn>]]></template>
<maxTokens>2</maxTokens>
<minTokens>2</minTokens>
<delimeters>_</delimeters>
<maxLength>7</maxLength>
<minLength>7</minLength>
<appDataList>
<appData name="CARRIERTYPEORPROTOCOLALLOWED"
value="H323" type="NONE" />
<appData name="CAPACITYDEFAULT"
value="32" type="INTL" />
<appData name="CAPACITYDEFAULT"
value="24" type="NA" />
</appDataList>
<tokenFormats>
<tokenFormat tokenKey="TYPE">
<name>Type</name>
<template>EPG</template>
<type>alphanumeric</type>
<template><!CDATA[<nnn>]]></template>
</tokenFormat>
</tokenFormats>
</nameFormat>
</EndpointNameFormatList>
</certificate>
</getCertificateData>
</Methods>
</Response>
</CommandList>
```

Provisioning third party large line gateways and endpoints

Third party large line gateway provisioning is supported from (I)SN08 from OSSGate. To provision these gateways the gateway profiles need to be known to the CS 2000 management tool system. Refer to details in *GWC Configuration* , NN10112-511, for new profile addition. A typical sequence to add a third party gateway, provision configuration data and bring the system to service includes the following steps:

- Provision table SITE in CS 2000 Core. SITE values for third party gateways should be restricted for use for this type of gateways.
- Provision table PADDATA in CS 2000 Core (PKLNL and PKNIL entries).
- Add the gateway profile certificate to the CS 2000 management tool system.
- Provision the third party gateway using CS 2000 management tool or OSSGate node provisioning interface
- Pre-provision the GW end-points from OSSGATE or BPT.
- Execute SERVORD+ commands via OSSGate when service is assigned to LENs in LNINV.

Associating Third party gateways to GWCs

Existing AssociateMG XML interface supports associating third party gateways via OSSGate. Additional XML elements that are applicable to the associateMG operation are frameNumber, unitNumber,frameType, floorPosition, rowPosition, framePosition, unitPosition,slotpostion. This information enables the Gateway and End-point Name mapping to Logical Group and Line Equipment Number.

Endpoint name mapping to Logical Group and Line Equipment Number (LEN)

The logical group name is derived from the Gateway SITE, the frame number and the unit number. The end-point name is user defined and its associated LEN has absolutely no bearing on the end-point name. The LEN is derived from the logical group name and the user specified circuit number.

Example

```
Gateway <GWNAM>, for example "MyGW"
Logical Group <SITE> <FFF> <U>, for example "CARY 142 0"
where
<SITE> = SITE name chosen at gateway provisioning time.
Matching gateway name to SITE name is not an enforced
rule, however it is suggested that a third party GW SITE
```

```
name be defined in the CS 2000 Core and that all third
party gateway provisioning utilize the “third party GW”
SITE name.  <FFF> = Frame number that is specified during
the third party GW provisioning.  This number can range
between 0 - 511 <U> = Unit number that is specified during
the third party GW provisioning.  This number can range
from 0 - 9.
End-point
The end-point name is flexible.  The format of the name is
defined during the GW profile creation time and enforced
during the GW provisioning time.  The name is completely
user defined.  The name can include any physical location
data of the hardware.  The end-point name can be up to 32
characters long with no spaces in between.
LEN Format
<SITE> <FFF> <U> <TT> <tt>, for example "CARY 142 0 10 13"
where
<SITE>, <FFF>, <U>, <TT>, and <tt> are as defined above.
```

The following summarizes the correlation between a GW, its end-points and the associated LENs:

- One third party "Gateway" mapped to one logical group in the CS 2000 Core.
- 1023 terminations are supported per logical group.
- 1 terminal number is reserved in the CS 2000 Core for maintenance/messaging.

GW Name (User Input)	Profile	LGRP Name	Endpoint (User Input)	Circuit Number (User Input)	LEN
MyGW.ext.net	PRF_1	CARY 100 5	ext/100/5/1/1	0	EXT 100 5 00 00
MyGW.ext.net	PRF1	CARY 100 5	ext/100/5/1/2	1	EXT 100 5 00 01
..	..	..	..	..	..
..	..	..	..	..	..
MyGW2.ext.net	PRF_2	EXT2 200 8	ext2-ep-last	1022	EXT2 200 8 10 22

Terminal number 0 is reserved in the CS 2000 Core for the maintenance/messaging terminal on the node.

LEN is generated using the following formula:

(LGRPName) (CN/100) (CN%100) where CN => Circuit Number

The last 2 parts of LEN are 0 padded to keep the value to 2 digits similar to the other LENs.

Terminology and description of method parameters

Nodes provisioning with OSSGate details the various parameters needed for Associate MG operation. This section details the additional parameters needed for AssociateMG operation to successfully provision a third party gateway.

frameNumber

Frame number. The range of this value is 0 - 511. TYPE unsignedShort.

unitNumber

Unit/Shelf number. The range of this value is 0 - 9. TYPE unsignedShort.

frameType

Frame Type. These types are pre-defined as the data type FRAME_NAME in the Core. TYPE string.

Values in the following list are valid: {TME,DCE,RSE,N6E,CCC,MTC,NET,DNI,PDC,SLC,MIS,MSS,MEX, LGE,DTE,LTE,LCE,MS6E,MS7E,ST6E,SPR,RTR,ST7E,PCSM,IOE, RTS,RUB,SME,RCE,RLCM,IDTE,DME,OPE,RCME,SCE,DLE,DNPC, PCMM,PCPM,DICP,IAE,ISTE,CMDC,DPCC,MSSC,MSDC,TAE,RUC, MCCM,MCNM,MCGM,MCLM,MCTM,MCPM,MCAM,MCSM,MCMM, MCDM,MCRM,MCHM,MCBM,LIM,LRE,VSRE,ILGE,LTEI,PCLM, PCTM,PCAM,TACE,THE,LCEI,MCDR,MC7M,MCEP,MCEM,MCEX, MCAP,MCAA,RCEI,MCOR,MLNK,MC6M,MNET,IPEC,SCC,MCRMI, MCLME,MCLMI,MCTMI,MCAM2,EMC,DTEI,CRSC,CEXT,PMTC, PMPM,ICGM,CTME,CIPE,CIDE,CMS7,CIOE,CDSN,HSI,MCRMS, ISME,CLCE,CMTA,CMVI,MVIE,CGPP,LGCI,SMUI,SPME,RSCM, CDTO,TRLE,LGEI,UR256,UR512,UR1152,VLCE,CDTE,CIDT,CLTE, CRLC,CISM,CPDT,CLMI,CPDC,CMIS,CMSS,CSDM,CCPE,MTAE, OPM256,SCPM,CSME,MCGS,UEE,RMME,RSCE,SMGE,HUB,MCNI, MIEC,CDT2,DTEO,MVDD,MG9F,OTHR,DDRM}

floorPosition

Floor position. The range of this value is 0 - 99. TYPE unsignedShort.

rowPosition

Row position. This value is pre-defined in the core. TYPE string.

Values in the following list are valid: {A,B,C,D,E,F,G,H,J,K,L,M,N,P,Q,R,S,T,U,V,W,X,Y,Z,AA,BB,CC,DD, EE,FF,GG,HH,JJ,KK,LL,MM,NN,PP,QQ,RR,SS,TT,UU,VV,WW,XX, YY,ZZ}

framePosition

Frame position. The range of this value is 0 - 99. TYPE unsignedShort.

unitPosition

Unit/Shelf position. The range of this value is 0 - 99. TYPE unsignedShort.

slotPosition

Slot position. TYPE unsignedShort

An example of assocMG request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<assocMG usn="1" version="1.0">
<Parameters>
<mgUIName>TPARTY.us.net</mgUIName>
<mgProfileName>TPARTY</mgProfileName>
<mgIpAddr>2.2.2.4</mgIpAddr>
<mgProtocolType>4</mgProtocolType>
<mgProtocolVersion>1.0</mgProtocolVersion>
<mgProtocolPort>2944</mgProtocolPort>
<mgSiteName>CARY</mgSiteName>
<gwcUIName>GWC-6</gwcUIName>
<reservedTerminations>1023</reservedTerminations>
<frameNumber>4</framNumber>
<unitNumber>6</unitNumber>
<frameType>OTHR</frameType>
<floorPosition>10</floorPosition>
<rowPosition>BB</rowPosition>
<framePosition>20</framePosition>
<unitPosition>30</unitPosition>
</Parameters>
</assocMG>
</Methods>
</Command>
</CommandList>
```

An example of Query MG response for a third party gateway

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
```

```
<Response>
<Interface>cs2kCfgMgrIf</Interface>
<Methods><queryMG usn="1"  version="1.0">
<ReturnData>
<Row>
<mgUIList>TPARTY.us.net</mgUIList>
<mgIpList>2.2.2.4</mgIpList>
<callServerId>COMPACT5</callServerId>
<gwcUIName>GWC-6</gwcUIName>
<svcTypeList>0</svcTypeList>
<nodeName>CARY 04 6</nodeName>
<pepServerName>NOT_SET</pepServerName>
<itransMiddleboxName>NOT_SET</itransMiddleboxName>
<algName>NOT_SET</alg_name>
<engrEndPoints>1023</engrEndPoints>
<protType>megaco</protType>
<protVersion>1.0</protVersion>
<protPort>2944</protPort>
<profileName>TPARTY</profileName>
<maxEngrEndPoints>1023</maxEngrEndPoints>
<rootMiddleboxName1/>
<rootMiddleboxName2/>
<rootMiddleboxName3/>
<rootMiddleboxName4/>
<rootMiddleboxName5/>
<frameNumber>4</frameNumber>
<unitNumber>6</unitNumber>
<frameType>OTHR</frameType>
<floorPos>10</floorPos>
<rowPos>BB</rowPos>
<framePos>20</framePos>
<unitPos>30</unitPos>
</Row>
<RC>0</RC>
<MsgTxt>Query of a Single MG was successful</MsgTxt>
</ReturnData>
</queryMG>
</Methods>
</Response>
</CommandList>
```

Endpoint provisioning

Third party gateway end-point pre-provisioning using XML is supported from OSSGATE interface. Following are the supported XML commands:

- addLineEndpoint
- delLineEndpoint
- queryLineEndpoint

- queryAllLineEndpoints

Terminology and description of method parameters

mgUIName

Fully qualified gateway name. This can be 1 - 32 characters long. TYPE string.

endpointName

Name of the end-point that is being provisioned. This can be 1 - 32 characters long. TYPE string.

circuitNumber

Number associated with the terminal. The range of this value is 0 - 1022. TYPE unsignedShort

Note: The circuitNumber equals terminalNumber -1. Therefore, <circuitNumber>1</circuitNumber> => Terminal Number = 2. A standard (although not required) practice is to express parameters endpointName and circuitNumber as <endpointName>aaln/1</endpoint> and <circuitNumber>0</circuitNumber>. In this case, aaln/1 maps to 0.

cardCode

String that defines the card code associated with the card. TYPE string. For International solutions, this parameter is required. For North American solutions, this is an optional parameter. The default is RDLSG.

Note: The cardCode parameter must be specified for the international Core.

padGroup

String that defines the pad group. TYPE string. Optional parameter. The default is PKLNL.

ground

String that defines the ground property. TYPE string. Values can be “N” or “Y”. Optional parameter. The default is “N”.

version

The version refers to the version of the method or operation - e.g., the AddSubscriber operation may have version 1.0 and 2.0, where version 1.0 takes parameters X and Y and version 2.0 takes parms X, Y, and Z. TYPE decimal.

Description of method parameters

Method parameters are defined below. Input data is mandatory except where indicated otherwise.

- addLineEndpoint - Interface/Method that adds a specified end-point to the SESM database and allocates a LEN to the corresponding terminal in CS 2000 Core.
    - Input data
        - usn
        - version
        - mgUIName
        - endpointName
        - circuitNumber
        - cardCode - Default: RDLSG
        - padGroup - Default: PKLNL
        - ground - Default: N
    - Output data
        - usn (value should be the same as the input)
        - version (value should be the same as the input)
        - RC - Return Code
        - MsgTxt - Text data detailing specific results of the operation in terms of success or failure of the operation and possible indication of failure reason in case of a failure.
- delLineEndpoint - Interface/Method that deletes a specified end-point from the SESM database and de-allocates LEN from the corresponding terminal in CS 2000 Core. Pre-condition: Service on the end-point must be un-assigned before attempting this operation.
    - Input data
        - usn
        - version
        - mgUIName
        - endpointName
OR
To delete mulitple endpoints with a specified range of circuit numbers
        - circuitNumberBegin

- circuitNumberEnd
  OR
  To delete all the endpoints on this GW
- ALL

— Output data

- usn (value should be the same as the input)
- version (value should be the same as the input)
- RC - Return Code
- MsgTxt - Text data detailing specific results of the operation in terms of success or failure of the operation and possible indication of failure reason in case of a failure.

- queryLineEndpoint - Interface/Method that queries a specific end-point data on this gateway. The end-point data will be enclosed within the <Row> tags similar to the output of the other nodes XML operations.
  - Input data
    - usn
    - version
    - mgUIName
    - endpointName
  - Output data
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - mgUIName
    - endpointName
    - CircuitNumber
    - cardCode
    - padGroup
    - status
    - ground
    - status
    - tid
    - LEN
    - RC - Return Code

  - MsgTxt - Text data detailing specific results of the operation in terms of success or failure of the operation and possible indication of failure reason in case of a failure.

- queryAllLineEndpoints - Interface/Method that queries all the end-points on this gateway. The end-point data for each of the end-points will be enclosed within the <Row> tags similar to the output of the other nodes XML operations.
  - Input data
    - usn
    - version
    - mgUIName
  - Output data
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - mgUIName
    - endpointName
    - CircuitNumber
    - cardCode
    - status
    - ground
    - status
    - tid
    - LEN
    - RC - Return Code
    - MsgTxt - Text data detailing specific results of the operation in terms of success or failure of the operation and possible indication of failure reason in case of a failure.

An example of assocMG request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<addLineEndpoint usn="1" version="1.0">
<Parameters>
<mgUIName>TPARTY.us.net</mgUIName>
```

```
<endpointName>aaln/1</endpointName>
<circuitNumber>1</circuitNumber>
</Parameters>
</addLineEndpoint>
</Methods>
</Command>
</CommandList>
```

An example of addLineEndpoint response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<addLineEndpoint usn="1" version="1.0">
<ReturnData>
<RC>0</RC>
<MsgTxt>The add line endpoint operation was successful:
Gateway = TPARTY.us.net, Endpoint= aaln/1.  </MsgTxt>
</ReturnData>
</addLineEndpoint>
</Methods>
</Response>
</CommandList>
```

An example of delLineEndpoint request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<delLineEndpoint usn="1" version="1.0">
<Parameters>
<mgUIName>TPARTY.us.net</mgUIName>
<endpointName>aaln/1</endpointName>
</Parameters>
</delLineEndpoint>
</Methods>
</Command>
</CommandList>
```

An example of delLineEndpoint response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<delLineEndpoint usn="1" version="1.0">
<ReturnData>
<RC>0</RC>
```

```
<MsgTxt>The delete line endpoint operation was successful:
Gateway = TPARTY.us.net, Endpoint= aaln/1.  </MsgTxt>
</ReturnData>
</delLineEndpoint>
</Methods>
</Response>
</CommandList>
```

An example request for deleting a range of endpoints with circuit numbers ranging from 1-4 with limits inclusive from a gateway

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<delLineEndpoint usn="1" version="1.0">
<Parameters>
<mgUIName>TPARTY.us.net</mgUIName>
<circuitNumberBegin>1</circuitNumberBegin>
<circuitNumberEnd>4</circuitNumberEnd>
</Parameters>
</delLineEndpoint>
</Methods>
</Command>
</CommandList>
```

An example of delLineEndpoint request that deletes ALL endpoints of the gateway

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<delLineEndpoint usn="1" version="1.0">
<Parameters>
<mgUIName>TPARTY.us.net</mgUIName>
</Parameters>
</delLineEndpoint>
</Methods>
</Command>
</CommandList>
```

An example of queryLineEndpoint request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<queryLineEndpoint usn="1" version="1.0">
<Parameters>
```

```
<mgUIName>TPARTY.us.net</mgUIName>
<endpointName>aaln/5</endpointName>
</Parameters>
</queryLineEndpoint>
</Methods>
</Command>
</CommandList>
```

An example of queryLineEndpoint response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<queryLineEndpoint usn="1" version="1.0">
<ReturnData>
<mgUIName>TPARTY.us.net</mgUIName>
<Row>
<endpointName>aaln/1</endpointName>
<circuitNumber>5</circuitNumber>
<cardCode>RDTLSG</cardCode>
<padGroup>PKLNL</padGroup>
<ground>N</ground>
<status>HASU</status>
<terminalNumber>6</terminalNumber>
<LEN>CARY 04 6 00 05</LEN>
</Row>
<RC>0</RC>
<MsgTxt>Query line endpoints operation was successful:
Gateway = TPARTY.us.net, Endpoint = aaln/5.</MsgTxt>
</ReturnData>
</queryLineEndpoint>
</Methods>
</Response>
</CommandList>
```

An example of queryAllLineEndpoints request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<queryAllLineEndpoints usn="1" version="1.0">
<Parameters>
<mgUIName>TPARTY.us.net</mgUIName>
</Parameters>
</queryAllLineEndpoints>
</Methods>
</Command>
</CommandList>
```

Note: The queryAllLineEndpoints query can potentially return a large list.

An example of queryAllLineEndpoints response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>cs2kCfgMgrIf<Interface>
<Methods>
<queryAllLineEndpoints usn="1" version="1.0">
<ReturnData>
<mgUIName>TPARTY.us.net</mgUIName>
<Row>
<endpointName>aaln/1</endpointName>
<circuitNumber>0</circuitNumber>
<cardCode>RDLSG</cardCode>
<padGroup>PKLNL</padGroup>
<ground>N</ground>
<status>HASU</status>
<terminalNumber>1</terminalNumber>
<LEN>CARY 04 6 00 00</LEN>
</Row>
<Row>
<endpointName>aaln/1</endpointName>
<circuitNumber>5</circuitNumber>
<cardCode>RDLSG</cardCode>
<padGroup>PKLNL</padGroup>
<ground>N</ground>
<status>HASU</status>
<terminalNumber>6</terminalNumber>
<LEN>CARY 04 6 00 05</LEN>
</Row>
<Row>
<endpointName>aaln/1</endpointName>
<circuitNumber>6</circuitNumber>
<cardCode>RDLSG</cardCode>
<padGroup>PKLNL</padGroup>
<ground>N</ground>
<status>HASU</status>
<terminalNumber>7</terminalNumber>
<LEN>CARY 04 6 00 06</LEN>
</Row>
<Row>
<endpointName>aaln/1</endpointName>
<circuitNumber>7</circuitNumber>
<cardCode>RDLSG</cardCode>
<padGroup>PKLNL</padGroup>
<ground>N</ground>
<status>HASU</status>
<terminalNumber>8</terminalNumber>
```

```
<LEN>CARY 04 6 00 07</LEN>
</Row>
<RC>0</RC>
<MsgTxt>Query line endpoints operation was successful:
Gateway = TPARTY.us.net</MsgTxt>
</ReturnData>
</queryLineEndpoint>
</Methods>
</Response>
</CommandList>
```

An example of addLineEndpoint request (endpoint in upper half of 2K LGRP)

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<<addLineEnpoing usn="1" version="1.0">
<Parameters>
<mgUIName>audiocodes1.com</mgUIName>>
<endpointName>tp/2016</endpointName>
<circuitNumber>2015</circuitNumber>
<cardcode>GWLPOT</cardCode>
<padGroup>PKNIL</padGroup>
<ground>N</ground>
</Parameters>
</addLineEndpoint>
</Methods>
</Command>
</CommandList>
```

Table 1
TP6310 card affected tags

Tag name	Description of valid values
<endpointName>	tp/1 to tp/2016 for TP6310 large line gateways
<circuitNumber>	0 to 2047 for 2K LGRP ***Note:*** Recommended values for large line gateways are 0 to 2015.

An example of delLineEndpoint request (endpoint in upper half of 2K LGRP)

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<delLineEndpoint usn="1" version="1.0">
```

```
<Parameters>
<<mgUIName> audiocodes1.com</mgUIName>
<endpointName>tp/2016</endpointName>
<</Parameters>
</delLineEndpoint>
</Methods>
</Command>
</CommandList>
```

Table 2
TP6310 affected tags

Tag name	Description of valid values
<endpointName>	tp/1 to tp/2016 for TP6310 large line gateways

An example of addCarrier request (Carrier last E1 carrier)

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<addCarrire usn="1" version="1.0">
<Parameters>>
<mgName>AC6310t1</mgName>
<carrierName>E1/63</carrierName>
<svcData>
<DPNSS>
<IID>63<IID>
</DPNSS>
</svcData>
<firstTN>1924</firstTN>
</Parameters>
</addCarrier>
</Methods>
</Command>
</CommandList>
```

Table 3
TP6310 card affected tags

Tag name	Description of valid values
<carrierName>	E1 carrier from E1/01 to E1/63
<IID>	DPNSS IID from 1 to 63

An example of addCarrier request (carrier last DS1 carrier)

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
```

```
<Methods>
<addCarrier usn="1" version="1.0">
<Parameters>
<mgName>AC6310T1</mgName>
<carrierName>DS1/84</carrierName>
<svcData>
<PRI>
<IID>31</IID>
</PRI>
</svdData>
<firstTN>1992</firstTN>
</Parameters>
</addCarrier>
</Methods>
</Command>
</CommandList>
```

Table 4
TP6310 affected tags

Tag name	Description of valid values
<carrierName>	DS1 carrier from DS1/01 to DS1/84

An example of deleteCarrier request (carrier last E1 carrier)

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<deleteCarrier usn="1" version="1.0">
<Parameters>
<mgName>AC63101</mgName>
<carrierName>E1/63</carrierName>
</Parameters>
</deleteCarrier>
</Methods>
</Command>
</CommandList>
```

Table 5
TP6310 affected tags

Tag name	Description of valid values
<carrierName>	E1 carrier from E1/01 to E1/63

An example of deleteCarrier request (carrier last DS1 carrier)

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
```

```
<Interface>cs2kCfgMgrIf</Interface>
<Methods>
<deleteCarrier usn="1" version="1.0">
<Parameters>
<mgName>AC63101</mgName>
<carrierName>DS1/84</carrierName>
</Parameters>
</deleteCarrier>
</Methods>
</Command>
</CommandList>
```

Table 6
TP6310 affected tags

Tag name	Description of valid values
<carrierName>	DS1 carrier from DS1/01 to DS1/84

Batch provisioning

Line Endpoint provisioning can be done using Batch Provisioning Tool (BPT). Details of the usage is explained in the "Batch Provisioning Tool" (page 325)chapter.

User authorization for line endpoint provisioning

The security for the OSSGate interface includes support for permission or ‘Authorization’ levels for commands. Each operation is associated with one or more authorization groups. To execute a command, a user must belong to at least one of the associated authorization groups. The authorization groups associated with each operation is listed in the table below. (For assocMG and queryMG authorization levels, see Nodes Provisioning permissions).

Command	User Group				
	mgcadm	mgcrw	mgcmtc	mgcsprov	mgcro
addLineEndpoint	X	X			
delLineEndpoint	X	X			
qryLineEndpoint	X	X	X	X	X

Return codes

The following table lists the possible return codes from the Line Endpoint provisioning operations.

Return code	Meaning
0	Successful operation
72	Add Line Endpoint operation, invalid input from client
73	Add Line Endpoint operation, unsupported GW type
74	Add Line Endpoint operation, not sufficient input
75	Add Line Endpoint operation, input value not in a correct format
76	Add Line Endpoint operation, failed to read the gateway data from the database
77	Add Line Endpoint operation, failed to add the endpoint to the GWC
78	Add Line Endpoint operation, specified endpoint already exists
79	Add Line Endpoint operation, have reached the maximum supported endpoint capacity on the GWC
80	Add Line Endpoint operation, failed to add the endpoint to the Core although the endpoint was added to the GWC successfully
81	Add Line Endpoint operation, GWC failed to add the end point
82	Add Line Endpoint operation, encountered an error after successfully adding the endpoint to the GWC
83	Delete Line Endpoint operation, invalid input from client
84	Delete Line Endpoint operation, not sufficient input
85	Delete Line Endpoint operation, input value not in a correct format
86	Delete Line Endpoint operation, failed to read the gateway data from the database
87	Delete Line Endpoint operation, unsupported GW type
88	Delete Line Endpoint operation, failed to delete the endpoint from its GWC
89	Delete Line Endpoint operation, failed to delete the endpoint from the Core
90	Delete Line Endpoint operation, failed to get the end point data from the database
91	Delete Line Endpoint operation, failed to retrieve the end point data from the database
92	Delete Line Endpoint operation, circuit number range is not valid
93	Delete Line Endpoint operation, delete operation was successful only partially due to failures in deleting the endpoints from the Core
94	Delete Line Endpoint operation, delete operation was successful only partially due to failures in deleting the endpoints from GWC
95	Delete Line Endpoint operation, delete operation was successful only partially
100	Query Line Endpoint operation, invalid input from client

Return code	Meaning
101	Query Line Endpoint operation, no endpoints are found in GWCEM
102	Query Line Endpoint operation, failed to retrieve the LEN data from the Core
103	Query Line Endpoint operation, failed to retrieve the endpoint data from the GWCEM

Limitations and restrictions

- Conversion of existing third party large line GW data to (I)SN08 and later formats is not supported. To use the third party gateway provisioning interface, the existing GW, associated end-points, service must be de-provisioned and re-added. Both new and old format data are allowed to co-exist, however the existing GW and the end-points will be managed in the same format as defined in the previous release.
- The capabilities can not be fully utilized until all the components of the OAMP are at the base (I)SN08 or later load line up. There is a dependency on the new LGRPTYPE in the CS 2000 Core that will be used by the third party GW nodes in the Core. The GWC devices are required to be on the (I)SN08 or later release as well to support spare profiles.

Carrier provisioning with OSSGate

Supported carrier provisioning commands

The following existing Carrier Provisioning operations are supported:

- Add Carrier
- Delete Carrier
- Get Carrier
- Get Endpoint
- List All Carriers

XML commands

AddCarrier: This interfaces will add carriers to a trunk gateway or a End Point Groups (EPG) of size 24 or 32 for facilities based trunks or from 4 to 1024 for 323 endpoint groups to a H.323 Gateway dependent upon market configuration. For H.323 gateways In order to provision a Gateway with multiple EPGs, this command should be called repeatedly with a unique EPG. This command will automatically grab the first available block of contiguous TID space TNs available on the GWC if the optional “FirstTN” tag is not used. The user can also specify the starting TID location. This option gives users the ability to assign EPs to specific TID locations if they desire. If a firstTN tag is included, the system will assign a contiguous group of 24 or 32 EPs starting at the given location. If any of the TNs in the range are unavailable the command will fail.

Description of method parameters

Method parameters are defined below. Input data is mandatory except where indicated otherwise.

- addCarrier - Method to add a group of carriers to any trunk gateway including H.323 gateways. This method has the following parameters:
  - — Input data
    - – usn - Unique Sequence Number. This parameter is introduced to facilitate sending multiple XML commands in one file. This parameter value will be used to match the responses with the requests. TYPE integer.
    - – version - The version refers to the version of the method or operation - Current version is 1.0
    - – mgName - A string value starting with PVG followed by alphanumeric or underscore

  - carrierName - A string value of size range 1 to 32 characters that represents the various carrier types like DS3, E1, SM, ST
  - svcData - A optional parameter. This can be one of PRI service type, V5.2 service type, ISUP or H.323
    A PRI service type has one parameter, IID - an integer value between 0 to 254 A v5.2 service type has three parameters, V5.2IID, V5.2LinkID and V5.2UALinkId. All these parameters are integers 1 to 4094 range.
  - firstTN - A optional parameter. An integer value between 1 to 4094
  - tidSize - A optional parameter. An integer value between 1 to 4094

- Output data
- usn (value should be the same as the input)
- version (value should be the same as the input)
- firstTID - A integer value that is a concatenation of node number and terminal number
- numberOfTids - An integer value that indicates the total number of allocated tids for the carrier.
- ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.

- getCarrier - Method to get a group of carriers associated to any trunk gateway including H.323 gateways. This method has the following parameters:

  Note: The same functionality is provided by new operation listAllCarriers when used with the FILTER_CARRIER option.

  - Input data
    - usn - Unique Sequence Number. This parameter is introduced to facilitate sending multiple XML commands in one file. This parameter value will be used to match the responses with the requests. TYPE integer.
    - version - The version refers to the version of the method or operation - Current version is 1.0
    - mgName A string value starting with PVG followed by alphanumeric or underscore
    - carrierName A string value of size range 1 to 32 characters that represents the various carrier types like DS3, E1, SM, ST

- — Output data
  - – usn (value should be the same as the input)
  - – version (value should be the same as the input)
  - – gwcName A string value starting with GWC- followed by a number between 0 to 255
  - – mgName A string value starting with PVG followed by alphanumeric or underscore
  - – carrierName A string value of size range 1 to 32 characters that represents the various carrier types like DS3, E1, SM, ST
  - – svcData - A optional parameter. This can be one of PRI service type, V5.2 service type, ISUP or H.323
A PRI service type has one parameter, IID - an integer value between 0 to 254 A v5.2 service type has three parameters, V5.2IID, V5.2LinkID and V5.2UALinkId. All these parameters are integers 1 to 4094 range.
  - – service - The type of service the carrier is associated with. It can be one of PRI, V5.2, ISUP or H.323
  - – numberOfTids - An integer value that indicates the total number of allocated tids for the carrier.
  - – ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.

- getEndPoint - Method to get endpoints associated with carriers on any trunk gateway including H.323 gateways. This method has the following parameters:
  - — Input data
    - – usn - Unique Sequence Number. This parameter is introduced to facilitate sending multiple XML commands in one file. This parameter value will be used to match the responses with the requests. TYPE integer.
    - – version - The version refers to the version of the method or operation - Current version is 1.0
    - – mgName - A string value starting with PVG followed by alphanumeric or underscore
    - – endpointName - A string value of size range 1 to 32 characters that represents the various carrier types like DS3, E1, SM, ST
  - — Output data
    - – usn (value should be the same as the input)

  - version (value should be the same as the input)
  - mgName A string value starting with PVG followed by alphanumeric or underscore
  - endpointName A string value of size range 1 to 32 characters that represents the various carrier types like DS3, E1, SM, ST
  - tid - A integer value that represent the Terminal-ID associated with the channel.
  - ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.

- deleteCarrier - Method to delete a group of carriers to any trunk gateway including H.323 gateways. This method has the following parameters:
  - Input data
    - usn - Unique Sequence Number. This parameter is introduced to facilitate sending multiple XML commands in one file. This parameter value will be used to match the responses with the requests. TYPE integer.
    - version - The version refers to the version of the method or operation - Current version is 1.0
    - mgName - A string value starting with PVG followed by alphanumeric or underscore
    - carrierName - A string value of size range 1 to 32 characters that represents the various carrier types like DS3, E1, SM, ST
  - Output data
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.
- listAllCarrier - Method to list all carriers associated to any trunk gateway including H.323 gateways. This method has the following parameters:
  - Input data
    - usn - Unique Sequence Number. This parameter is introduced to facilitate sending multiple XML commands in one file. This parameter value will be used to match the responses with the requests. TYPE integer.

  - version - The version refers to the version of the method or operation - Current version is 1.0
  - FILTER_GWC - A option from which one of the following four may be specified: FILTER_ALL, FILTER_GWC, FILTER_MG, FILTER_CARRIER.
  - gwcname - A string value starting with GWC- followed by a number between 0 to 255

- Output data
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - gwcname - A string value starting with GWC followed by a number between 0 to 255
  - mg name - A string value starting with PVG followed by alphanumeric or underscore
  - carrier name - A string value of size range 1 to 32 characters that represents the various carrier types like DS3, E1, SM, ST
  - svcData This can be one of PRI, ISUP, V5.2, H.323. A PRI type has one parameter - IID an integer value between 1 to 4094
  - service - The type of service the carrier is associated with. It can be one of PRI, V5.2, ISUP or H.323
  - IID - Data associated with PRI service type. Integer value between 0 to 254
  - endptList - A list of all endpoints associated with the carrier. Each endpoint has a name (string) and associated tid value (a integer value that is a concatenation of node number and terminal number).
  - ReturnCode - indicates via an integer value if the command has been successful or, if not, the error type and includes a brief textual message with further information.

Examples of XML commands and response messages

An example of addCarrier request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<addCarrier usn="1" version="1.0">
<Parameters>
<mgName>PVG190</mgName>
```

```
<carrierName>DS3_20.28</carrierName>
<svcData>
<PRI>
<IID>7</IID>
</PRI>
</svcData>
<firstTN>649</firstTN>
</Parameters>
</addCarrier>
</Methods>
</Command>
</CommandList>
```

Note: The firstTN field is optional.

An example of H.323 addCarrier request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<addCarrier usn="1" version="1.0">
<Parameters>
<mgName>PVG190</mgName>
<carrierName>DS3_20.28</carrierName>
<svcData>
<H323 />
</svcData>
<firstTN>649</firstTN>
</Parameters>
</addCarrier>
</Methods>
</Command>
</CommandList>
```

An example addCarrier response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<addCarrier usn="1" version="1.0">
<ReturnData>
<Row>
<firstTID>13649</firstTID>
<numberofTids>24</numberofTids>
</Row>
<RC>0</RC>
<MsgTxt>EGP: AddCarrierOpReplyStAdd CARRIER operation was
successful.</MsgTxt>
```

```
</ReturnData>
</addCarrier>
</Methods>
</Response>
</CommandList>
```

An example of getCarrier request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<getCarrier usn="1" version="1.0">
<Parameters>
<mgName>PVG190</mgName>
<carrierName>DS3_20.1</carrierName>
</Parameters>
</getCarrier>
</Methods>
</Command>
</CommandList>
```

An example of getCarrier response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<getCarrier usn="1" version="1.0">
<ReturnData>
<gwcName>GWC-1</gwcName>
<mgName>PVG190</mgName>
<Row>
<carrierName>DS3_20.1</carrierName>
<svcData>
<service>ISUP</service>
</svcData>
</Row>
<RC>0</RC>
<MsgTxt>Carrier was successfully retrieved</MsgTxt>
</ReturnData>
</getCarrier>
</Methods>
</Response>
</CommandList>
```

An example of getEndPoint request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
```

```
<Interface>EndptGrpProvIf</Interface>
<Methods>
<getEndPoint usn="1" version="1.0">
<Parameters>
<mgName>PVG8</mgName>
<endpointName>DS3_20.1.23</endpointName>
</Parameters>
</getEndPoint>
</Methods>
</Command>
</CommandList>
```

An example of getEndPoint response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<getEndPoint usn="1" version="1.0">
<ReturnData>
<mgName>PVG8</mgName>
<endpointName>DS3_20.1</endpointName>
<tid>200 23</tid>
<RC>0</RC>
<MsgTxt>Get Endpoint operation was successful.</MsgTxt>
</ReturnData>
</getEndPoint>
</Methods>
</Response>
</CommandList>
```

An example of deleteCarrier request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<deleteCarrier usn="1" version="1.0">
<Parameters>
<mgName>PVG8</mgName>
<carrierName>DS3_20.1</carrierName>
</Parameters>
</deleteCarrier>
</Methods>
</Command>
</CommandList>
```

An example of deleteCarrier response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
```

```
<Response>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<deleteCarrier usn="1" version="1.0">
<ReturnData>
<RC>0</RC>
<MsgTxt>Delete Carrier operation was successful.</MsgTxt>
</ReturnData>
</deleteCarrier>
</Methods>
</Response>
</CommandList>
```

An example of listAllCarriers request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<listAllCarriers usn="1" version="1.0">
<Parameters>
<FILTER_GWC>
<gwcName>GWC-6</gwcName>
</FILTER_GWC>
</Parameters>
</listAllCarriers>
</Methods>
</Command>
</CommandList>
```

An example of listAllCarriers response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>EndptGrpProvIf</Interface>
<Methods>
<listAllCarriers usn="1" version="1.0">
<ReturnData>
<GWC name= "GWC=6">
<mg name= "TrunkMG">
<carrier name= "DS3_20.2">
<svcData>
<service>PRI</service>
<IID>7</IID>
</svcData>
<endptList>
<endPt name= "DS3_20.2.1">
<tid>36 1</tid>
</endPt>
<endPt name="DS3_20.2.2">
```

```
<tid>36 2</tid>
</endPt>
<!--...-->
<endPt name="DS3_20.2.22">
<tid>36 22</tid>
</endPt>
<endPt name="DS3_20.2.23">
<tid>36 23</tid>
</endPt>
<endPt name="DS3_20.2.24">
<tid>36 24</tid>
</endPt>
</endptList>
</carrier>
<carrier name="DS3_20.9">
<svcData>
<service>ISUP</service>
</svcData>
<endptList>
<endPt name="DS3_20.9.1">
<tid>36 1401</tid>
</endPt>
<endPt name="DS3_20.9.2">
<tid>36 1402</tid>
</endPt>
<endPt name="DS3_20.9.3">
<tid>36 1403</tid>
</endPt>
<!--...-->
<endPt name="DS3_20.9.22">
<tid>36 1422</tid>
</endPt>
<endPt name="DS3_20.9.23">
<tid>36 1423</tid>
</endPt>
<endPt name="DS3_20.9.24">
<tid>36 1424</tid>
</endPt>
</endptList>
</carrier>
</mg>
</GWC>
<RC>0</RC>
<MsgTxt>
Retrieval of Carriers Successfully Completed.</MsgTxt>
</ReturnData>
</listAllCarriers>
</Methods>
</Response>
</CommandList>
```

User authorization for carrier operations

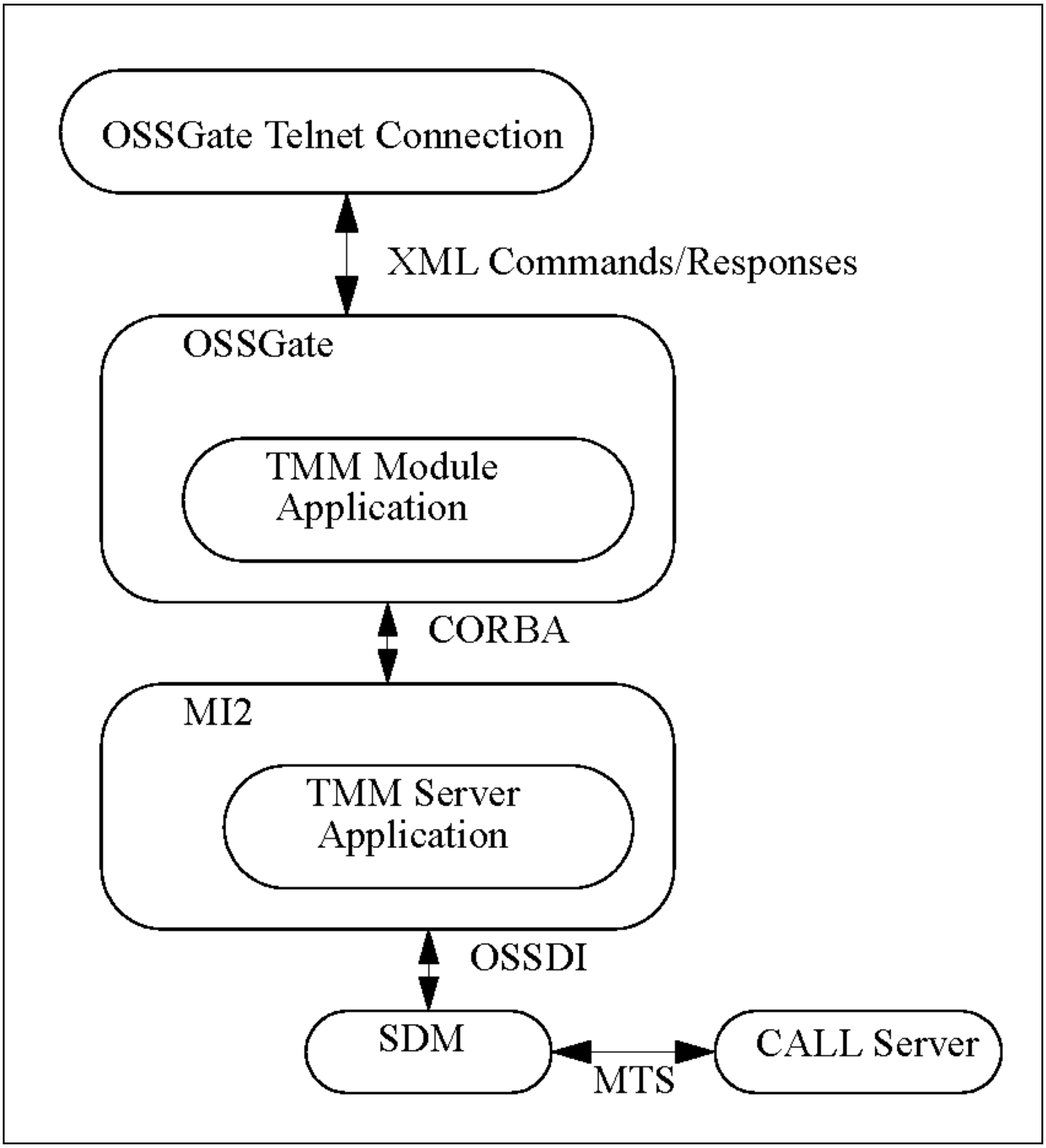
The security for the OSSGate interface includes support for permission or 'Authorization' levels for commands. Each operation is associated with one or more user groups. In order to execute a command, a user must belong to at least one of the associated user groups. The user groups associated with each Carrier Provisioning command are specified in the table below.

Command	User Group				
	trkadm	trkrw	trksprov	trkmtc	trlro
addCarrier	X	X			
deleteCarrier	X	X			
getEndpoint	X	X	X	X	X
get Carrier	X	X	X	X	X
get CarrierByFilter	X	X	X	X	X

Return code	Meaning
0	Successful operation
1	More data coming
2	Request Rejected due to Resource Limits Reached
3	Application was Commanded to Abort
4	Invalid input from client
5	Failed at GWC EM
6	Failed at Network View
7	Successful operation, but no data
8	Timeout at GWC EM

Trunk provisioning with OSSGate

The Trunk Provisioning application enables authenticated OSS clients to perform provisioning operations on legacy and MG-based trunk data from CS 2000 Management tool server. The provisioning operations include adding, changing, listing and deleting tuples. Access to this functionality is by connection to OSSGate.



Carrier VoIP trunk provisioning of gateway-hosted endpoints is a multiple-step process:

Step	Action
1	Carrier endpoints on a MG must be provisioned by using the Bulk Carrier Provisioning application. The Carrier Provisioning application is responsible for allocating TIDs for all channels of a carrier.
2	Provision the CS 2000 tables CLLI and TRKGRP with the appropriate data.
3	Provision trunk service on MG-based endpoints by using the Trunk Provisioning application to populate tables TRKSGRP and TRKMEM. The endpoint/TID which was allocated for the carrier channel in step 1 is specified when provisioning tables TRKSGRP and TRKMEM.

—End—

TRKSGRP tuples for PRI (Primary Rate Interface) trunk groups specify the D channel endpoint/TID. TRKMEM tuples specify endpoint/TID of a respective trunk group.

> ***Note 1:*** Non-MG-hosted endpoints (legacy trunks) do not require a separate TID allocation step. Provisioning of table TRKMEM with the carrier channel is all that is required.
>
> ***Note 2:*** The IID table on the respective GWC must be populated with the corresponding endpoint/channel data for MG-based PRI trunks.
>
> ***Note 3:*** The remaining tables on the CS 2000 for trunk provisioning (i.e. CLLI, TRKGRP, etc.) are unchanged by this activity and are still managed through Table Editor.

Supported trunk provisioning commands

The following XML commands are supported for Trunk Provisioning use OSSGate:

- AddTuple
- DelTuple
- ReplaceTuple
- GetTuple
- GetRange
- GetTupleWithoutGwcemCheck
- GetRangeWithoutGwcemCheck

Terminology and description of method parameters

- TableName - Name of the table. TYPE string. Min length 1, Max length 8. The valid values include "TRKSGRP" and "TRKMEM".
- Tuple - Information of the Tuple data. TYPE string.
- MGName - Name of the Media Gateway. TYPE string.
- EndPointName - Name of the Endpoint. TYPE string.
- BackupMGName - Name of the backup Media Gateway. TYPE string.
- BackupEndPointName - Name of the backup Endpoint. TYPE string.
- TupleKey - Key of the Tuple. TYPE string.
- NumberOfTuples - Number of the returnedTuples. TYPE string. If NumberOfTuples is less than 0, the number of the returned tuples is the default value in System.
- value - Value of the Return Code. TYPE string only contains number.
- text - Message of the operation result. TYPE string.
- severity - Type of the serverity. TYPE string. The valid values include "INFORMATION", "WARNING", "MINOR", "MAJOR" and "CRITICAL".
- msgTxtParm - Information of the parameters. TYPE string.
- GWCNumber - No. of the GWC. TYPE unsigned short. Min is 0.
- NodeNumber - No. of the Node. TYPE unsigned short. Min is 0.
- TerminalNumber - No. of the Terminal. TYPE unsigned short. Min is 0.
- BackupTerminalNumber - No. of the Backup Terminal. TYPE unsigned short. Min is 0.

The following parameters will be returned when some errors occur.

- Number - No. of the error. TYPE string only contains number. Valid values include "14100", "14101", "14102", "14103", "14104", "14105", "14107" and "14108".
- Message - Message of the error. TYPE string.
- Severity - Severity of the error. TYPE string. The valid values include "INFORMATION", "WARNING", "MINOR", "MAJOR" and "CRITICAL".
- Parami - Parameter, i is integer. TYPE string.
- name - Name of the Sub Component. TYPE string.
- value - Value of the Return Code. TYPE integer.
- text - Message of the operation result. TYPE string.

Description of method parameters

- AddTuple - Interface/Method that adds a tuple to the respective table
  - Input data
    - Version - Optional
    - TableName - Required
    - Tuple - Required
    - MGName - Optional
    - EndPointName - Optional
    - BackupMGName - Optional
    - BackupEndPointName - Optional
  - Output data
    - TableName - Required
    - value - Required. TYPE string only contains numbers.
    - text - Optional
    - severity - Optional
    - msgTxtParm - Optional
- DelTuple - Interface/Method that deletes a tuple from the respective table based on the tuple key or the full tuple.
  - Input data
    - Version - Optional
    - TableName - Required
    - TupleKey - Required
  - Output data
    - TableName - Required
    - value - Required. TYPE string only contains numbers.
    - text - Optional
    - severity - Optional
    - msgTxtParm - Optional
- ReplaceTuple - Interface/Method that changes a tuple based on the tuple key or the full tuple.
  - Input data

    - Version - Optional
    - TableName - Required
    - Tuple - Required
    - MGName - Optional
    - EndPointName - Optional
    - BackupMGName - Optional
    - BackupEndPointName - Optional

  - Output data
    - TableName - Required
    - value - Required. TYPE string only contains numbers.
    - text - Optional
    - severity - Optional
    - msgTxtParm - Optional

- GetTuple - Interface/Method that lists or queries a specific tuple based on the tuple key or full tuple from the respective table. If no key or tuple is provided, the first tuple in the specified table will be returned to the OSS.
  - Input data
    - Version - Optional
    - TableName - Required
    - TupleKey - Required

  - Output data
    - TableName - Required
    - value - Required. TYPE string only contains numbers.
    - text - Optional
    - severity - Optional
    - msgTxtParm - Optional
    - Tuple - Optional
    - GWCNumber - Optional
    - NodeNumber - Optional
    - TerminalNumber - Optional
    - BackupTerminalNumber - Optional

- GetRange - Interface/Method that lists the specified number of tuples from the respective table based on the tuple key or the full tuple of the first tuple in the range and the number of tuples requested. If a tuple key is not provided, the list of tuples will be captured from the beginning or “top” of the specified table. If the NumberOfTuples field is omitted, the default value is 1.
  - Input data
    - Version - Optional
    - TableName - Required
    - TupleKey - Required
    - NumberOfTuples - Optional
  - Output data
    - TableName - Required
    - value - Required. TYPE string only contains numbers.
    - text - Optional
    - severity - Optional
    - msgTxtParm - Optional
    - Tuple - Optional
    - GWCNumber - Optional
    - NodeNumber - Optional
    - TerminalNumber - Optional
    - BackupTerminalNumber - Optional
- GetTupleWithoutGwcemCheck - Interface/Method that lists or queries a specific tuple based on the tuple key or full tuple from the respective table without GWCEM check. If no key or tuple is provided, the first tuple in the specified table will be returned to the OSS.
  - Input data
    - Version - Optional
    - TableName - Required
    - TupleKey - Required
  - Output data
    - TableName - Required
    - value - Required. TYPE string only contains numbers.

    - text - Optional
    - severity - Optional
    - msgTxtParm - Optional
    - Tuple - Optional
    - GWCNumber - Optional
    - NodeNumber - Optional
    - TerminalNumber - Optional
    - BackupTerminalNumber - Optional

- GetRangeWithoutGwcemCheck - Interface/Method that lists the specified number of tuples from the respective table based on the tuple key or the full tuple of the first tuple in the range and the number of tuples requested without GWCEM check. If a tuple key is not provided, the list of tuples will be captured from the beginning or “top” of the specified table.
  - Input data
    - Version - Optional
    - TableName - Required
    - TupleKey - Required
    - NumberOfTuples - Required
  - Output data
    - TableName - Required
    - value - Required. TYPE string only contains numbers.
    - text - Optional
    - severity - Optional
    - msgTxtParm - Optional
    - Tuple - Optional
    - GWCNumber - Optional
    - NodeNumber - Optional
    - TerminalNumber - Optional
    - BackupTerminalNumber - Optional

XML commands

The following are examples of XML commands and the corresponding response messages for each operation supported for ADSL flow through provisioning. The XML coding in these examples is formatted for ease of understanding.

An example of addTuple request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>TrunkProv<Interface>
<Methods Version="2.0">
<AddTuple TableName="TRKMEM" Tuple="SUC101ISUPV2LP 1 0 GWC
5 18 1"/>
</Methods>
</Command>
</CommandList>
```

An example of AddTuple response without MGName and EndpointName

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>TrunkProv<Interface>
<Methods>
<AddTuple>
<ReturnCode TableName="TRKMEM" value="0" text="Command
Successful!"/>
</AddTuple>
</Methods>
</Response>
</CommandList>
```

An example of addTuple request for with MGName and EndpointName

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>TrunkProv<Interface>
<Methods Version="2.0">
<AddTuple TableName="TRKMEM" Tuple="PRIFORUPGRADEDOC
1 0 GWC PRIFORDOC DS1_10.1" MGName="PRIFORDOC"
EndPointName="DS1_10.1"/>
</Methods>
</Command>
</CommandList>
```

An example of AddTuple response for trunk

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkProv<Interface>
```

```
<Methods>
<AddTuple>
<ReturnCode TableName="TRKMEM" value="0" text="Command
Successful!"/>
</AddTuple>
</Methods>
</Response>
</CommandList>
```

An example of DelTuple request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>TrunkProv</Interface>
<Methods Version="2.0">
<DelTuple TableName="TRKMEM" TupleKey="IBNT2ISUPOG 0"/>
</Methods>
</Command>
</CommandList>
```

An example of DelTuple response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkProv</Interface>
<Methods>
<DelTuple>
<ReturnCode TableName="TRKMEM" value="0" text="Command
Successful!"/>
</DelTuple>
</Methods>
</Response>
</CommandList>
```

An example of ReplaceTuple request without MGName and EndpointName

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>TrunkProv</Interface>
<Methods Version="2.0">
<ReplaceTuple TableName="TRKMEM" Tuple="PRIFORUPGRADEDOC 3
0 GWC 0 67 38"/>
</Methods>
</Command>
</CommandList>
```

An example of ReplaceTuple response without MGName and EndpointName

```
<?xml version="1.0"?>
<CommandList>
```

```
<Response>
<Interface>TrunkProv</Interface>
<Methods>
<ReplaceTuple>
<ReturnCode TableName="TRKMEM" value="0" text="Command
Successful!"/>
</ReplaceTuple>
</Methods>
</Response>
</CommandList>
```

An example of ReplaceTuple request with MGName and EndpointName

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>TrunkProv</Interface>
<Methods Version="2.0">
<ReplaceTuple TableName="TRKMEM" Tuple="PRIFORUPGRADEDOC
3 0 GWC PRIFORDOC DS1_10.6" MGName="PRIFORDOC"
EndPointName="DS1_10.6"/>
</Methods>
</Command>
</CommandList>
```

An example of ReplaceTuple response with MGName and EndpointName

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkProv</Interface>
<Methods>
<ReplaceTuple>
<ReturnCode TableName="TRKMEM" value="0" text="Command
Successful!"/>
</ReplaceTuple>
</Methods>
</Response>
</CommandList>
```

An example of GetTuple request

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Command>
<Interface>TrunkProv</Interface>
<Methods Version="2.0">
<GetTuple TableName="TRKSGRP" TupleKey="IBNT2ISUPOG"/>
</Methods>
</Command>
</CommandList>
```

An example of GetTuple response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>TrunkProv</Interface>
<Methods>
<GetTuple>
<ReturnCode TableName="TRKSGRP" value="0" text="Command
Successful!"/>
<TupleData>
<Tuple>IBNT2ISUPOG 0 DS1SIG C7UP 2W N N UNEQ NONE Q764
THRH 100 DMSNODE $ NIL CIC</Tuple>
</TupleData>
</GetTuple>
</Methods>
</Response>
</CommandList>
```

An example of GetRange request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkProv</Interface>
<Methods Version="2.0">
<GetRange TableName="TRKMEM" TupleKey="PRIFORUPGRADEDOC 1"
NumberOfTuples="5"/>
</Methods>
</Command>
</CommandList>
```

An example of GetRange response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>TrunkProv</Interface>
<Methods>
<GetRange>
<ReturnCode TableName="TRKMEM" value="0" text="Command
Successful!"/>
<TupleData>
<Tuple>PRIFORUPGRADEDOC 1 0 PRIFORDOC DS1_10.1</Tuple>
<GWCNumber>0</GWCNumber>
<NodeNumber>67</NodeNumber>
<TerminalNumber>32</TerminalNumber>
</TupleData>
<TupleData>
<Tuple>PRI 2 0 GAOPRI E1_1001.2</Tuple>
<GWCNumber>0</GWCNumber>
<NodeNumber>67</NodeNumber>
<TerminalNumber>2</TerminalNumber>
```

```
</TupleData>
<TupleData>
<Tuple>PRI 3 0 GAOPRI E1_1001.3</Tuple>
<GWCNumber>0</GWCNumber>
<NodeNumber>67</NodeNumber>
<TerminalNumber>3</TerminalNumber>
</TupleData>
<TupleData>
<Tuple>H323TEST 2 0 GWC 3 72 2 ; TID not found in GWCEM;
node#=72;term#=2.</Tuple>
<GWCNumber>3</GWCNumber>
<NodeNumber>72</NodeNumber>
<TerminalNumber>2</TerminalNumber>
</TupleData>
<TupleData>
<Tuple>H323TEST 3 0 GWC 3 72 3; TID not found in GWCEM;
node#=72;term#=3</Tuple>
<GWCNumber>3</GWCNumber>
<NodeNumber>72</NodeNumber>
<TerminalNumber>3</TerminalNumber>
</TupleData>
</GetRange>
</Methods>
</Response>
</CommandList>
```

An example of GetTupleWithoutGwcemCheck request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkProv</Interface>
<Methods Version="2.0">
<GetTupleWithoutGwcemCheck TableName="TRKSGRP"
TupleKey="PVG40_2_PRI_USR 0"/>
</Methods>
</Command>
</CommandList>
```

An example of GetTupleWithoutGwcemCheck response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>TrunkProv</Interface>
<Methods>
<GetTupleWithoutGwcemCheck>
<ReturnCode TableName="TRKSGRP" value="0" text="Command
Successful!"/>
<TupleData>
```

```
<Tuple>PVG40_2_PRI_USR 0 DS1SIG ISDN 20 20 96ISOQSIG 2 N
STAND USER PT_PT USER N INTERNAL N N 255 N DEFAULT GWC 5
18 47 64K HDLC $ $</Tuple>
<GWCNumber>5</GWCNumber>
<NodeNumber>18</NodeNumber>
<TerminalNumber>47</TerminalNumber>
</TupleData>
</GetRange>
</Methods>
</Response>
</CommandList>
```

An example of GetRangeWithoutGwcemCheck request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkProv</Interface>
<Methods Version="2.0">
<GetRangeWithoutGwcemCheck TableName="TRKMEM"
TupleKey="PVG40_2_PRI_USR 1" NumberOfTuples="5"/>
</Methods>
</Command>
</CommandList>
```

An example of GetRangeWithoutGwcemCheck response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>TrunkProv</Interface>
<Methods>
<GetRangeWithoutGwcemCheck>
<ReturnCode TableName="TRKMEM" value="0" text="Command
Successful!"/>
<TupleData>
<Tuple>PVG40_2_PRI_USR 1 0 GWC 5 18 32</Tuple>
<GWCNumber>5</GWCNumber>
<NodeNumber>18</NodeNumber>
<TerminalNumber>32</TerminalNumber>
</TupleData>
<TupleData>
<Tuple>PVG40_2_PRI_USR 2 0 GWC 5 18 33</Tuple>
<GWCNumber>5</GWCNumber>
<NodeNumber>18</NodeNumber>
<TerminalNumber>33</TerminalNumber>
</TupleData>
<TupleData>
<Tuple>PVG40_2_PRI_USR 3 0 GWC 5 18 34</Tuple>
<GWCNumber>5</GWCNumber>
<NodeNumber>18</NodeNumber>
<TerminalNumber>34</TerminalNumber>
```

```
</TupleData>
<TupleData>
<Tuple>PVG40_2_PRI_USR 4 0 GWC 5 18 35</Tuple>
<GWCNumber>5</GWCNumber>
<NodeNumber>18</NodeNumber>
<TerminalNumber>35</TerminalNumber>
</TupleData>
<TupleData>
<Tuple>PVG40_2_PRI_USR 5 0 GWC 5 18 36</Tuple>
<GWCNumber>5</GWCNumber>
<NodeNumber>18</NodeNumber>
<TerminalNumber>36</TerminalNumber>
</TupleData>
</GetRangeWithoutGwcemCheck>
</Methods>
</Response>
</CommandList>
```

An example of error response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkProv</Interface>
<Methods>
<AddTuple>
<ReturnCode TableName="TRKMEM" value="70" text="ERROR, An
error was received from the Communication Server.">
<Subcomponent name="Communication Server" value="20"
text=’RC=3; TxtIdx=20; USN=a.20; message m0=RC=3;
TxtIdx=20; CMError="The GWC node number has been corrected
to 67\015TERMINAL ALREADY BOUND TO ANOTHER TRUNK\015";
end; end’/>
</ReturnCode>
</AddTuple>
</Methods>
</Response>
</CommandList>
```

User authorization for trunk provisioning operations

The security for the OSSGate interface includes support for permission or ‘Authorization’ levels for commands. Each operation is associated with one or more user groups. In order to execute a command, a user must belong to at least one of the associated user groups. The user groups associated with each Trunk Provisioning operation is listed in the table below.

Command	User Group				
	trkadm	trkiprov	trksprov	trkmtc	trlro
GetTuple	X	X	X	X	X
GetRange	X	X	X	X	X
AddTuple	X	X			
ReplaceTuple	X	X			
DelTuple	X	X			
GetTupleWithoutGwcemCheck	X	X	X	X	X
GetRangeWithoutGwcemCheck	X	X	X	X	X

Limitations and restrictions

The Abort Operation command is unsupported. The ability to abort an XML request received from OSSGate is not supported. The Trunk Provisioning application writes to only one data repository and does not need to “roll back” data. Either the XML request is processed successfully, or a corresponding error message is returned to OSSGate. If an OSS client would like to terminate or abort a table operation, the client can break the telnet link to OSSGate.

The Pause Operation command is unsupported. The ability to pause an XML request received from OSSGate will not be supported. The Trunk Provisioning application resides in a unified framework which cannot pause one application while allowing others to continue unabated. If an OSS client wishes to pause a table operation, the client can abort it by breaking the telnet link to OSSGate. When the client reconnects, the operation will have to be restarted.

Return codes

The Trunk Provisioning application will respond to each XML request with an XML response. The XML response will contain a return code that will signify whether the request was processed properly or if an error was incurred.

The return code of “0” means the table operation request was processed successfully.

The following return codes for errors may be sent back if an error occurs during the processing of a table operation request.

Operation	Return Code	Description	MsgText	Comment
All Trunk Provisioning operations	0	Successful operation	Varies, depending on the operation and specific scenario.	This error code means that there was no error and the request was processed successfully.
All Trunk Provisioning operations	501	Insufficient security privileges to perform this action.	The User is not authorized to perform the operation	User is not a member of correct authorization group.
All Trunk Provisioning operations	701	Unsuccessful operation. Trunk Provisioning is unavailable.	“The Trunk Provisioning application is currently off-line. Please try your request again later.”	This error generally means the Trunk Provisioning application is off-line and should be manually restarted using the administrative control tool.
All Trunk Provisioning operations	702	Unsuccessful operation.	“An error has occurred while attempting to execute the Trunk Provisioning command.”	Probably due to catching an unexpected exception. An example is the database connection being down.
All Trunk Provisioning operations	703	Unsuccessful operation. Invalid customer input.	“An invalid XML message was received. Please try your request again.”	This error generally means the OSS sent in an invalid message. The OSS should verify their XML using the XSD. If it complies, Nortel GPS should be notified.
All Trunk Provisioning operations	704	Unsuccessful operation. GWCEM error.	“An error has occurred within the GWCEM application.”	Probably due to catching an unexpected exception.

Operation	Return Code	Description	MsgText	Comment
Add Tuple Change Tuple	705	Unsuccessful operation.	Varies, depending on the operation and specific scenario.	There is a potential data integrity issue within a database.
Add Tuple	706	Unsuccessful operation. Tuple Already Exists.	Varies, depending on the operation and specific scenario.	It is possible someone back-door added the tuple into the CS (bypassing the CS 2000 Management Tools server).
All Trunk Provisioning operations	707	Unsuccessful operation.	Varies, depending on the operation and specific scenario.	Major problem affecting many areas of the server and needs to be resolved immediately. (possible loss of connection between the provisioning applications and the Communication Server).
Change Tuple Delete Tuple Query Tuple	708	Unsuccessful operation. Tuple Not Found.	Varies, depending on the operation and specific scenario.	It is possible someone erased the tuple from the Communication Server.
All Trunk Provisioning operations	709	Unsuccessful operation. Unknown internal problem.	Varies, depending on the operation and specific scenario.	
GetRange GetRangeWithoutGwcemCheck	710	Unsuccessful operation. NumberOfTuples Invalid.	Varies, depending on the operation and specific scenario.	

V5.2 carrier provisioning with OSSGate

V5 is a standard interface between the access network and the carrier switch for basic telephony, ISDN and semi-permanent leased lines. A V5.2 Interface consists of a ‘bundle’ of ‘n’ E1 carriers where 1 £ n £ 16. OSS can provision the V5.2 interface with OSSGate. The XML template defines the data which the OSS must send in order to provision a V5.2 interface. The data arriving from the OSS is parsed to route it to the right application. The V5.2 configuration data which is likely to be passed from the OSS is as follows:

- The interface ID
- The number of E1 links in the interface.
- The gateways on which these links are located.
- The links which contain the signaling.

Functionality supported by OSSGate for V5.2 interface include add, delete and list V5.2 interfaces.

Supported V5.2 provisioning commands

The following V5.2 Provisioning operations are supported in (I)SN06 and later:

- Add Interface
- Delete Interface
- List Interface
- List All Interfaces
- Add V5Prov Template
- Delete V5Prov Template
- List V5Prov Template
- List All V5Prov Templates
- Add V5Signalling Template
- Delete V5Signalling Template
- List V5Signalling Template
- List All V5Signalling Templates
- Add V5Ring Template
- Delete V5Ring Template
- List V5Ring Template

- List All V5Ring Templates

Terminology and description of method parameters

- siteGwcLoc - A four character site identifier, together with frame and unit numbers that together allow the AN to be identified. Format: <Host identifier>, <Frame number=0-511> <Unit number=0-9> Example - LG 02 1
- gwcId - This is the number of the GWC on which the V5.2 Interface will reside. Range 0~255. Type integer.
- v52InterfaceId - A unique value for interface ID. Range 0~16777215. Type integer.
- v5ProvRef - V5 provisioning identifier, identifies V5 provisioning variant used with the V5.2.
- v5SigTableRef - V5 signaling identifier, identifies V5 signaling variant used with the V5.2.
- v5RingTableRef - V5 ring identifier, identifies V5 ring variant used with the V5.2.
- maxlinesSelector - A selector field used for future support of increased MAXLINES of 3072. Currently, REG is only supported for IP_V52. Possible values are "REG", "PRIM", "SIS1", "SIS2". Type string.
- maxlines - Maximum lines defined on a V5.2 interface. Range 1~3072. Type integer.

  Note: Each V5.2 Interface can consists of a ‘bundle’ of ‘n’ E1 carriers where 1 £ n £ 16 and each bundle require the following parameters data filled:

- linkId - Identifiers to each of the E1 carriers (up to 16) that make up the V5.2 interface. Range 1~4094. Type integer.
- epGrp - E1 carrier on media gateway in Aspen format.
  Formats:
  Aspen: <GW Name>.E1_<([1-9]1[0-5])(0[1-9]|1[0-9]|2[0-9]|3[0-2])>
  Example: PVG1.E1_1111
- v5provid - V5 provisioning variant ID. Range 0~127. Type integer.
- bcctimer - Two Bearer channel control (BCC) timers sets both timers TBCC1 and TBCC4 to values between 500 and 1500 ms. (1=100ms). Range 5~15. Type integer.
- cchnlinflist - C-channel link information. Defines on what link and channel the protocol process are carried on. Array of 1-16.
- chnlid - Control Channel Number. Range 0~65535. Type integer.

- lcc - Link and C-channel.
- lnk - V5.2 Link Number associated with physical GPP P-side link. Range 1~16. Type integer.
- chnl - C-channel Number. Possible values are 16, 15, 31. Type integer.
- cpathlist - Type of control messaging carried on C-Channel. Possible values are "CTRL", "PSTN", "ISDD", "ISDF", "ISDP", "PSET". Type string array.
- prot1 - Protection Link 1. Secondary link protection group 1 switches to if primary C-channel link fails. Range 1~16. Type integer.
- prot2 - Protection link 2. Standby link and C-channel for protection group 2, first entry being link second entry. Array of up to 3 (lnk, chnl) pairs.
- lnk - link id. Range 1~16. Type integer.
- chnl - channel. Possible values are 16, 15, 31. Type integer.
- alarmthreshold - Alarm Threshold level of V5.2 link failures before a major alarm is triggered. Range 0~100. Type integer.
- v5sigid - A unique identifier for V5Sig. Range string of up to 16 alphanumeric. Type string.
- atten - Attenuation. Possible values are "V5_NONE", "V5_ANALOG", "V5_DIGITAL". Type string
- apa - Accelerated port alignment. Possible values are "N", "Y". Type string.
- plf - Park line feed. Possible values are "N", "Y". Type string.
- ds1flash - Digit 1 register recall. Possible values are "N", "Y". Type string.
- eoc - End of call signaling. Possible values are "N", "Y". Type string.
- suppind - Suppression indicator. Possible values are "NO_SUPP", "LE_SUPP", "TE_SUPP", "LE_TE_SUPP". Type string.
- plsdur - Plus duration type. Range 0~31. Type integer.
- mtrpn - Meter pulse notification. Possible values are "N", "Y". Type string.
- lroa - Line reversal on answer. Possible values are "N", "Y", "CHKLN". Type string.
- lrosfd - Line reversal on seizure and forward disconnect. Possible values are "N", "Y". Type string.
- rngtype - Provisionable ring type. Possible values are "C3C", "C3D", "C6F", "C5I", "C6R". Type string.

- v5ringid - A unique identifier for V5Ring. Range string of up to 16 alphanumeric. Type string.
- std - Standard ring. Range 0~31. Type integer.
- r01~r15 - Ring char 1 to ring char 15. Range 0~31. Type integer.
- version - The version refers to the version of the method or operation - e.g., the AddInterface operation may have version 1.0 and 2.0, where version 1.0 takes parameters X and Y and version 2.0 takes parms X, Y, and Z. TYPE decimal.
- usn - Unique Sequence Number. This parameter is introduced to facilitate sending multiple XML commands in one file. This parameter value will be used to match the responses with the requests. TYPE integer.
- RC - Return Code. A numeric value indicating the result of the attempted operation (see Table 15, for range and definitions of each possible value). TYPE short.
- MsgTxt - Appropriate text that describes the result of the operation. TYPE string.

Description of method parameters

- addInterface - Interface/Method that adds a V5.2 Interface and associated carriers. This method has the following parameters:
  - Input data
    - usn
    - version
    - siteGwcLoc - The AN node location.
    - gwcId - This is the number of the GWC on which the V5.2 Interface will reside.
    - v52InterfaceId - A unique value for interface ID. Range 0~16777215. Type integer.
    - v5ProvRef - V5 provisioning template identifier.
    - v5SigTableRef - V5 signaling template identifier.
    - v5RingTableRef - V5 ring template identifier.
    - maxlinesSelector - Possible values are "REG", "PRIM", "SIS1", "SIS2".
    - maxlines - Maximum lines defined on a V5.2 interface. Range 1-3072. Type integer.
      The following data will be specified for each of the E1 carriers. At least one pair of E1 carriers has to be data filled. Each V5.2

Interface can consists of a ‘bundle’ of ‘n’ E1 carriers where 1 < n < 16 and each bundle require the following parameters data filled: Required data for each of E1 carrier:

- linkId - Identifiers to each of the E1 carriers (up to 16) that make up the V5.2 interface. Range 1-4094.
- epGrp - E1 carrier on media gateway in Aspen format.

- Output date
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - RC - Return Code
  - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

- deleteInterface - Interface/Method that deletes the specified V5.2 interface and associated carriers. This method has the following parameters:
  - Input data
    - usn
    - version
    - v52InterfaceId
  - Output date
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - RC - Return Code
    - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.
- listInterface - Interface/Method that returns a V5.2 interface and associated carriers. This method has the following parameters:
  - Input data
    - usn
    - version
    - v52InterfaceId

- Output date
  - usn (value should be the same as the input)
  - version (value should be the same as the input)
  - siteGwcLoc - The AN node location.
  - gwcId - This is the number of the GWC on which the V5.2 Interface will reside.
  - v52InterfaceId - A unique value for interface ID. Range 0~16777215. Type integer.
  - v5ProvRef - V5 provisioning template identifier.
  - v5SigTableRef - V5 signaling template identifier.
  - v5RingTableRef - V5 ring template identifier.
  - maxlinesSelector - Possible values are "REG", "PRIM", "SIS1", "SIS2".
  - maxlines - Maximum lines defined on a V5.2 interface. Range 1~3072. Type integer.
  - RC - Return Code
  - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.
    The following data will be returned for each of the E1 carriers:
  - linkId - Identifiers to each of the E1 carriers (up to 16) that make up the V5.2 interface. Range 1~4094.
  - epGrp - E1 carrier on media gateway in Aspen format.

- listAllInterfaces - Interface/Method that returns all V5.2 Interfaces and associated carriers. This method has the following parameters:
  - Input data
    - usn
    - version
  - Output date
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - RC - Return Code
    - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

        - keylist - This is the number of the GWC on which the V5.2 Interface will reside.

- addV5ProvTemplate - Interface/Method that adds a V5PROV template. This method has the following parameters:
    - Input data
        - usn
        - version
        - v5provid - V5 provisioning variant ID. Range 0~127. Type integer.
        - bcctimer - Two Bearer channel control (BCC) timers sets both timers TBCC1 and TBCC4 to values between 500 and 1500 ms. (1=100ms). Range 5~15. Type integer.
          The following data will be specified for each of the C-channel link information
            - chnlid
            - lnk - link id. Range 1~16. Type integer.
            - chnl - channel. Possible values are 16, 15, 31. Type integer.
            - cpathlist - Type of control messaging carried on C-Channel. Possible values are "CTRL", "PSTN", "ISDD", "ISDF", "ISDP", "PSET". Type string array.
            - prot1- Protection Link 1. Range 1~16. Type integer.
              The following data will be specified for each protection link 2. Prot2 is an array of up to 3 (lnk, chnl) pairs.
                - link - link id. Range 1-16. Type integer.
                - chnl - channel. Possible values are 16, 15, 31. Type integer.
        - alarmthreshold - Alarm Threshold level of V5.2 link failures before a major alarm is triggered. Range 0~100. Type integer.
    - Output date
        - usn (value should be the same as the input)
        - version (value should be the same as the input)
        - RC - Return Code
        - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

- deleteV5ProvTemplate - Interface/Method that deletes a V5PROV template. This method has the following parameters:
    - Input data
        - usn
        - version
        - v5provid - the V5 provisioning variant ID. Range 0~127.
    - Output date
        - usn (value should be the same as the input)
        - version (value should be the same as the input)
        - RC - Return Code
        - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.
- listV5ProvTemplate - Interface/Method that returns a V5PROV template. This method has the following parameters:
    - Input data
        - usn
        - version
        - v5provid - the V5 provisioning variant ID. Range 0~127.
    - Output date
        - usn (value should be the same as the input)
        - version (value should be the same as the input)
        - RC - Return Code
        - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.
        - v5provid - V5 provisioning variant ID. Range 0~127.
        - bcctimer - Two Bearer channel control (BCC) timers sets both timers TBCC1 and TBCC4 to values between 500 and 1500 ms. (1=100ms). Range 5~15.
        - The following data will be specified for each of the C-channel link information.
            - chnlid
            - Link and C-channel

        - lnk - link id. Range 1~16. Type integer.
        - chnl - channel. Possible values are 16, 15, 31. Type integer.
        - cpathlist - Type of control messaging carried on C-Channel. Possible values are "CTRL", "PSTN", "ISDD", "ISDF", "ISDP", "PSET". Type string array.

    - prot1- Protection Link 1. Range 1~16. Type integer.
    - The following data will be specified for each protection link 2. Prot2 is an array of up to 3 (lnk, chnl) pairs.
    - lnk - link id. Range 1~16. Type integer.
    - chnl - channel. Possible values are 16, 15, 31. Type integer.
    - alarmthreshold - Alarm Threshold level of V5.2 link failures before a major alarm is triggered. Range 0~100. Type integer.

- listAllV5ProvTemplates - Interface/Method that returns all V5PROV templates. This method has the following parameters:
  - Input data
    - usn
    - version

  - Output date
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - keylist - V5 provisioning variant ID. Range 0~127.

- addV5SigTemplate - Interface/Method that adds a V5SIG template. This method has the following parameters:
  - Input data
    - usn
    - version
    - v5sigid - A unique identifier for V5Sig. Range string of up to 16 alphanumeric. Type string.
    - atten- Attenuation. Possible values are "V5_NONE", "V5_ANALOG", "V5_DIGITAL". Type string
    - apa - Accelerated port alignment. Possible values are "N", "Y". Type string.
    - plf - Park line feed. Possible values are "N", "Y". Type string.

- ds1flash - Digit 1 register recall. Possible values are "N", "Y". Type string.
- eoc - End of call signaling. Possible values are "N", "Y". Type string.
- suppind - Suppression indicator. Possible values are "NO_SUPP", "LE_SUPP", "TE_SUPP", "LE_TE_SUPP". Type string.
- plsdur - Plus duration type. Range 0~31. Type integer.
- mtrpn - Meter pulse notification. Possible values are "N", "Y". Type string.
- lroa- Line reversal on answer. Possible values are "N", "Y", "CHKLN". Type string.
- lrosfd - Line reversal on seizure and forward disconnect. Possible values are "N", "Y". Type string.
- rngtype - Provisionable ring type. Possible values are "C3C", "C3D", "C6F", "C5I", "C6R". Type string.

— Output date

- usn (value should be the same as the input)
- version (value should be the same as the input)
- RC - Return Code
- MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

- deleteV5SigTemplate - Interface/Method that deletes a V5SIG template. This method has the following parameters:
  - Input data
    - usn
    - version
    - v5sigid - The unique identifier for V5Sig.
  - Output date
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - RC - Return Code

    - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

- listV5SigTemplate - Interface/Method that returns a V5SIG template. This method has the following parameters:
  - Input data
    - usn
    - version
    - v5sigid - The unique identifier for V5Sig.
  - Output date
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - RC - Return Code
    - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.
    - v5sigid - A unique identifier for V5Sig.
    - atten - Attenuation. Possible values are "V5_NONE", "V5_ANALOG", "V5_DIGITAL".
    - apa - Accelerated port alignment. Possible values are "N", "Y".
    - plf - Park line feed. Possible values are "N", "Y".
    - ds1flash - Digit 1 register recall. Possible values are "N", "Y".
    - eoc - End of call signaling. Possible values are "N", "Y".
    - suppind - Suppression indicator. Possible values are "NO_SUPP", "LE_SUPP", "TE_SUPP", "LE_TE_SUPP".
    - plsdur - Plus duration type. Range 0~31.
    - mtrpn - Meter pulse notification. Possible values are "N", "Y".
    - lroa - Line reversal on answer. Possible values are "N", "Y", "CHKLN".
    - lrosfd - Line reversal on seizure and forward disconnect. Possible values are "N", "Y".
    - rngtype - Provisionable ring type. Possible values are "C3C", "C3D", "C6F", "C5I", "C6R".
    - ssonhook - Signal SS: On-hook message flag. Possible values are "N", "Y".

- listAllV5SigTemplates - Interface/Method that returns all V5SIG templates. This method has the following parameters:
    - Input data
        - usn
        - version
    - Output date
        - usn (value should be the same as the input)
        - version (value should be the same as the input)
        - RC - Return Code
        - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.
        - keylist - The unique identifier for V5Sig.
- addV5RingTemplate - Interface/Method that adds a V5RING template. This method has the following parameters
    - Input data
        - usn
        - version
        - v5ringid - A unique identifier for V5Ring. Range string of up to 16 alphanumeric. Type string.
        - std - Standard ring. Range 0~31. Type integer.
        - r01 ~ r15 - Ring char 1 to ring char 15. Range 0~31. Type integer.
    - Output date
        - usn (value should be the same as the input)
        - version (value should be the same as the input)
        - RC - Return Code
        - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.
- deleteV5RingTemplate - Interface/Method that deletes a V5RING template. This method has the following parameters:
    - Input data
        - usn

    - version
    - v5ringid - A unique identifier for V5Ring. Range string of up to 16 alphanumeric. Type string.
  - Output date
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - RC - Return Code
    - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

- listV5RingTemplate - Interface/Method that gets the configuration data for a V5.2 Ring Template.
  - Input data
    - usn
    - version
    - v5ringid - The unique identifier for V5Ring.
  - Output date
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - RC - Return Code
    - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.
    - v5ringid - The unique identifier for V5Ring.
    - std - Standard ring. Range 0~31.
    - r01 ~ r15 - Ring char 1 to ring char 15. Range 0~31.

- listAllV5RingTemplates - Interface/Method that returns all V5RING templates. This method has the following parameters:
  - Input data
    - usn
    - version
  - Output date

- usn (value should be the same as the input)
- version (value should be the same as the input)
- RC - Return Code
- MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.
- keylist - The unique identifier for V5Ring.

XML commands

The following are examples of XML commands and the corresponding response messages for each operation supported for V5.2. The XML coding in these examples is formatted for ease of understanding.

Operation addInterface

An example of addInterface request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<addInterface usn="1" version="1.0">
<Parameters>
<siteGwcLoc>LG 02 1</siteGwcLoc>
<gwcId>73</gwcId>
<v52InterfaceId>123</v52InterfaceId>
<linkMapTable>
<linkId>1</linkId>
<epGrp>PVG1.E1_705</epGrp>
<linkId>2</linkId>
<epGrp>PVG1.E1_706</epGrp>
</linkMapTable>
<v5ProvRef>20</v5ProvRef>
<v5SigTableRef>DEFAULT</v5SigTableRef>
<v5RingTableRef>DEFAULT</v5RingTableRef>
<maxlinesSelector>REG</maxlinesSelector>
<maxlines>100</maxlines>
</Parameters>
</addInterface>
</Methods>
</Command>
</CommandList>
```

An example of addInterface response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
```

```
<Interface>v5CfgMgrIf</Interface>
<Methods>
<addInterface usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
</addInterface>
</Methods>
</Response>
</CommandList>
```

Operation deleteInterface

An example of deleteInterface request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<deleteInterface usn="1" version="1.0">
<Parameters>
<v52InterfaceId>123</v52InterfaceId>
</Parameters>
</deleteInterface>
</Methods>
</Command>
</CommandList>
```

An example of deleteInterface response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<deleteInterface usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
</deleteInterface>
</Methods>
</Response>
</CommandList>
```

Operation listInterface

An example of listInterface request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listInterface usn="1" version="1.0">
<Parameters>
```

```
<v52InterfaceId>123</v52InterfaceId>
</Parameters>
</listInterface>
</Methods>
</Command>
</CommandList>
```

An example of listInterface response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listInterface usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
<ReturnData>
<siteGwcLoc>LG 02 1</siteGwcLoc>
<gwcId>73</gwcId>
<v52InterfaceId>123</v52InterfaceId>
<linkMapTable>
<linkId>1</linkId>
<epGrp>PVG1.E1_705</epGrp>
<linkId>2</linkId>
<epGrp>PVG1.E1_706</epGrp>
</linkMapTable>
<v5ProvRef>20</v5ProvRef>
<v5SigTableRef>DEFAULT</v5SigTableRef>
<v5RingTableRef>DEFAULT</v5RingTableRef>
<maxlinesSelector>REG</maxlinesSelector>
<maxlines>100</maxlines>
</ReturnData>
</listInterface>
</Methods>
</Response>
</CommandList>
```

Operation listAllInterfaces

An example of listAllInterfaces request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listAllInterfaces usn="1" version="1.0">
<Parameters>
</Parameters>
</listAllInterfaces>
</Methods>
</Command>
```

```
</CommandList>
```

An example of listAllInterfaces response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listAllInterfaces usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
<ReturnData>
<keylist>73</keylist>
<keylist>74</keylist>
</ReturnData>
</listAllInterfaces>
</Methods>
</Response>
</CommandList>
```

Operation addV5ProvTemplate

An example of addV5ProvTemplate request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<addV5ProvTemplate usn="1" version="1.0">
0<Parameters>
<v5provid>66</v5provid>
<bcctimer>5</bcctimer>
<cchnlinflist>
<chnlid>0</chnlid>
<lcc>
<lnk>1</lnk>
<chnl>16</chnl>
</lcc>
<cpathlist>CTRL</cpathlist>
<chnlid>1</chnlid>
<lcc>
<lnk>3</lnk>
<chnl>16</chnl>
</lcc>
<cpathlist>PSTN</cpathlist>
</cchnlinflist>
<prot1>2</prot1>
<prot2>
<lnk>4</lnk>
<chnl>16</chnl>
</prot2>
```

```
<alarmthreshold>20</alarmthreshold>
</Parameters>
</addV5ProvTemplate>
</Methods>
</Command>
</CommandList>
```

An example of addV5ProvTemplate response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<addV5ProvTemplate usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
</addV5ProvTemplate>
</Methods>
</Response>
</CommandList>
```

Operation deleteV5ProvTemplate

An example of deleteV5ProvTemplate request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<deleteV5ProvTemplate usn="1" version="1.0">
<Parameters>
<v5provid>66</v5provid>
</Parameters>
</deleteV5ProvTemplate>
</Methods>
</Command>
</CommandList>
```

An example of deleteV5ProvTemplate response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<deleteV5ProvTemplate usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
</deleteV5ProvTemplate>
</Methods>
</Response>
```

```
</CommandList>
```

Operation listV5ProvTemplate

An example of listV5ProvTemplate request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listV5ProvTemplate usn="1" version="1.0">
<Parameters>
<v5provid>66</v5provid>
</Parameters>
</listV5ProvTemplate>
</Methods>
</Command>
</CommandList>
```

An example of listV5ProvTemplate response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listV5ProvTemplate usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
<v5provid>66</v5provid>
<bcctimer>5</bcctimer>
<cchnlinflist>
<chnlid>0</chnlid>
<lcc>
<lnk>1</lnk>
<chnl>16</chnl>
</lcc>
<cpathlist>
<cpathlist>CTRL</cpathlist>
</cpathlist>
<chnlid>1</chnlid>
<lcc>
<lnk>3</lnk>
<chnl>16</chnl>
</lcc>
<cpathlist>
<cpathlist>PSTN</cpathlist>
</cpathlist>
<cchnlinflist>
<prot1>2</prot1>
<prot2>
<lnk>4</lnk>
```

```
<chnl>16</chnl>
</prot2>
<alarmthreshold>20</alarmthreshold>
</listV5ProvTemplate>
</Methods>
</Response>
</CommandList>
```

Operation listAllV5ProvTemplates

An example of listAllV5ProvTemplates request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listAllV5ProvTemplates usn="1" version="1.0">
<Parameters>
</Parameters>
</listAllV5ProvTemplates>
</Methods>
</Command>
</CommandList>
```

An example of listAllV5ProvTemplates response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listAllV5ProvTemplates usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
<keylist>1</keylist>
<keylist>4</keylist>
<keylist>66</keylist>
</listAllV5ProvTemplates>
</Methods>
</Response>
</CommandList>
```

Operation addV5SigTemplate

An example of addV5SigTemplate request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<addV5SigTemplate usn="1" version="1.0">
<Parameters>
```

```
<v5sigid>SIG1</v5sigid>
<atten>V5_ANALOG</atten>
<apa>N</apa>
<plf>N</plf>
<ds1flash>N</ds1flash>
<eoc>N</eoc>
<suppind>NO_SUPP</suppind>
<plsdur>5</plsdur>
<mtrpn>N</mtrpn>
<lroa>N</lroa>
<lrosfd>N</lrosfd>
<mgtype>C3C</mgtype>
</Parameters>
</addV5SigTemplate>
</Methods>
</Command>
</CommandList>
```

An example of addV5SigTemplate response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<addV5SigTemplate usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
</addV5SigTemplate>
</Methods>
</Response>
</CommandList>
```

Operation deleteV5SigTemplate

An example of deleteV5SigTemplate request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<deleteV5SigTemplate usn="1" version="1.0">
<Parameters>
<v5sigid>SIG1</v5sigid>
</Parameters>
</deleteV5SigTemplate>
</Methods>
</Command>
</CommandList>
```

An example of deleteV5SigTemplate response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<deleteV5SigTemplate usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
</deleteV5SigTemplate>
</Methods>
</Response>
</CommandList>
```

Operation listV5SigTemplate

An example of listV5SigTemplate request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listV5SigTemplate usn="1" version="1.0">
<Parameters>
<v5sigid>SIG1</v5sigid>
</Parameters>
</listV5SigTemplate>
</Methods>
</Command>
</CommandList>
```

An example of listV5SigTemplate response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listV5SigTemplate usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
<ReturnData>
<v5sigid>SIG1</v5sigid>
<atten>V5_ANALOG</atten>
<apa>N</apa>
<plf>N<plf>
<ds1flash>N</ds1flash>
<eoc>N</eoc>
<suppind>NO_SUPP</suppind>
<plsdur>5</plsdur>
<mtrpn>N<mtrpn>
<lroa>N</lroa>
```

```
<lrosfd>N</lrosfd>
<mgtype>C3C</mgtype>
<ssonhook>N</ssonhook>
</ReturnData>
</listV5SigTemplate>
</Methods>
</Response>
</CommandList>
```

Operation listAllV5SigTemplates

An example of listAllV5SigTemplates request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listAllV5SigTemplates usn="1" version="1.0">
<Parameters>
</Parameters>
</listAllV5SigTemplates>
</Methods>
</Command>
</CommandList>
```

An example of listAllV5SigTemplates response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listAllV5SigTemplates usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
<ReturnData>
<keylist>DEFAULT</keylist>
<keylist>1</keylist>
<keylist>2</keylist>
<keylist>ETS</keylist>
<keylist>SIG1</keylist>
<keylist>SIG2</keylist>
<keylist>SIG3</keylist>
<keylist>SIG4</keylist>
</ReturnData>
</listAllV5SigTemplates>
</Methods>
</Response>
</CommandList>
```

Operation addV5RingTemplate

An example of addV5RingTemplate request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<addV5RingTemplate usn="1" version="1.0">
<Parameters>
<v5ringid>3</v5ringid>
<std>3</std>
<r01>1</r01>
<r02>1</r02>
<r03>1</r03>
<r04>1</r04>
<r05>1</r05>
<r06>1</r06>
<r07>1</r07>
<r08>1</r08>
<r09>1</r09>
<r10>1</r10>
<r11>1</r11>
<r12>1</r12>
<r13>1</r13>
<r14>1</r14>
<r15>1</r15>
</Parameters>
</addV5RingTemplate>
</Methods>
</Command>
</CommandList>
```

An example of addV5RingTemplate response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<addV5RingTemplate usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
</addV5RingTemplate>
</Methods>
</Response>
</CommandList>
```

Operation deleteV5RingTemplate

An example of deleteV5RingTemplate request

```
<?xml version="1.0"?>
```

```
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<deleteV5RingTemplate usn="1" version="1.0">
<Parameters>
<v5ringid>3</v5ringid>
</Parameters>
</deleteV5RingTemplate>
</Methods>
</Command>
</CommandList>
```

An example of deleteV5RingTemplate response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<deleteV5RingTemplate usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
</deleteV5RingTemplate>
</Methods>
</Response>
</CommandList>
```

Operation listV5RingTemplate

An example of listV5RingTemplate request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listV5RingTemplate usn="1" version="1.0">
<Parameters>
<v5ringid>3</v5ringid>
</Parameters>
</listV5RingTemplate>
</Methods>
</Command>
</CommandList>
```

An example of deleteV5RingTemplate response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
```

```
<listV5RingTemplate usn="1" version="1.0">
<RC>0</RC>
<MsgTxt>Table operation was successful</MsgTxt>
<ReturnData>
<v5ringid>3</v5ringid>
<std>3</std>
<r01>1</r01>
<r02>1</r02>
<r03>1</r03>
<r04>1</r04>
<r05>1</r05>
<r06>1</r06>
<r07>1</r07>
<r08>1</r08>
<r09>1</r09>
<r10>1</r10>
<r11>1</r11>
<r12>1</r12>
<r13>1</r13>
<r14>1</r14>
<r15>1</r15>
</ReturnData>
</listV5RingTemplate>
</Methods>
</Response>
</CommandList>
```

Operation listAllV5RingTemplates

An example of listAllV5RingTemplates request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listAllV5RingTemplates usn="1" version="1.0">
<Parameters>
</Parameters>
</listAllV5RingTemplates>
</Methods>
</Command>
</CommandList>
```

An example of listAllV5RingTemplates response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listAllV5RingTemplates usn="1" version="1.0">
<RC>0</RC>
```

```
<MsgTxt>Table operation was successful</MsgTxt>
<ReturnData>
<keylist>DEFAULT</keylist>
<keylist>3</keylist>
<keylist>TEST1</keylist>
<keylist>TEST</keylist>
</ReturnData>
</listAllV5RingTemplates>
</Methods>
</Response>
</CommandList>
```

Error handling

V5.2 operations can send error message back to client when an operation fails. The error response has a return code with a message text describing the reason.

Example of an error scenario for listAllInterfaces

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>v5CfgMgrIf</Interface>
<Methods>
<listAllIntefaces usn="1" version="1.0">
<RC>8</RC>
<MsgTxt>V5CFGMGR_INVALID_INPUT</MsgTxt>
<ReturnData>
</ReturnData>
</listAllInterfaces>
</Methods>
</Response>
</CommandList>
```

Authorization for V5.2 operations

In addition to users belonging to “succssn” group to login to OSSGate, user need to be in application specific groups to perform specific operations. Each operation is associated with one or more user groups. In order to execute a command, a user must belong to at least one of the associated user groups. The user groups associated with each OSSGate V5.2 provisioning operation is listed in the two tables below.

Commands	User groups				
	trkadm	trkrw	trkmtc	trksprov	trkro
addInterface	X	X			
deleteInterface	X	X			
listAllInterfaces	X	X	X	X	X
listInterface	X	X	X	X	X

Commands	User groups				
	trkadm	trkrw	trkmtc	trksprov	trkro
addV5ProvTemplate	X	X			
deleteV5ProvTemplate	X	X			
listV5ProvTemplate	X	X	X	X	X
listAllV5ProvTemplates	X	X	X	X	X
addV5SigTemplate	X	X			
deleteV5SigTemplate	X	X			
listV5SigTemplate	X	X	X	X	X
listAllV5SigTemplates	X	X	X	X	X
addV5RingTemplate	X	X			
deleteV5RingTemplate	X	X			
listV5RingTemplate	X	X	X	X	X
listAllV5RingTemplates	X	X	X	X	X

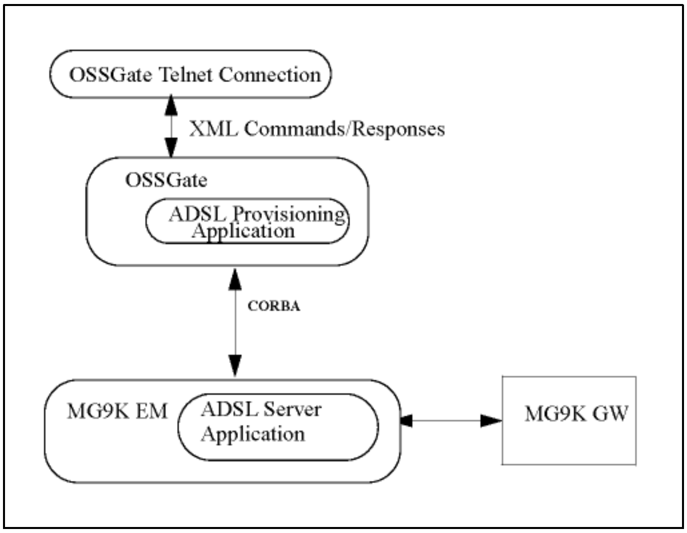
Commands	User groups				
	lnadm	lnrw	lnmtc	lnsprov	lnro
addInterface	X	X			
deleteInterface	X	X			
listAllInterfaces	X	X	X	X	X
listInterface	X	X	X	X	X
addV5ProvTemplate	X	X			
deleteV5ProvTemplate	X	X			
listV5ProvTemplate	X	X	X	X	X
listAllV5ProvTemplates	X	X	X	X	X
addV5SigTemplate	X	X			
deleteV5SigTemplate	X	X			
listV5SigTemplate	X	X	X	X	X
listAllV5SigTemplates	X	X	X	X	X
addV5RingTemplate	X	X			
deleteV5RingTemplate	X	X			
listV5RingTemplate	X	X	X	X	X
listAllV5RingTemplates	X	X	X	X	X

V5.2 return codes

Return code	Meaning
0	Successful operation
1	More data coming
2	Request Rejected due to Resource Limits Reached
3	Application was Commanded to Abort
4	Invalid input from client
5	Failed SERVRINV update or could not get node name and number
6	Failed Add to GWCEM
7	Failed to update NV
8	Failed Rollback of ADD GWC
9	Query GWC Operation Failed to Get Data from GWC EM
10	Query GWC Operation Failed to Read GWC List from the Network View

ADSL flow through provisioning for MG 9000

The ADSL flow through provisioning application enables authenticated OSS clients to perform provisioning operations on ADSL data circuits on MG 9000 through MG 9000 Manager from CS 2000 Management Tools server. The provisioning operations include querying, adding, modifying and deleting ADSL subscriber data. Access to this functionality is by connection to OSSGate.



Supported ADSL provisioning commands

The following XML commands are supported for ADSL provisioning using OSSGate:

- GetSubscriber
- AddSubscriber
- AddCrossConnection
- ModifySubscriber
- ModifyCrossConnection
- DeleteSubscriber
- DeleteCrossConnection

Terminology and description of method parameters

- VMGName - Name of the media gateway. TYPE string
  - Format: <SITEname 4 characters><officeframe=000-511> -<logicalframe=0-7>-<shelf=0-3>.
  - Example: LAKE017-6-2

- TerminationPoint - Name of the termination point. This parameter identifies the ADSL data circuit that is being provisioned. TYPE string
  - — Format tp/<slot=02-21, 0-padded>/<circuit=0-31, 0-padded>
  - — Example tp/02/03.
- version - The version refers to the version of the method or operation - e.g., the AddSubscriber operation may have version 1.0 and 2.0, where version 1.0 takes parameters X and Y and version 2.0 takes parms X, Y, and Z. TYPE decimal.
- usn - Unique Sequence Number. This parameter is introduced to facilitate sending multiple XML commands in one file. This parameter value will be used to match the responses with the requests. TYPE integer.
- RC - Return Code. A numeric value indicating the result of the attempted operation. TYPE short.
- MsgTxt - Appropriate text that describes the result of the operation. TYPE string.
- DownMaxSpeed - The value in kbits, Max DownStream 32 - 13376, multiple of 32. TYPE positiveInteger.
- UpMaxSpeed - The value in kbits, Max Upstream 32 - 1440, multiple of 32. TYPE unsignedShort.
- DownMaxInterleaveDelay - The value in milliseconds - DownStream delay 10 - 255. TYPE unsignedShort.
- UpMaxInterleaveDelay - The value in milliseconds - Upstream delay 10 - 255. TYPE unsignedShort.
- DownSignalNoiseMargin - The value in dB - DownStream noise margin 6 - 31. TYPE unsignedShort.
- UpSignalNoiseMargin - The value in dB - Upstream noise margin 6 - 31. TYPE unsignedShort.
- TransmissionMode - Possible values are “Auto Mode”, “ANSI”, “G.DMT”. Default value is “Auto Mode”. TYPE string.

Each ADSL data circuit can provide up to 8 cross connections and each cross connection require the following parameters data filled:

- VPI - Virtual Path Identifier. Range 16 - 31. TYPE unsignedShort.
- VCI - Virtual Circuit Identifier. Range 33 - 2047. TYPE unsignedShort.
- UpStreamTrafficDescriptor - Name of the traffic descriptor user label for transmission. TYPE string.

- DownStreamTrafficDescriptor - Name of the traffic descriptor user label for reception. TYPE string.
- State - State of the cross connection. Possible values are “Inactive”, “Active”. TYPE string.

Description of method parameters

- GetSubscriber - Interface/Method that gets the configuration data for a subscriber that includes the cross connection data associated with the specified MG and the termination point.
  - Input data
    - usn
    - version
    - VMGName
    - TerminationPoint
  - Output data
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - DownMaxSpeed - The value in kbits, Max DownStream 32 - 13376, multiple of 32
    - UpMaxSpeed - The value in kbits, Max Upstream 32 - 1440, multiple of 32.
    - DownMaxInterleaveDelay - The value in milliseconds - DownStream delay 10 - 255
    - UpMaxInterleaveDelay - The value in milliseconds - Upstream delay 10 - 255
    - DownSignalNoiseMargin - The value in dB - DownStream noise margin 6 - 31
    - UpSignalNoiseMargin - The value in dB - Upstream noise margin 6 - 31
    - TransmissionMode - Possible values are “Auto Mode”, “ANSI”, “G.DMT”.
The following data will be returned for each of the 8 cross connections.
      - VCC.ID - Virtual Cross Connection Identifier. The value is range is 1-8 both inclusive.
      - VPI - Virtual Path Identifier. Range 16 - 31.

        - VCI - Virtual Circuit Identifier. Range 33 - 2047.
        - UpStreamTrafficDescriptor - Name of the traffic descriptor user label for transmission.
        - DownStreamTrafficDescriptor - Name of the traffic descriptor user label for reception.
        - State - State of the cross connection. Possible values are “Inactive”, “Active”, or “De-provision”.
    - RC - Return Code
    - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

- AddSubscriber - Interface/Method that will add a new subscriber with the given configuration data. Using this method, multiple cross connections can be configured in one transaction. This method has the following parameters:
  - Input data
    - usn
    - version
    - VMGName
    - TerminationPoint
    - Config data (applies to all cross connections)
      - DownMaxSpeed - The value in kbits, Max DownStream 32 - 13376, multiple of 32
      - UpMaxSpeed - The value in kbits, Max Upstream 32 - 1440, multiple of 32.
      - DownMaxInterleaveDelay - The value in milliseconds - DownStream delay 10 - 255
      - UpMaxInterleaveDelay - The value in milliseconds - Upstream delay 10 - 255
      - DownSignalNoiseMargin - The value in dB - DownStream noise margin 6 - 31
      - UpSignalNoiseMargin - The value in dB - Upstream noise margin 6 - 31
      - TransmissionMode - Possible values are “Auto Mode”, “ANSI”, “G.DMT”. Default value is the ‘Auto Mode”.

> ***Note:*** The following data will be specified for each of the 8 cross connections that need to be configured. Typically, a single cross connection is configured per subscriber. At least one cross connection has to be data filled to add a new subscriber.

- Required data for each cross connection
  - VCC.ID - Virtual Cross Connection Identifier. The value is between 1-8 both inclusive.
  - VPI - Virtual Path Identifier. Range 16 - 31.
  - VCI - Virtual Circuit Identifier. Range 33 - 2047.
  - UpStreamTrafficDescriptor - Name of the traffic descriptor user label for transmission.
  - DownStreamTrafficDescriptor - Name of the traffic descriptor user label for reception.
  - State - State of the cross connection. Possible values are “Inactive”, “Active”.

— Output data

- usn (value should be the same as the input)
- version (value should be the same as the input)
- RC - Return Code
- MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

• AddCrossConnection - Interface/Method that will add a new subscriber with the specified cross connection data. This method has the following parameters:

> ***Note:*** Typically, a single cross connection is configured per subscriber.

— Input data

- usn
- version
- VMGName
- TerminationPoint
- VCC.ID - Virtual Cross Connection Identifier. The value range is 1-8 both inclusive.

- VPI - Virtual Path Identifier. Range 16 - 31.
- VCI - Virtual Circuit Identifier. Range 33 - 2047.
- UpStreamTrafficDescriptor - Name of the traffic descriptor user label for transmission.
- DownStreamTrafficDescriptor - Name of the traffic descriptor user label for reception.
- State - State of the cross connection. Possible values are “Inactive”, “Active”.

— Output data

- usn (value should be the same as the input)
- version (value should be the same as the input)
- RC - Return Code
- MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

- ModifySubscriber - Interface/Method that will modify an existing subscriber with the given configuration data without having to delete and re-add the service. Using this method, multiple cross connection data can also be modified in one transaction. This method has the following parameters:

— Input data

- usn
- version
- VMGName
- TerminationPoint

> ***Note:*** Zero or more of the following config parameters that need to be modified are specified as the input data. This configuration data applies to all cross connections in the circuit.

- DownMaxSpeed - The value in kbits, Max DownStream 32 - 13376, multiple of 32
- UpMaxSpeed - The value in kbits, Max Upstream 32 - 1440, multiple of 32.
- DownMaxInterleaveDelay - The value in milliseconds - DownStream delay 10 - 255

    - UpMaxInterleaveDelay - The value in milliseconds - Upstream delay 10 - 255
    - DownSignalNoiseMargin - The value in dB - DownStream noise margin 6 - 31
    - UpSignalNoiseMargin - The value in dB - Upstream noise margin 6 - 31
    - TransmissionMode - Possible values are “Auto Mode”, “ANSI”, “G.DMT”.

      Note: The following data will be specified for each of the 8 cross connections that need to be modified. Only the data that needs to be modified is specified for each cross connection. Zero or more of the following are specified as input data for each cross connection. For any cross connection data that needs to be modified, the VCC.ID is the required parameter to identify the VCC.

    - VCC.ID - Virtual Cross Connection Identifier. The value is between 1-8 both inclusive.
    - VPI - Virtual Path Identifier. Range 16 - 31.
    - VCI - Virtual Circuit Identifier. Range 33 - 2047.
    - UpStreamTrafficDescriptor - Name of the traffic descriptor user label for transmission.
    - DownStreamTrafficDescriptor - Name of the traffic descriptor user label for reception.
    - State - State of the cross connection. Possible values are “Inactive”, “Active”.
  - Output data
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - RC - Return Code
    - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

- ModifyCrossConnection - Interface/Method that will modify an existing subscriber with the given cross connection data, without having to delete and re-add the service. This method has the following parameters:
  - Input data
    - usn

- version
- VMGName
- TerminationPoint
- VCC.ID - Virtual Cross Connection Identifier. The value is between 1-8 both inclusive.

  Note: Only the data that needs to be modified is specified for the cross connection. One or more of the following are specified as input data.

- VPI - Virtual Path Identifier. Range 16 - 31.
- VCI - Virtual Circuit Identifier. Range 33 - 2047.
- UpStreamTrafficDescriptor - Name of the traffic descriptor user label for transmission.
- DownStreamTrafficDescriptor - Name of the traffic descriptor user label for reception.
- State - State of the cross connection. Possible values are “Inactive”, “Active”.

— Output data

- usn (value should be the same as the input)
- version (value should be the same as the input)
- RC - Return Code
- MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

- DeleteSubscriber - Interface/Method that deletes the specified subscriber data.

— Input data

Note: All the cross connection data and the circuit configuration data are deleted. To add a new subscriber for this circuit will require specifying the optional configuration parameters as well as the cross connection parameters using the ‘AddSubscriber’ method.

- usn
- version
- VMGName
- TerminationPoint

  - Output data
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - RC - Return Code
    - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

- DeleteCrossConnection - Interface/Method that deletes the specified cross connection data.

  Note: All the cross connection data is deleted. To add cross connection back, use the ‘AddCrossConnection’ method.

  - Input data
    - usn
    - version
    - VMGName
    - TerminationPoint
    - VCC.ID - Virtual Cross Connection Identifier. The value is between 1-8 both inclusive.
  - Output data
    - usn (value should be the same as the input)
    - version (value should be the same as the input)
    - RC - Return Code
    - MsgTxt - Text (string) detailing specific results of the operation in terms of Success or Failure of the operation and possible indication of failure reason.

XML commands

The following are examples of XML commands and the corresponding response messages for each operation supported for ADSL flow through provisioning. The XML coding in these examples is formatted for ease of understanding.

An example of GetSubscriber request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>AdslCircuitProvisioningManager<Interface>
<Methods>
```

```
<GetSubscriber usn="1" version="1.0">
<Parameters>
<VMGName>LAKE017-6-2</VMGName>
<TerminationPoint>tp/02/03</TerminationPoint>
</Parameters>
</GetSubscriber>
</Methods>
</Command>
</CommandList>
```

An example of GetSubscriber response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<GetSubscriber usn="1" version="1.0">
<ReturnData>
<VMGName>LAKE017-6-2</VMGName>
<TerminationPoint>tp/02/03</TerminationPoint>
<UpMaxSpeed>1440</UpMaxSpeed>
<DownMaxSpeed>13376</DownMaxSpeed>
<UpSignalNoiseMargin>6</UpSignalNoiseMargin>
<DownSignalNoiseMargin>6</DownSignalNoiseMargin>
<UpMaxInterleaveDelay>10</UpMaxInterleaveDelay>
<DownMaxInterleaveDelay>10</DownMaxInterleaveDelay>
<TransmissionMode>Auto Mode</TransmissionMode>
<VCC ID=”1‟ VPI=”0” VCI=”0”
UpStreamTrafficDescriptor=”userTDLabel1"
”DownStreamTrafficDescriptor=”userTDLabel1” State=”Active
”/>
<VCC ID=”2‟ VPI=”1” VCI =”2” UpStreamTrafficDescriptor=
”userTDLabel2” DownStreamTrafficDescriptor=
”userTDLabel2” State=”Active”/>
<RC>0</RC>
<MsgTxt>GetSubscriber operation was successful.</MsgTxt>
</ReturnData>
</GetSubscriber>
</Methods>
</Response>
</CommandList>
```

AddSubscriber command

An example of AddSubscriber (with 2 cross connections) request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<AddSubscriber usn="1" version="1.0">
```

```
<Parameters>
<VMGName>LAKE017-6-2</VMGName>
<TerminationPoint>tp/02/03</TerminationPoint>
<UpMaxSpeed>1440</UpMaxSpeed>
<DownMaxSpeed>13376</DownMaxSpeed>
<UpMaxInterleaveDelay>10</UpMaxInterleaveDelay>
<DownMaxInterleaveDelay>10</DownMaxInterleaveDelay>
<UpSignalNoiseMargin>6</UpSignalNoiseMargin>
<DownSignalNoiseMargin>6</DownSignalNoiseMargin>
<TransmissionMode>Auto Mode</TransmissionMode>
<VCC>
<ID>1</ID>
<VPI>21</VPI>
<VCI>40</VCI>
<UpStreamTrafficDescriptor>upTD</
UpStreamTrafficDescriptor>
<DownStreamTrafficDescriptor>dnTD</
DownStreamTrafficDescriptor>
<State>Active</State>
</VCC>
<VCC>
<ID>2</ID>
<VPI>21</VPI>
<VCI>400</VCI>
<UpStreamTrafficDescriptor>upTD</
UpStreamTrafficDescriptor>
<DownStreamTrafficDescriptor>dnTD</
DownStreamTrafficDescriptor>
<State>Inactive</State>
</VCC>
</Parameters>
</AddSubscriber>
</Methods>
</Command>
</CommandList>
```

An example of AddSubscriber response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<AddSubscriber usn="1" version="1.0">
<ReturnData>
<RC>0</RC>
<MsgTxt>AddSubscriber operation was successful.</MsgTxt>
</ReturnData>
</AddSubscriber>
</Methods>
</Response>
```

```
</CommandList>
```

AddCrossConnection command

An example of AddCrossConnection request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<AddCrossConnection usn="1" version="1.0">
<Parameters>
<VMGName>LAKE017-6-2</VMGName>
<TerminationPoint>tp/02/03</TerminationPoint>
<VCC>
<ID>8</ID>
<VPI>21</VPI>
<VCI>500</VCI>
<UpStreamTrafficDescriptor>upTD</
UpStreamTrafficDescriptor>
<DownStreamTrafficDescriptor>dnTD</
DownStreamTrafficDescriptor>
<State>Active</State>
</VCC>
</Parameters>
</AddCrossConnection>
</Methods>
</Command>
</CommandList>
```

An example of AddCrossConnection response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<AddCrossConnection usn="1" version="1.0">
<ReturnData>
<RC>0</RC>
<MsgTxt>AddCrossConnection operation was successful.</M
sgTxt>
</ReturnData>
</AddCrossConnection>
</Methods>
</Response>
</CommandList>
```

ModifySubscriber command

An example of ModifySubscriber request

```
<?xml version="1.0"?>
```

```
<CommandList>
<Command>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<ModifySubscriber usn="1" version="1.0">
<Parameters>
<VMGName>LAKE017-6-2</VMGName>
<TerminationPoint>tp/02/03</TerminationPoint>
<UpMaxInterleaveDelay>15</UpMaxInterleaveDelay>
<DownSignalNoiseMargin>8</DownSignalNoiseMargin>
<VCC>
<ID>1</ID>
<VPI>31</VPI>
<VCI>2047</VCI>
<State>Inactive</State>
</VCC>
<VCC>
<ID>2</ID>
<State>Active</State>
</VCC>
</Parameters>
</ModifySubscriber>
</Methods>
</Command>
</CommandList>
```

An example of ModifySubscriber response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<ModifySubscriber usn="1" version="1.0">
<ReturnData>
<RC>0</RC>
<MsgTxt>ModifySubscriber operation was successful.</MsgT
xt>
</ReturnData>
</ModifySubscriber>
</Methods>
</Response>
</CommandList>
```

ModifyCrossConnection command

An example of ModifyCrossConnection request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
```

```
<ModifyCrossConnection usn="1" version="1.0">
<Parameters>
<VMGName>LAKE017-6-2</VMGName>
<TerminationPoint>tp/02/03</TerminationPoint>
<VCC>
<ID>8</ID>
<VPI>30</VPI>
<VCI>300</VCI>
<State>Active</State>
</VCC>
</Parameters>
</ModifyCrossConnection>
</Methods>
</Command>
</CommandList>
```

An example of ModifyCrossConnection response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<ModifyCrossConnection usn="1" version="1.0">
<ReturnData>
<RC>0</RC>
<MsgTxt>ModifyCrossConnection operation was successful.</M
sgTxt>
</ReturnData>
</ModifyCrossConnection>
</Methods>
</Response>
</CommandList>
```

DeleteSubscriber command

An example of DeleteSubscriber request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<DeleteSubscriber usn="1" version="1.0">
<Parameters>
<VMGName>LAKE017-6-2</VMGName>
<TerminationPoint>tp/02/03</TerminationPoint>
</Parameters>
</DeleteSubscriber>
</Methods>
</Command>
</CommandList>
```

An example of DeleteSubscriber response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<DeleteSubscriber usn="1" version="1.0">
<ReturnData>
<RC>0</RC>
<MsgTxt>DeleteSubscriber operation was successful.</MsgT
xt>
</ReturnData>
</DeleteSubscriber>
</Methods>
</Response>
</CommandList>
```

DeleteCrossConnection command

An example of DeleteCrossConnection request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<DeleteCrossConnection usn="1" version="1.0">
<Parameters>
<VMGName>LAKE017-6-2</VMGName>
<TerminationPoint>tp/02/03</TerminationPoint>
<VCC>
<ID>1</ID>
</VCC>
</Parameters>
</DeleteCrossConnection>
</Methods>
</Command>
</CommandList>
```

An example of DeleteCrossConnection response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<DeleteCrossConnection usn="1" version="1.0">
<ReturnData>
<RC>0</RC>
<MsgTxt>DeleteCrossConnection operation was successful.</M
sgTxt>
</ReturnData>
</DeleteCrossConnection>
```

```
</Methods>
</Response>
</CommandList>
```

User authorization for ADSL provisioning operations

In addition to users belonging to “succssn” group to login to OSSGate, user needs to be in application specific groups to perform specific operations. Each operation is associated with one or more user groups. To execute a command, user must belong to at least one of the associated user groups. The user groups associated with ADSL flow through provisioning operations are detailed in the table below.

Command	User Group				
	Inadm	Inrw	Inmtc	Insprov	Inro
getSuscriber	X	X	X	X	X
addSubscriber	X	X		X	
addCrossConnection	X	X		X	
modifySubscriber	X	X		X	
modifyCrossConnection	X	X		X	
deleteSubscriber	X	X		X	
deleteCrossConnection	X	X		X	

An example of authorization failure response

```
<?xml version="1.0" encoding="UTF-8"?>
<CommandList>
<Response>
<Interface>AdslCircuitProvisioningManager</Interface>
<Methods>
<GetSubscriber usn="1" version="1.0">
<RC>501</RC>
<MsgTxt>Insufficient security privileges to perform this
action</MsgTxt>
</ReturnData>
</GetSubscriber>
</Methods>
</Response>
</CommandList>
```

Return codes

Return code	MsgText
0	Request Successful.
1	Operation failure. The name of ADSL circuit received is invalid or the circuit can not be identified.
2	Add Operation Failure. ATM index not found in interface MIB table
3	Modify Operation Failure. ATM index not found in interface MIB table
4	Delete Operation Failure. ATM index not found in interface MIB table
5	Query request failed.
501	Insufficient security privileges to perform this action.

Limitations and restrictions

Step Action

1 Endpoint names must include the VMG name and the termination point.

2 The VMG name must conform to the following format:

- SITEXXX-Y-Z - Example: LAKE017-6-2
  XXX is the frame number within the office (000 to 511 both inclusive)
- Y is the logical frame number within the physical MG (0 to 7 both inclusive)
- Z is the shelf number (0 to 3 both inclusive)

3 The termination point must conform to the following format:

- tp/AA/BB - Example tp/02/04
- AA is the card number (02-09 (both inclusive) and 14-21 (both inclusive))
- BB is circuit number (00 to 31 both inclusive)

4 Abort operation is not supported. The ability to abort an XML request received from OSSGate is not supported.

—End—

Batch Provisioning Tool

The Batch Provisioning Tool (BPT) is a command line user interface. It uses OSSGate as the backend to perform all batch operations. BPT command line user interface allows users to batch provision line and ADSL circuit commands. The BPT supports the following functionality:

- Execute Line and ADSL batch operation
- Execute Line endpoint provisioning for third party large line gateways
- View/Delete Batch operation results
- View/Delete Batch operation logs

Prerequisite to using the Batch Provisioning Tool

- User must be logged in to the CS 2000 Management Tools server.
- User must belong to group “succssn” to start the Batch Provisioning Tool.
- User must have appropriate authorization to execute the required line provisioning or ADSL or line endpoint provisioning commands.
- Batch provisioning input files must be generated by user before using the tool.

Using the Batch Provisioning Tool

After successfully logging in to the CS 2000 Management Tool server, user can type “bpt” from any directory level. Once the user has successfully authenticated, the main menu is displayed. A typical login session is as follows.

Example of BPT login:

```
$bpt
BPT Main Menu
----------------------------------------------
==================================
    Batch Provisioning Tool (BPT V1.0)
==================================
    Username: tom Password:
    Login in progress...
    You are currently logged in as: tom!
  ========== Main Menu: ==========
       (1) Execute Batch File
       (2) Display Output
       (3) Display Logs
       (4) Delete Output or Log Files
(5) Configuration
       (h) Help
       (x) Exit
```

```
   Selection: [1/2/3/4/h/x:1]
   -------------------------------------------------
```

Main Menu:

Option (1) Execute Batch File permits user to execute batch provisioning commands. When option (1) is selected, the “Provisioning Input Entry Menu” will be displayed as shown in section "Provisioning input entry menu" (page 326).

Option (2) Display Output permits user to view the provisioning output results. The output files are in a text format for line provisioning and in XML format for ADSL provisioning.

Option (3) Display Logs is permits user the view the log files that are associated to the provisioned batch operation.

Option (4) Delete Output or Log Files permits user to remove any unnecessary output or log files.

Option (5) enables the user to set the BPT for Stop Mode on or off, whereby batch provisioning can be configured to stop after an error has occurred.

Option (h) Help provides a generic operation help using BPT.

Option (x) Exit permits user to exit out the BPT.

Provisioning input entry menu

When Option (1) Execute Batch File of the BPT Main Menu is selected, BPT prompts the following menu.

```
Provisioning Input Entry Menu
------------------------------------------------
==============================
Provisioning Input Entry Menu:
==============================
       (1) Lines
       (2) ADSL
       (3) Line Endpoints
       (4) Go to shell prompt
       (r) Return to the main menu.
       (x) Exit BPT.
Selection:  [1/2/3/4/r/x:1]
-----------------------------------------------
```

Option (1) Lines permits user to do lines batch provisioning.

Option (2) ADSL permits user to do the ADSL lines batch provisioning.

Option (3) Line Endpoints permits user to perform batch provisioning on line endpoints. The following sub-menu displays.

```
================================
Line Endpoints Sub-Menu:
================================
       (1) Provision third party large line endpoints
       (2) Export 'Query Line Endpoints' output
       (b) Back to the previous menu
       (r) Return to the main menu
       (x) Exit BPT
Selection:  [1/2/b/r/x:1]1
```

If "1" is selected from the above "Line Endpoint Sub-Menu," or if the option 1 or 2 is selected from the "Provisioning Input Entry Menu" menu, the BPT displays the following prompt:

```
Which mode would you like to have the log file on?
Critical/Verbose/Major/Minor [c/v/ma/mi:c]:c
Please enter the input file name:
```

> ***Note:*** The data needed for batch provisioning must be generated by user. The input file should be available with proper permissions for use in the CS 2000 Management Tools server. The files can be located in any directory. The absolute path of the input file must be entered. Sample input file templates are located at: /opt/nortel/NTsesm/tools/bpt/templates

After the input file name is entered, the tool processes the batch. For example, after entering the line batch provisioning command file “linesBatch,” the BPT displays the following response:

```
----------------------------------------------
Please wait while processing the batch commands...
........
Provisioning is complete!  The output file is located at:
/opt/nortel/NTsesm/tools/bpt/output/lines/
linesBatch_021002_114504.out
----------------------------------------------
```

The BPT then displays the Main Menu as in Section "Provisioning input entry menu" (page 326) again for further selection.

Option (r) "Return to the main menu" allows the user to return to the Main Menu in section "Batch Provisioning Tool" (page 325).

Option (x) "Exit" allows the user to exit the BPT.

Option (4) "Go to shell prompt" of the Provisioning Input Entry Menu enables the user to execute UNIX commands on the command line. Type “exit” to return to the “Provisioning Input Entry Menu.”

Line Endpoints Sub-Menu

If Option (3) is selected from the Provisioning Input Entry Menu, the following submenu displays:

```
================================
Line Endpoints Sub-Menu:
================================
       (1) Provision third party large line endpoints
       (2) Export 'Query Line Endpoints' output
       (b) Back to the previous menu
       (r) Return to the main menu
       (x) Exit BPT
  Selection: [1/2/b/r/x:1]2
```

Option 2 takes the output XML file from one of the queryLineEndpoints or queryAllLineEndpoints command as input and generates an XML suitable for addLineEndpoints operation. The exported file is available at the following location:

/opt/nortel/NTsesm/tools/bpt/output/export

The generated output file name has the same format as the other commands, which is the input file name with appended current time stamp. The contents of the generated file match the format of the addLineEndpoint XML schema. See addLineEndpoint examples in the earlier section for the details.

Display Output menu

When Option (2) Display Output File of the BPT Main Menu is selected, the “Display Output Menu” displays as follows.

```
Display Output Menu
---------------------------------
====================
  Display Output Menu:
====================
       (1) Lines
       (2) ADSL
       (3) Line Endpoints
       (4) Go to shell prompt
       (r) Return to the main menu.
       (x) Exit BPT.
Selection:  [1/2/3/4/r/x:1] 3
---------------------------------------
```

Option (1) lines permits user to view lines batch provisioning output.

Option (2) ADSL permits user to view the ADSL lines batch provisioning output.

Option (3) for Line endpoints displays the following sub-menu.

```
================================
Line Endpoints Sub-Menu:
================================
       (1) Provision third party large line endpoints
       (2) Export 'Query Line Endpoints' output
       (b) Back to the previous menu
       (r) Return to the main menu
       (x) Exit BPT
  Selection: [1/2/b/r/x:1] 1
```

If the user selects Option (1) for Lines, Option (2) for ADSL output or any of the sub-options from the "Line Endpoints Sub-Menu," BPT displays the output files that are currently in the output directory for the selected option. For example, if the user selected option (1) Lines, BPT displays all the output files that are currently in the lines output directory. The following example lists the line provisioning output files.

```
----------------------------------------------
The following files are listed in the output directory:
linesBatch_020930_154219.out
linesBatch_021001_131924.out
linesBatch_021001_111451.out
linesBatch_021002_114504.out
Please enter the file name:
---------------------------------------------
```

Enter the file name. BPT displays the file content. Use space bar to view the next page. Type “q” at any time to quit without having to finish viewing the file.

After viewing one output file, view other available files by returning to the previous menu. For example, the following menu displays after the user finishes viewing a line provisioning output file.

```
Display Line Output File Menu
----------------------------------------------
========================================
Display lines output file menu:
========================================
       (1) View another lines output File
       (b) Back to the previous menu.
       (r) Return to the main menu.
       (x) Exit BPT.
Selection:  [1/b/r/x:1]
```

> ***Note:*** The CS 2000 Management Tools provides a GUI-based application named "Batch configuration Monitor" that displays the ADSL provisioning output files and the "Line Endpoints" provisioning output files without the XML tags in a readable form.

Display Log menu

When Option (3) Display Logs is selected from the BPT Main Menu, the following menu displays.

```
======================
Display Log File Menu:
======================
       (1) Lines
       (2) ADSL
       (3) Line Endpoints
       (4) Go to shell prompt
       (r) Return to the main menu.
       (x) Exit BPT.
Selection:  [1/2/3/4/r/x:1]
----------------------------------------
```

This menu operates similar to “Display Output Menu” in section Line endpoint provisioning sub-menu. The only difference is that it allows the user to view log files instead of the provisioning output files.

Delete Output or Log files menu

When Option (4) Delete Output or Log Files is selected from the BPT Main Menu, the following menu is displayed.

```
Delete Output or Log File Menu
-------------------------------------
==================
Delete Files Menu:
==================
       (1) Lines
       (2) ADSL
       (3) Line Endpoints
       (4) Go to shell prompt
       (r) Return to the main menu.
       (x) Exit BPT.
  Selection: [1/2/3/4/r/x:1]
----------------------------------------
```

Option (1) Lines permits users to delete the log or output files for lines provisioning.

Option (2) ADSL permits users to delete the log or output files for ADSL lines provisioning.

Option (3) Line Endpoints permits users to delete the log or output files from any of the "Line Endpoints" operations. Option Line Endpoints has the same sub-menu options as in the previous sections to work with the output or log files for both the provisioning and the export commands. The sub-menu is displayed as follows:

```
================================
Line Endpoints Sub-Menu:
```

```
================================
       (1) Provision third party large line endpoints
       (2) Export 'Query Line Endpoints' output
       (b) Back to the previous menu
       (r) Return to the main menu
       (x) Exit BPT
Selection:  [1/2/b/r/x:1]
```

Option (1) or Option (2) menus from the above sub-menu are displayed as follows:

```
===========================
Delete line_ep File Menu:
===========================
       (1) Delete line_ep Output Files
       (2) Delete line_ep Log Files
       (3) Go to shell
       (b) Back to the previous menu.
       (r) Return to the main menu.
       (x) Exit BPT.
Selection:  [1/2/3/b/r/x:1]
       (1) Delete export Output Files
       (2) Delete export Log Files
       (3) Go to shell
       (b) Back to the previous menu.
       (r) Return to the main menu.
       (x) Exit BPT.
  Selection: [1/2/3/b/r/x:1]
```

When Option (1) or Option (2) is selected from the above sub-menus, the menu flow matches the "lines" or "ADSL" as in the later section.

All the commands work in the same way. An example of "lines" command is explained below.

When the Option (1) from the "Delete Files Menu" is selected, the following menu is displayed.

Delete lines file menu

```
--------------------------------------------
===========================
Delete lines File Menu:
===========================
       (1) Delete lines Output Files
       (2) Delete lines Log Files
       (3) Go to shell
       (b) Back to the previous menu.
       (r) Return to the main menu.
       (x) Exit BPT.
  Selection: [1/2/3/b/r/x:1]
```

Option (1) permits user to delete output files for the selected provisioning category. Option (2) permits user to delete the log files for the selected provisioning category.

When option (1) or (2) is selected, BPT displays the output files that are in the output directory for that provisioning category. An example, listing when "delete lines output files" is selected is listed below.

```
------------------------------------------------
The following files are listed in the specified directory:
  linesBatch_020930_154219.out
linesBatch_021001_131924.out
linesBatch_021001_111451.out
linesBatch_021002_114504.out
Please enter the file name(s) that you want to delete (you
can enter more than 1 file using the space as a separator:
--------------------------------------------------
```

The user enters the output file name for it to be deleted. It operates the same way if the user deletes one or more log files. The command returns a confirmation message on files that were deleted.

> ***Note:*** In lines batch, each line represent a SERVORD+ command. This is valid even when the line length exceeds 75 characters. There is no need to add "+" when the commands exceed 75 characters while using BPT.

Configuration menu

When Option (5) Configuration is selected from the BPT Main Menu, the following menu is displayed.

```
==================
Configuration Menu:
==================
       (1) Set Stop Mode
       (r) Return to the main menu.
       (x) Exit BPT.
  Selection: [1/r/x:1]
----------------------------------------
```

This menu allows the user to set the BPT for Stop Mode On or Off, whereby batch provisioning can be configured to stop after an error has occurred. Enabling the Stop Mode halts the batch provisioning process when an error is encountered. When OSSGate provisioning stops due to this condition, the BPT returns to the BPT Main Menu.

On the other hand, if Stop Mode is set to Off, the batch provisioning process ignores the error and provisioning continues through completion. The default value is Stop Mode Off.

The Stop Mode option applies only to XML-based provisioning commands. It does not apply to SERVORD+ commands.

To change the Stop Mode, select option (1) Set Stop Mode during display of the Configuration Menu. The following prompt displays.

```
Provisioning Stop Mode On/Off?[on/off:off]:
```

Since Off is the default value, enter On to enable the Stop Mode. The only acceptable input for this prompt is On or Off. Entering an invalid entry displays the following error response.

```
Undefined selection...please try again.
Provisioning Stop Mode On/Off?[on/off:off]:
```

Option (r) returns the program to the BPT Main Menu. Option (x) exits the BPT.

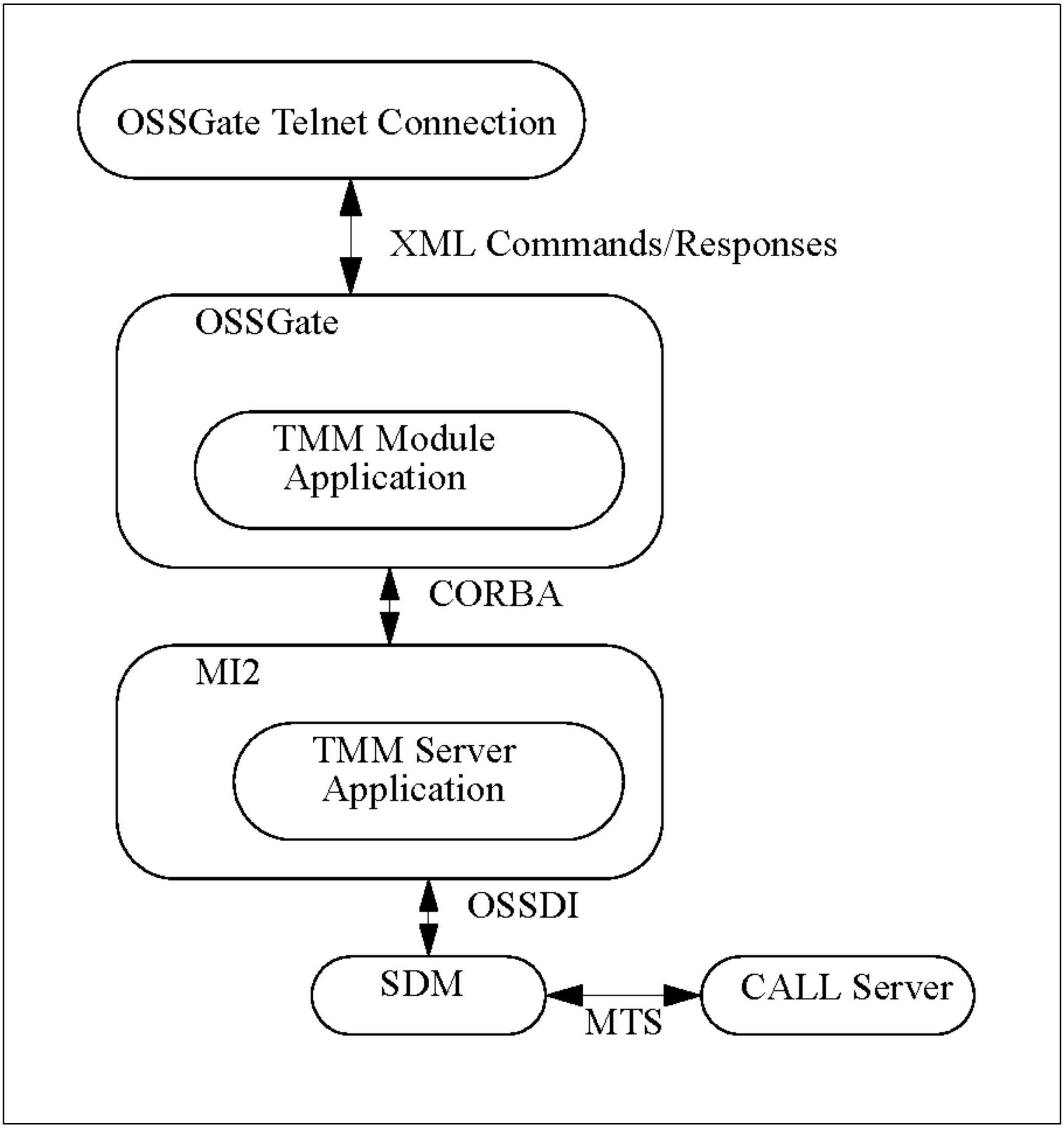
Third party gateway provisioning limitations

Conversion of existing third party large line GW data to (I)SN08 or higher format is not supported. To use the third party gateway provisioning interface feature for the existing GW and the associated end-points, service must be de-provisioned and re-added in (I)SN08 Both new and old format data will be allowed to co-exist in (I)SN08 and higher releases. However, the existing GW and the end-points will be managed in the same format as defined in pre-(I)SN08 releases.

The capabilities can not be fully utilized until all the components of the OAMP are at the base (I)SN08 or higher load line up because here is a dependency on the new LGRPTYPE in the Core that is used by the third party GW nodes. The GWC devices are required to be on the 08 release or higher as well to support spare profiles.

Trunk maintenance with OSSGate

The TMM application enables authenticated OSS clients to perform full maintenance capability for ISUP, PRI, and PTS trunks, as well as query all end point states in a trunk gateway. The maintenance operations include QES, POST, BSY, RTS, BSY INB, FRLS trunks in a trunk gateway. Access to this functionality is by connection to OSSGate.



Supported TMM commands

The following XML commands are supported for TMM using OSSGate:

- PostByGatewayName
- QESByGatewayName
- BSYByGatewayName
- RTSByGatewayName
- INBByGatewayName
- FRLSByGatewayName
- PostByTrunkCLLI
- BSYByTrunkCLLI
- INBByTrunkCLLI
- RTSByTrunkCLLI
- FRLSByTrunkCLLI
- ICOTTest
- PostGroupDChannelByTrunkCLLI
- RTSDChannelByTid
- INBDChannelByTid
- BSYDChannelByTid
- PostByCarrier
- QESByCarrier
- BSYByCarrier
- INBByCarrier
- RTSByCarrier
- FRLSByCarrier
- GetTrunkCllisByGatewayName

Terminology and description of method parameters

- GatewayName - Name of the media gateway. TYPE string. Format: Min length 1; Max length 32
- EndpointRange - Range of maintained endpoints. TYPE string. Format: [Start member number [,-End member number]]* [Start member number [-[End member number]]]
  - Examples
    - "1" - Post trunk 1

  - “1,2” - Post trunk 1 and 2
  - “1-3” - Post trunks 1 through 3 inclusive
  - “1, 3-5” - Post trunks 1 and 3-5 inclusive
  - “1, 3-” - Post trunks 1, and post the rest of the trunk group starting with trunk member 3.

- TrunkMembers - Range of maintained trunk members. TYPE string. Format: same as EndpointRange
- CMCLLI - CLLI Name of Call Server. TYPE string. Format: Min length 1; Max length 32
- TrunkCLLI - CLLI Name of trunks. TYPE string. Format: Min length 1; Max length 16
- GWCName - Name of the gateway controller. TYPE string. Format: Min length 1; Max length 32
- CarrierNames - Name of the carriers. TYPE string. Format: Min length 1
- ShowDetails - Whether show the details of trunks. TYPE boolean.
- FilterState - The valid states of trunks. TYPE string. The valid values include "ALL", "CPD", "IDL", "MB", "NEQ", "INB", "NMB", "PMB", "RMB", "SB", "CPB", "CFL", "LO", "DEL", "INI", "RES", "SZD", "DMB", "DFL", "INS", "STB" and "UNKNOWN".
- NodeNumber - The Node No. TYPE string.
- Value - The value of trunks’ state. TYPE string. The valid values are the same as FilterState except "ALL".
- Count - The number of trunks in the state. TYPE non negative integer.
- State - The current state for the trunk. TYPE string. Min length 2, Max length 5. The valid values are the same as FilterState except "ALL".
- ConnectedTo - What the trunk is connected to. TYPE string.
- TrunkDirection - The trunk direction. TYPE string.
- TrunkSignaling - The trunk signaling. TYPE string.
- PMType - The PM type. TYPE string. The valid values include "SPM" and "GWC_Node."
- PMNumber - The PM number. TYPE string.
- TerminalNumber - Terminal Number. TYPE string.
- EndpointName - Name of endpoint. TYPE string.
- TrunkMember - Trunk member. TYPE non negative integer.
- PMCarrier - The PM Carrier. TYPE string.

- PMTimeSlot - The PM timeslot. TYPE string.
- TrunkType - Type of Trunk. TYPE string.
- TestResult - ICOT test result. TYPE string.
- ContinuityCondition - The continuity condition. TYPE string.
- AdditionalInfo - The additional information. TYPE string.
- CallID - The call identification. TYPE string.
- SupplementInfo - The supplement information. TYPE string.
- Version - The version refers to the version of the method or operation - e.g., the PostByGatewayName operation may have version 1.0 and 2.0, where version 1.0 takes parameters X and Y and version 2.0 takes parms X, Y, and Z. TYPE string.

The following parameters will be returned when some errors happen:

- Number - No. of the error. TYPE string. Required
- Message - Message of the error. TYPE string. Required
- Severity - Type of the error. TYPE string. The valid values include "INFORMATION", "WARNING", "MINOR", "MAJOR" and "CRITICAL". Optional
- Param1 - First parameter. TYPE string. Optional
- Param2 - Second parameter. TYPE string. Optional
- Param3 - Third parameter. TYPE string. Optional

Description of method parameters

Step **Action**

1 PostByGatewayName - Interface/Method that gets all the endpoint information for ISUP, PRI, PTS trunks based on Gateway Name and EndpointRange.

- Input data
  - — Version - Optional
  - — GatewayName - Required
  - — EndpointRange - Required
  - — FilterState - Optional
  - — ShowDetails - Optional
- Output data

- — Version - Optional
- — GatewayName - Required
- — NodeNumber - Required
- — FilterState - Required
- — EndpointRange
- — Value - Required
- — Count - Required
- — SupplementInfo - Optional
- — State - Optional
- — ConnectedTo - Optional
- — TrunkDirection - Optional
- — TrunkSignaling -Optional
- — PMType - Optional
- — PMNumber - Optional
- — TerminalNumber - Optional
- — EndpointName - Optional
- — TrunkCLLI - Optional
- — TrunkMember - Optional
- — PMCarrier - Optional
- — PMTimeSlot - Optional
- — TrunkType - Optional

2 QESByGatewayName - Interface/Method that query the endpoint state for ISUP, PRI, PTS trunks based on Gateway Name and EndpointRange.

- Input data
  - — Version - Optional
  - — GatewayName - Required
  - — EndpointRange - Required
  - — FilterState - Optional
  - — ShowDetails - Optional
- Output data
  - — Version - Optional

- GatewayName - Required
- NodeNumber - Required
- FilterState - Required
- EndpointRange
- Value - Required
- Count - Required
- SupplementInfo - Optional
- State - Optional
- TerminalNumber - Optional

3 BSYByGatewayName - Interface/Method that will busy ISUP, PRI, PTS trunks based on Gateway Name and EndpointRange.

- Input data
  - Version - Optional
  - GatewayName - Required
  - EndpointRange - Required
- Output data
  - Version - Optional
  - GatewayName - Required
  - NodeNumber - Required
  - EndpointRange
  - TerminalNumber - Optional

4 RTSByGatewayName - Interface/Method that will return ISUP, PRI, PTS trunks to service based on Gateway Name and EndpointRange.

- Input data
  - Version - Optional
  - GatewayName - Required
  - EndpointRange - Required
- Output data
  - Version - Optional
  - GatewayName - Required
  - NodeNumber - Required

- — EndpointRange
- — TerminalNumber - Optional

5 INBByGatewayName - Interface/Method that will make ISUP, PRI, PTS trunks be in Installation busy state based on Gateway Name and EndpointRange.

- Input data
  - — Version - Optional
  - — GatewayName - Required
  - — EndpointRange - Required
- Output data
  - — Version - Optional
  - — GatewayName - Required
  - — NodeNumber - Required
  - — EndpointRange
  - — TerminalNumber - Optional

6 FRLSByGatewayName - Interface/Method that will force to release endpoints of ISUP, PRI, PTS trunks based on Gateway Name and EndpointRange.

- Input data
  - — Version - Optional
  - — GatewayName - Required
  - — EndpointRange - Required
- Output data
  - — Version - Optional
  - — GatewayName - Required
  - — NodeNumber - Required
  - — EndpointRange
  - — TerminalNumber - Optional

7 PostByTrunkCLLI - Interface/Method that gets all the endpoint information for ISUP, PRI, PTS trunks based on Tunk CLLI and Trunk members.

- Input data

  - Version - Optional
  - CMClli - Optional
  - TrunkClli - Required
  - TrunkMembers - Required

- Output data
  - Version - Optional
  - TrunkCLLI - Required
  - CMCLLI - Required
  - FirstMember - Optional
  - GroupSize - Optional
  - TrunkMembers - Optional
  - State - Optional
  - ConnectedTo - Optional
  - TrunkDirection - Optional
  - TrunkSignaling - Optional
  - PMType - Optional
  - PMNumber - Optional
  - TerminalNumber - Optional
  - EndpointName - Optional
  - NodeNumber - Optional
  - GatewayName - Optional
  - TrunkMember - Optional, TYPE non negative integer.
  - PMCarrier - Optional
  - PMTimeSlot - Optional

8 BSYByTrunkCLLI - Interface/Method that will busy ISUP, PRI, PTS trunks based on TrunkCLLI and TrunkMembers.

- Input data
  - Version - Optional
  - CMClli - Optional
  - TrunkClli - Required
  - TrunkMembers - Required

- Output data
  - — Version - Optional
  - — TrunkCLLI - Required
  - — CMCLLI - Required
  - — FirstMember - Optional
  - — GroupSize - Optional
  - — TrunkMembers - Optional
  - — TrunkMember - Optional. TYPE string. Its format is the same as TrunkMembers.

9 RTSByTrunkCLLI - Interface/Method that will return ISUP, PRI, PTS trunks to service based on TrunkCLLI and TrunkMembers.

- Input data
  - — Version - Optional
  - — CMClli - Optional
  - — TrunkClli - Required
  - — TrunkMembers - Required

- Output data
  - — Version - Optional
  - — TrunkCLLI - Required
  - — CMCLLI - Required
  - — FirstMember - Optional
  - — GroupSize - Optional
  - — TrunkMembers - Optional
  - — TrunkMember - Optional. TYPE string. Its format is the same as TrunkMembers.

10 INBByTrunkCLLI - Interface/Method that will make ISUP, PRI, PTS trunks be in Installation Busy based on TrunkCLLI and TrunkMembers.

- Input data
  - — Version - Optional
  - — CMClli - Optional
  - — TrunkClli - Required

  - TrunkMembers - Required

- Output data
  - Version - Optional
  - TrunkCLLI - Required
  - CMCLLI - Required
  - FirstMember - Optional
  - GroupSize - Optional
  - TrunkMember - Optional. TYPE string. Its format is the same as TrunkMembers.

11 FRLSByTrunkCLLI - Interface/Method that will force to release endpoints of ISUP, PRI, PTS trunks based on TrunkCLLI and TrunkMembers.

- Input data
  - Version - Optional
  - CMClli - Optional
  - TrunkClli - Required
  - TrunkMembers - Required. TYPE string. Format: member number [, member number]*
    Examples:
    - "1" - Post trunk 1
    - “1,2” - Post trunk 1 and 2
- Output data
  - Version - Optional
  - TrunkCLLI - Required
  - CMCLLI - Required
  - FirstMember - Optional
  - GroupSize - Optional
  - TrunkMembers - Optional
  - TrunkMember - Optional. TYPE string. Its format is the same as TrunkMembers.

12 ICOTTest - Interface/Method that will do ICOT test for ISUP trunks based on TrunkCLLI and TrunkMember.

- Input data

    - Version - Optional
    - CMClli - Optional
    - TrunkClli - Required
    - TrunkMember - Required. TYPE non negative integer.

- Output data
    - Version - Optional
    - TrunkCLLI - Required
    - CMCLLI - Required
    - FirstMember - Optional
    - GroupSize - Optional
    - TrunkMembers - Optional
    - TrunkMember - Optional. TYPE non negative integer.
    - TestResult - Optional
    - ContinuityCondition - Optional
    - AdditionalInfo - Optional
    - CallID - Optional

13 PostGroupDChannelByTrunkCLLI - Interface/Method that gets all the endpoint information for PRI DChannel based on TrunkCLLI.

- Input data
    - Version - Optional
    - CMClli - Optional
    - TrunkClli - Required

- Output data
    - Version - Optional
    - TrunkCLLI - Required
    - CMCLLI - Required
    - FirstMember - Optional
    - GroupSize - Optional
    - TrunkMembers - Optional
    - State - Optional
    - ConnectedTo - Optional

  - — TrunkDirection - Optional
  - — TrunkSignaling - Optional
  - — PMType - Optional
  - — PMNumber - Optional
  - — TerminalNumber - Optional
  - — EndpointName - Optional
  - — NodeNumber - Optional
  - — GatewayName - Optional

14 BSYDChannelByTid - Interface/Method that busy PRI DChannel based on TerminalNumber.

- Input data
  - — Version - Optional
  - — CMClli - Optional
  - — NodeNumber - Required
  - — TerminalNumber - Required
  - — Force - Optional. If the DChannel is in INS and Force is false, BSYDChannelByTid will fail.
- Output data
  - — Version - Optional
  - — TrunkCLLI - Required
  - — TrunkMember - Optional. TYPE string. Its format is the same as TrunkMembers.

15 RTSDChannelByTid - Interface/Method that return PRI DChannel service based on TerminalNumber.

- Input data
  - — Version - Optional
  - — CMClli - Optional
  - — NodeNumber - Required
  - — TerminalNumber - Required
- Output data
  - — Version - Optional
  - — TrunkCLLI - Required

- TrunkMember - Optional. TYPE string. Its format is the same as TrunkMembers.

16 INBDChannelByTid - Interface/Method that make PRI DChannel in installation busy based on TerminalNumber.

- Input data
  - Version - Optional
  - CMClli - Optional
  - NodeNumber - Required
  - TerminalNumber - Required
- Output data
  - Version - Optional
  - TrunkCLLI - Required
  - TrunkMember - Optional. TYPE string. Its format is the same as TrunkMembers.

17 PostByCarrier - Interface/Method that gets all the endpoint information for ISUP, PRI, PTS trunks based on Gateway Name and CarrierNames.

- Input data
  - Version - Optional
  - GatewayName - Required
  - GWCName - Required
  - CarrierNames - Required
  - EndpointRange - Required
  - FilterState - Optional
  - ShowDetails - Optional
- Output data
  - Version - Optional
  - GatewayName - Required
  - NodeNumber - Required
  - FilterState - Required
  - EndpointRange - Required
  - Value - Required

- — Count - Required
- — SupplementInfo - Optional
- — State - Optional
- — ConnectedTo - Optional
- — TrunkDirection - Optional
- — TrunkSignaling - Optional
- — PMType - Optional
- — PMNumber - Optional
- — TerminalNumber - Optional
- — EndpointName - Optional
- — TrunkCLLI - Optional
- — TrunkMember - Optional
- — PMCarrier - Optional
- — PMTimeSlot - Optional
- — TrunkType - Optional

18 QESByCarrier - Interface/Method that query the endpoint state for ISUP, PRI, PTS trunks based on Gateway Name and Carrier.

- Input data
  - — Version - Optional
  - — GatewayName - Required
  - — GWCName - Required
  - — CarrierNames - Required
  - — EndpointRange - Required
  - — FilterState - Optional
  - — ShowDetails - Optional
- Output data
  - — Version - Optional
  - — GatewayName - Required
  - — NodeNumber - Required
  - — FilterState - Required
  - — EndpointRange - Required
  - — Value - Required

- Count - Required
- SupplementInfo - Optional
- State - Optional
- TerminalNumber - Optional

19
- Input data
  - Version - Optional
  - GatewayName - Required
  - GWCName - Required
  - CarrierNames - Required
  - EndpointRange - Required
- Output data
  - Version - Optional
  - GatewayName - Required
  - NodeNumber - Required
  - EndpointRange - Required
  - TerminalNumber - Optional

20 RTSByCarrier - Interface/Method that will return ISUP, PRI, PTS trunks to service based on Gateway Name and CarrierNames.

- Input data
  - Version - Optional
  - GatewayName - Required
  - GWCName - Required
  - CarrierNames - Required
  - EndpointRange - Required
- Output data
  - Version - Optional
  - GatewayName - Required
  - NodeNumber - Required
  - EndpointRange - Required
  - TerminalNumber - Optional

21 INBByCarrier - Interface/Method that will make ISUP, PRI, PTS trunks in installation busy based on Gateway Name and CarrierNames.

- Input data
  - — Version - Optional
  - — GatewayName - Required
  - — GWCName - Required
  - — CarrierNames - Required
  - — EndpointRange - Required
- Output data
  - — Version - Optional
  - — GatewayName - Required
  - — NodeNumber - Required
  - — EndpointRange - Required
  - — TerminalNumber - Optional

22 FRLSByCarrier - Interface/Method that will force to release endpoints of ISUP, PRI, PTS trunks based on Gateway Name and CarrierNames.

- Input data
  - — Version - Optional
  - — GatewayName - Required
  - — GWCName - Required
  - — CarrierNames - Required
  - — EndpointRange - Required
- Output data
  - — Version - Optional
  - — GatewayName - Required
  - — NodeNumber - Required
  - — EndpointRange - Required
  - — TerminalNumber - Optional

23 GetTrunkCllisByGatewayName - Interface/Method that will get trunk CLLIs of ISUP, PRI, PTS trunks based on Gateway Name.

- Input data
  - — Version - Optional
  - — CMCLLI - Optional
  - — GatewayName - Required
- Output data
  - — Version - Optional
  - — TrunkClli - Required

24 GetGatewayNames - Interface/Method that will get gateways' names.

- Input data
  - — Version - Optional
  - — MaxReturn - Optional.
  - — StartingIndex - Optional
  - — CMCLLI - Optional
  - — GWC - Optional
- Output data
  - — Version - Optional
  - — Names - Required

25 GetCMCLLIs - Interface/Method that will get CM CLLIs.

- Input data
  - — Version - Optional
  - — MaxReturn - Optional.
  - — StartingIndex - Optional
- Output data
  - — Version - Optional
  - — Names - Required

26 GetGWCNames - Interface/Method that will get Gateway Controllers' names.

- Input data
  - — Version - Optional
  - — MaxReturn - Optional.

- StartingIndex - Optional
- CMCLLI - Optional

- Output data
  - Version - Optional
  - Names - Required

—End—

XML commands

The following are examples of XML commands and the corresponding response messages for each operation supported for TMM. The XML coding in these examples is formatted for ease of understanding.

PostByGatewayName command

An example of PostByGatewayName request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<PostByGatewayName GatewayName="PRIFORDOC"
EndpointRange="35-36" ShowDetails="true"
FilterState="ALL"/>
</Methods>
</Command>
</CommandList>
```

An example of PostByGatewayName response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<PostByGatewayName>
<Header
GatewayName="PRIFORDOC"
EndpointRange="35-36"
NodeNumber="67"
FilterState="ALL">
<Summary>
<State Value="INB" Count="2"/>
</Summary>
</Header>
<Members>
```

```
<Member
TerminalNumber="35"
State="INB"
ConnectedTo="                    "
TrunkDirection="2W"
TrunkSignaling="ISD ISD"
PMType="GWC_NODE"
PMNumber="0"
EndpointName="DS1_10.4"
TrunkCLLI="PRIFORUPGRADEDOC"
TrunkMember="3"
PMCarrier="0"
PMTimeSlot="1"
TrunkType="PRI"/<
<Member
TerminalNumber="36"
State="INB"
ConnectedTo="                    "
TrunkDirection="2W"
TrunkSignaling="ISD ISD"
PMType="GWC_NODE"
PMNumber="0"
EndpointName="DS1_10.5"
TrunkCLLI="PRIFORUPGRADEDOC"
TrunkMember="4"
PMCarrier="0"
PMTimeSlot="1"
TrunkType="PRI"/>
</Members>
</PostByGatewayName>
</Methods>
</Response>
</CommandList>
```

QESByGatewayName command

An example of QESByGatewayName request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<QESByGatewayName
GatewayName="PRIFORDOC"
EndpointRange="35-36"
ShowDetails="true"
FilterState="ALL"/>
</Methods>
</Command>
</CommandList>
```

An example of QESByGatewayName response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<QESByGatewayName>
<Header
GatewayName="PRIFORDOC"
EndpointRange="35-36"
NodeNumber="67"
FilterState="ALL">
<Summary>
<State Value="INB" Count="2"/>
</Summary>
</Header>
<Members>
<Member TerminalNumber="35" State="INB"/>
<Member TerminalNumber="36" State="INB"/>
</Members>
</QESByGatewayName>
</Methods>
</Response>
</CommandList>
```

BSYByGatewayName command

An example of BSYByGatewayName request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<BSYByGatewayName
GatewayName="PRIFORDOC"
EndpointRange="35-36"/>
</Methods>
</Command>
</CommandList>
```

An example of BSYByGatewayName response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<BSYByGatewayName>
<Header
GatewayName="PRIFORDOC"
EndpointRange="35-36"
NodeNumber="67"/>
```

```
</BSYByGatewayName>
</Methods>
</Response>
</CommandList>
```

RTSByGatewayName command

An example of RTSByGatewayName request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<RTSByGatewayName
GatewayName="PVG194"
EndpointRange="674-675"/>
</Methods>
</Command>
</CommandList>
```

An example of RTSByGatewayName response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<RTSByGatewayName>
<Header
GatewayName="PVG194"
EndpointRange="674-675"
NodeNumber="16"/>
</RTSByGatewayName>
</Methods>
</Response>
</CommandList>
```

INBByGatewayName command

An example of INBByGatewayName request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<INBByGatewayName
GatewayName="PVG194"
EndpointRange="674-675"/>
</Methods>
</Command>
</CommandList>
```

An example of INBByGatewayName response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<INBByGatewayName>
<Header
GatewayName="PVG194"
EndpointRange="674-675"
NodeNumber="16"/>
</INBByGatewayName>
</Methods>
</Response>
</CommandList>
```

FRLSByGatewayName command

An example of FRLSByGatewayName request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<FRLSByGatewayName
GatewayName="PVG194"
EndpointRange="674,675"/>
</Methods>
</Command>
</CommandList>
```

An example of FRLSByGatewayName response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<FRLSByGatewayName>
<Header
GatewayName="PVG194"
EndpointRange="674,675"
NodeNumber="16"/>
</FRLSByGatewayName>
</Methods>
</Response>
</CommandList>
```

PostByTrunkCLLI command

An example of PostByTrunkCLLI request

```
<?xml version="1.0"?>
```

```
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<PostByTrunkCLLI
TrunkClli="P194ITLOOPOG1"
TrunkMembers="2-3"/>
</Methods>
</Command>
</CommandList>
```

An example of PostByTrunkCLLI response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<PostByTrunkCLLI>
<Header
TrunkCLLI="P194ITLOOPOG1"
CMCLLI="RTPS"
FirstMember="1"
GroupSize="24"
TrunkMembers="2-3"/>
<Members>
<Member
TrunkMember="2"
State="IDL"
ConnectedTo="                    "
TrunkDirection="2W"
TrunkSignaling="S7 S7"
PMType="GWC_NODE"
PMNumber="2"
PMCarrier="0"
PMTimeSlot="1"
GatewayName="PVG194"
EndpointName="SS_6003_VT15_0143.2"
NodeNumber="16"
TerminalNumber="674"/>
<Member
TrunkMember="3"
State="IDL"
ConnectedTo="                    "
TrunkDirection="2W"
TrunkSignaling="S7 S7"
PMType="GWC_NODE"
PMNumber="2"
PMCarrier="0"
PMTimeSlot="1"
GatewayName="PVG194"
```

```
EndpointName="SS_6003_VT15_0143.3"
NodeNumber="16"
TerminalNumber="675"/>
</Members>
</PostByTrunkCLLI>
</Methods>
</Response>
</CommandList>
```

BSYByTrunkCLLI command

An example of BSYByTrunkCLLI request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<BSYByTrunkCLLI
TrunkClli="P194ITLOOPOG1"
TrunkMembers="2-3"/>
</Methods>
</Command>
</CommandList>
```

An example of BSYByTrunkCLLI response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<BSYByTrunkCLLI>
<Header
TrunkCLLI="P194ITLOOPOG1"
CMCLLI="RTPS"
FirstMember="1"
GroupSize="24"
TrunkMembers="2-3"/>
<Members>
</Members>
</BSYByTrunkCLLI>
</Methods>
</Response>
</CommandList>
```

RTSByTrunkCLLI command

An example of RTSByTrunkCLLI request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
```

```
<Methods Version="2.0">
<RTSByTrunkCLLI
TrunkClli="P194ITLOOPOG1"
TrunkMembers="2-3"/>
</Methods>
</Command>
</CommandList>
```

An example of RTSByTrunkCLLI response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<RTSByTrunkCLLI>
<Header
TrunkCLLI="P194ITLOOPOG1"
CMCLLI="RTPS"
FirstMember="1"
GroupSize="24"
TrunkMembers="2-3"/>
<Members>
</Members>
</RTSByTrunkCLLI>
</Methods>
</Response>
</CommandList>
```

INBByTrunkCLLI command

An example of INBByTrunkCLLI request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<INBByTrunkCLLI
TrunkClli="P194ITLOOPOG1"
TrunkMembers="2-3"/>
</Methods>
</Command>
</CommandList>
```

An example of INBByTrunkCLLI response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<INBByTrunkCLLI>
```

```
<Header
TrunkCLLI="P194ITLOOPOG1"
CMCLLI="RTPS"
FirstMember="1"
GroupSize="24"
TrunkMembers="2-3"/>
<Members>
</Members>
</RTSByTrunkCLLI>
</Methods>
</Response>
</CommandList>
```

FRLSByTrunkCLLI command

An example of FRLSByTrunkCLLI request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<FRLSByTrunkCLLI
TrunkClli="P194ITLOOPOG1"
TrunkMembers="2-3"/>
</Methods>
</Command>
</CommandList>
```

An example of FRLSByTrunkCLLI response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<FRLSByTrunkCLLI>
<Header
TrunkCLLI="P194ITLOOPOG1"
CMCLLI="RTPS"
FirstMember="1"
GroupSize="24"
TrunkMembers="2-3"/>
<Members>
</Members>
</FRLSByTrunkCLLI>
</Methods>
</Response>
</CommandList>
```

PostByCarrier command

An example of PostByCarrier request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<PostByCarrier
GatewayName="PRIFORDOC"
GWCName="GWC-0"
CarrierNames="DS1_10"
EndpointRange="35-36"/>
</Methods>
</Command>
</CommandList>
```

An example of PostByCarrier response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<PostByCarrier>
<Header
GatewayName="PRIFORDOC"
EndpointRange="35-36"
NodeNumber="67"
FilterState="ALL">
<Summary>
<State Value="CFL" Count="2"/>
</Summary>
</Header>
</PostByCarrier>
</Methods>
</Response>
</CommandList>
```

QESByCarrier command

An example of QESByCarrier request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<QESByCarrier
GatewayName="PRIFORDOC"
GWCName="GWC-0"
CarrierNames="DS1_10"
EndpointRange="35-36"
```

```
ShowDetails="true"
FilterState="ALL"/>
</Methods>
</Command>
</CommandList>
```

An example of QESByCarrier response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<QESByCarrier>
<Header
GatewayName="PRIFORDOC"
EndpointRange="35-36"
NodeNumber="67"
FilterState="ALL">
<Summary>
<State Value="CFL" Count="2"/>
</Summary>
</Header>
<Members>
<Member
TerminalNumber="35"
State="CFL"/>
<Member
TerminalNumber="36"
State="CFL"/>
</Members>
</QESByCarrier>
</Methods>
</Response>
</CommandList>
```

BSYByCarrier command

An example of BSYByCarrier request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<BSYByCarrier
GatewayName="PRIFORDOC"
GWCName="GWC-0"
CarrierNames="DS1_10"
EndpointRange="35-36"/>
</Methods>
</Command>
</CommandList>
```

An example of BSYByCarrier response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<BSYByCarrier>
<Header
GatewayName="PRIFORDOC"
EndpointRange="35-36"
NodeNumber="67"/>
</BSYByCarrier>
</Methods>
</Response>
</CommandList>
```

RTSByCarrier command

An example of RTSByCarrier request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<RTSByCarrier
GatewayName="PRIFORDOC"
GWCName="GWC-0"
CarrierNames="DS1_10"
EndpointRange="35-36"/>
</Methods>
</Command>
</CommandList>
```

An example of RTSByCarrier response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<RTSByCarrier>
<Header
GatewayName="PRIFORDOC"
EndpointRange="35-36"
NodeNumber="67"/>
</RTSByCarrier>
</Methods>
</Response>
</CommandList>
```

INBByCarrier command

An example of INBByCarrier request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<INBByCarrier
GatewayName="PRIFORDOC"
GWCName="GWC-0"
CarrierNames="DS1_10"
EndpointRange="35-36"/>
</Methods>
</Command>
</CommandList>
```

An example of INBByCarrier response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<INBByCarrier>
<Header
GatewayName="PRIFORDOC"
EndpointRange="35-36"
NodeNumber="67"/>
</INBByCarrier>
</Methods>
</Response>
</CommandList>
```

FRLSByCarrier command

An example of FRLSByCarrier request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<FRLSByCarrier
GatewayName="PRIFORDOC"
GWCName="GWC-0"
CarrierNames="DS1_10"
EndpointRange="35-36"/>
</Methods>
</Command>
</CommandList>
```

An example of FRLSByCarrier response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<FRLSByCarrier>
<Header
GatewayName="PRIFORDOC"
EndpointRange="35-36"
NodeNumber="67"/>
</FRLSByCarrier>
</Methods>
</Response>
</CommandList>
```

PostGroupDChannelByTrunkCLLI command

An example of PostGroupDChannelByTrunkCLLI request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<PostGroupDChannelByTrunkCLLI
TrunkClli="PRIFORUPGRADEDOC"/>
</Methods>
</Command>
</CommandList>
```

An example of PostGroupDChannelByTrunkCLLI response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<PostGroupDChannelByTrunkCLLI>
<Header
TrunkCLLI="PRIFORUPGRADEDOC"
CMCLLI="COMPACT5"/>
<Members>
<Member
State="INB"
TrunkDirection="2W"
TrunkSignaling = "ISD ISD"
PMType="GWC_NODE"
PMNumber="0"
GatewayName="PRIFORDOC"
EndpointName="DS1_10.2"
NodeNumber="67"
TerminalNumber="33"/>
```

```
<Member
State="INB"
TrunkDirection="2W"
PMType="GWC_NODE"
PMNumber="0"
GatewayName="PRIFORDOC"
EndpointName="DS1_10.3"
NodeNumber="67"
TerminalNumber="34"/>
</Members>
</PostGroupDChannelByTrunkCLLI>
</Methods>
</Response>
</CommandList>
```

ICOTTest command

An example of ICOTTest request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<ICOTTest
TrunkClli="SUC101ISUPV2LP"
TrunkMember="2"/>
</Methods>
</Command>
</CommandList>
```

An example of ICOTTest response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<ICOTTest>
<Header
TrunkCLLI="SUC101ISUPV2LP"
CMCLLI="COMPACT2"
FirstMember="2"
GroupSize="2"
TrunkMembers="2"/>
<Members>
<Member
TrunkMember="2"
TestResult="TEST_PASSED"
ContinuityCondition="                                "
AdditionalInfo="
CallID=""/>
</Members>
```

```
</ICOTTest>
</Methods>
</Response>
</CommandList>
```

BSYDChannelByTid command

An example of BSYDChannelByTid request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<BSYDChannelByTid
NodeNumber="18"
TerminalNumber="543"
Force="true"/>
</Methods>
</Command>
</CommandList>
```

An example of BSYDChannelByTid response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<BSYDChannelByTid>
<Header
CMCLLI="COMPACT2"/>
<Members>
</Members>
</BSYDChannelByTid>
</Methods>
</Response>
</CommandList>
```

RTSDChannelByTid command

An example of RTSDChannelByTid request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<RTSDChannelByTid
NodeNumber="18"
TerminalNumber="543"/>
</Methods>
</Command>
</CommandList>
```

An example of RTSDChannelByTid response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<RTSDChannelByTid>
<Header
CMCLLI="COMPACT2"/>
<Members>
</Members>
</RTSDChannelByTid>
</Methods>
</Response>
</CommandList>
```

INBDChannelByTid command

An example of INBDChannelByTid request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<INBDChannelByTid
NodeNumber="18"
TerminalNumber="543"/>
</Methods>
</Command>
</CommandList>
```

An example of INBDChannelByTid response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<INBDChannelByTid>
<Header
CMCLLI="COMPACT2"/>
<Members>
</Members>
</INBDChannelByTid>
</Methods>
</Response>
</CommandList>
```

GetTrunkCllisByGatewayName command

An example of GetTrunkCllisByGatewayName request

```
<?xml version="1.0"?>
```

```
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<GetTrunkCllisByGatewayName
GatewayName="PRIFORDOC"/>
</Methods>
</Command>
</CommandList>
```

An example of GetTrunkCllisByGatewayName response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<GetTrunkCllisByGatewayName>
<TrunkClli>PRIFORUPGRADEDOC</TrunkClli>
</GetTrunkCllisByGatewayName>
</Methods>
</Response>
</CommandList>
```

GetGatewayNames command

An example of GetGatewayNames request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<GetGatewayNames/>
</Methods>
</Command>
</CommandList>
```

An example of GetGatewayNames response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<GetGatewayNames>
<Gateway Names="GAOPRI,PRIFORDOC,PVG1"/>
</GetGatewayNames>
</Methods>
</Response>
</CommandList>
```

GetCarriers command

An example of GetCarriers request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<GetCarriers/>
</Methods>
</Command>
</CommandList>
```

An example of GetCarriers response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<GetCarriers>
<Carriers GatewayName="PVG1" GWCName="GWC-112"
CarrierNames=
"E1_1111,E1_1112"/>
<Carriers GatewayName="GAOPRI" GWCName="GWC-0"
CarrierNames=
"E1_1001"/>
<Carriers GatewayName="PRIFORDOC" GWCName="GWC-0"
CarrierNames=
"DS1_10,DS3_10.1"/>
</GetCarriers>
</Methods>
</Response>
</CommandList>
```

GetCMCLLIs command

Note: The GetCMCLLIs command is only available in (I)SN08 and previous releases. In (I)SN08, this command was replaced by GetCMCLLIbySDMIP.

An example of GetCMCLLIs request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<GetCMCLLIs/>
</Methods>
</Command>
</CommandList>
```

An example of GetCMCLLIs response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<GetCMCLLIs>
<CallServer Names="COMPACT5"/>
</GetCMCLLIs>
</Methods>
</Response>
</CommandList>
```

GetCMCLLIbySDMIP command

An example of GetCMCLLIbySDMIP request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<GetCMCLLIbySDMIP SDMIP="47.140.120.11" />
</Methods>
</Command>
</CommandList>
```

An example of GetCMCLLIbySDMIP response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<GetCMCLLIbySDMIP>
<CMCLLI Names="COMPACT6"/>
</GetCMCLLIbySDMIP>
</Methods>
</Response>
</CommandList>
```

GetGWCNames command

An example of GetGWCNames request

```
<?xml version="1.0"?>
<CommandList>
<Command>
<Interface>TrunkMtc</Interface>
<Methods Version="2.0">
<GetGWCNames/>
</Methods>
</Command>
</CommandList>
```

An example of GetGWCNames response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<GetGWCNames>
<GWC Names="GWC-112,GWC-4,GWC-1,GWC-2,GWC-0"/>
</GetGWCNames>
</Methods>
</Response>
</CommandList>
```

Error response

An example of error response

```
<?xml version="1.0"?>
<CommandList>
<Response>
<Interface>TrunkMtc</Interface>
<Methods>
<RTSDChannelByTid>
<Header
CMCLLI="COMPACT5"/>
<Members>
<Member>
<Error
Number="1311"
Message="Unable to transition the endpoint to the
requested state"
Severity="MAJOR"/>
</Member>
</Members>
</RTSDChannelByTid>
</Methods>
</Response>
</CommandList>
```

User authorization for trunk maintenance operations

In addition to users belonging to “succssn” group to login to OSSGate, user need to be in application specific groups to perform specific operations. Each operation is associated with one or more user groups. In order to execute a command, a user must belong to at least one of the associated user groups. The user groups associated with each TMM maintenance operation are specified in the table below.

Command	User Group				
	trkadm	trkrw	trkmtc	trksprov	trkro
Post by Trunk CLLI	X	X	X	X	X
D-Channel Post by Trunk CLLI	X	X	X	X	X
Mtc by Trunk CLLI (BSY, RTS, INB, FRLS)	X	X	X		
ICOT	X	X	X		
D-Channel MTC by Trunk CLLI (BSY, RTS, BSYINB)	X	X	X		
Post by Gateway Name	X	X	X	X	X
QES by Gateway Name	X	X	X	X	X
Mtc by Gateway Name (BSY, RTS, INB, FRLS)	X	X	X		
Post by Carrier	X	X	X	X	X
QES by Carrier	X	X	X	X	X
Mtc by Carrier	X	X	X		

Limitations and restrictions

Step Action

1 Only individual trunk members (no lists or ranges) may be specified when performing an ICOT test. If an ICOT test is requested on an ISUP trunk which is not in the BSY state, the request will be rejected.

2 Maintenance operations are supported for ISUP, PRI, and PTS trunks. Other trunk types may work, but they are not explicitly supported.

3 Post Endpoints by Gateway name is supported.

4 Only one maintenance (RTS, BSY, BSY INB or FRLS) request, plus either POST or QES, is allowed per XML request. Batching of maintenance requests is not allowed.

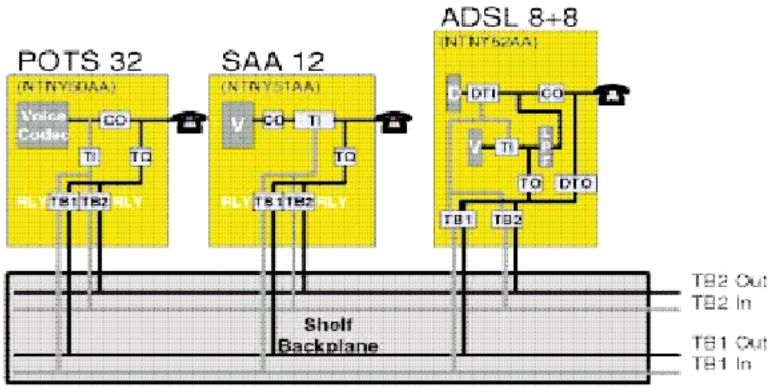
5 TMM supports the following commands: BSY, RTS, FRLS, BSY INB, Post (by Trunk CLLI/gateway name/carrier) and QES (by gateway name/carrier) for ISUP, PRI and PTS trunks. Other trunk types may work, but they are not explicitly supported.

—End—

TL1 Line Test Interface for MG 9000 lines

The TL1 Open line test access interface allow test OSS vendors to operate line test access relays. A TL1 interface in the CS 2000 management tool system provides a common point of access for MG 9000 line test. TL1 provides an accepted, standards-based interface and command set as an overlay to pre-existing line test functionality on the MG 9000. Carrier VoIP Networks MG 9000 line gateway supports POTS, coin, P-phone, ground start, 8x8 POTS and ADSL and DS-1 circuits. Line test access is via MTA (Metallic Test Access) matrix built in the gateway. Each gateway has eight (8) test access ports available; each of which supports 2-wire (one tip-ring pair) test access. The following test directions are supported by the circuits:

- test in (towards the circuit)
- test out (towards the subscriber loop)
- test bridged configuration



Legend:
TI - test in
TO - test out DTO - data test out
CO - cut off LPF - low pass filter
TB1 - Puts card's MTA bus on shelf test bus 1
TB2 - Puts card's MTA bus on shelf test bus 2
DTI - Combination cut off and data test in. When operated, cuts loop off from ADSL chip set, and connects the ADSL chipset to the test-in bus.

Although the high impedance isolation is not provided by the MTA in test bridged configurations, it is believed that nearly all currently available test heads have such isolation built in them. The MTA is designed to support both narrowband and wideband line test, provided the configuration meets Nortel Networks' guide lines.

A MG 9000 node can have multiple frames which can house up to four (4) shelves each.

Each shelf has multiple cards serving subscribers. Each card has multiple circuits which each support a subscriber loop. In an MG 9000 node, any test access port can reach any loop for testing. A maximum of two concurrent line test access connections can be established on each shelf. One concurrent line test connection can be established per card.

Open line test interface coverage

The following will be supported in (I)SN08:

- Line test access of MG 9000 lines.
- Line testing on all Carrier VoIP Networks supported POTS lines, 8x8 POTS & ADSL lines on MG 9000.
- Both narrowband and wideband line test wherever applicable; although the test access ports only support 2-wire test access

Communication and interface protocols

The Test System Controller (TSC) communicates with the CMT (Call server Management Tools) for line test access request. CMT is connected to a secured IP network. The communication between TSC and CMT is through the SSH (Secured SHell).

Once logged in to the OSSGate, the user will need to change modes to enable the TL1 interface.

From the OSSGate prompt, the user will enter ^B (control-B)

```
>^B <enter>
```

```
?
```

A question-mark will be displayed indicating the user is now in control mode. To change to TL1 mode, enter:

```
? mode tl1 <enter>
```

The user will see the following message.

```
mode is TL1
>
```

OSSGate is now ready to forward TL1 commands to the TL1 server for processing.

User authorization

Each TL1 command is assigned to one or more user authorization groups. For an OSSGate login user to have permission to execute a TL1 command, the user must belong to an authorization group associated with the command. The following table provides a mapping of TL1 commands to the required group membership of the user running them.

TL1 command	Permitted authorization changes
CONN-SLACC-MET	lnmtc, lnadm, lnrw
CHG-SLACC-DIR	lnmtc, lnadm, lnrw
CHG-SLACC-BW	lnmtc, lnadm, lnrw
REPT-STAT	lnmtc, lnadm, lnrw
DISC-SLACC	lnmtc, lnadm, lnrw
RTRV-SLACC-INFO	lnmtc, lnadm, lnrw, lnro

Members of “linemtc, lnadm, and lnrw ” have permission to run all TL1 line test commands. The “lnro” authorization group restricts users to query functions only via the RTRV-SLACC-INFO command.

Users can be assigned to different authorization groups with the “usermod” command from the UNIX command line on the CMT server. Only the root user is permitted to run this command. See document: NN10276-500 “ATM/IP Configuration Management” for details

Line test access functionality

TL1 interface behavior and dependencies

Step **Action**

1 Users are required to login to the OSSGate on the CMT via the SSH protocol using a user ID and password.

2 The same user ID and password can be used to login more than one TL1 session.

3 The MG 9000 maintains an inactivity timeout for line test connections of 180 seconds. If the timeout expires, the TL1 user’s existing test access connection will be disconnected and further commands will fail until a new test connection is established. The user login session will not be impacted by this timeout.

4 The OSSGate application maintains an inactivity timeout of 5 minutes (300 seconds). If this timeout expires, the TL1 user will be logged out of the system.

5 Each TL1 login session can have multiple active test connections. The maximum number of test connections of each TL1 login user will be determined during the implementation.

6 There will not be unsolicited messages sent to the user. One TL1 request can expect one TL1 response.

7 Directory numbers must be specified in their fully qualified form (FQDN) for commands that operate on test access relays. It is optional only for the RTRV-SLACC-INFO.

—End—

TL1 line test access command summary

Command Code	Description
CONN-SLACC-MET	Connect the subscriber loop on the test bus.
CHG-SLACC-DIR	Set the test direction to IN (towards the line circuit), OUT (towards the subscriber loop) or BRIDGED.
CHG-SLACC-BW	Change the test access band between narrowband access and wideband access.
REPT-STAT	Reset the 180 second inactivity timer in the system.
DISC-SLACC	Reset all the test access relays to their default state.
RTRV-SLACC-INFO	Retrieve a subscribers’ line information.

Note: Transaction Language 1 (TL1) is the line test access protocol used for this interface. It is based on the Telcordia specifications GR-831-CORE and GR-834-CORE. The following sections describe the commands and syntax supported by this interface.

TL1 syntax rules

The following are the TL1 syntax rules enforced by this implementation:

- All TL1 fields are delimited by “:” characters.
- The first 5 fields are mandatory - even if blank. This includes Command code, Target Id, Access Id, Correlation Tag, and General Block.
- If a specific field allows white-space, then it must be enclosed by double quotes “” if the white-space is to be preserved, otherwise it is ignored.
- Each field may consist of one or more parameters separated by white-space or optional comma. This implementation uses commas to delimit parameters within a given field.

These components will each be discussed in turn.

Command code (mandatory)

This is the actual command name to be executed and is typically defined in several parts:

```
<TL1Verb>-<Modifier1>-<Modifier2>
```

A dash “-” character is used to delimit these components.

For this feature, the following command codes are supported:

- CONN-SLACC-MET
- CHG-SLACC-DIR
- CHG-SLACC-BW
- RTRV-SLACC-INFO
- REPT-STAT
- DISC-SLACC

Valid characters for this field are: letters, digits, or hyphen “-”. Any other characters found in this field will cause the operation to fail. This field is not case sensitive.

Target Identifier (TID) - mandatory

As per the TL1 specification (GR-831), this field can consist of up to 20 characters and must be letters, digits, or hyphens. In this implementation, it is used to define the address format of the request and must be set to one of the following values:

- DN
- LEN

Although this field is mandatory, certain commands require that it be left empty. This field is not case sensitive.

Access Identifier Block - mandatory

This field can consist of one or more parameters. The parser mainly checks for illegal white-space in unquoted parameters.

In this implementation, this field is used to specify the DN or LEN to be tested and is not case sensitive.

Correlation Tag (CTAG) - mandatory

This field can have a maximum size of 6 alphanumeric characters. This is a mandatory field for all commands. This field is case sensitive.

General Block - mandatory

This field is not used in this implementation. However, it is still mandatory and must be specified as an empty field.

Payload Block - optional

The payload block contains user-defined data and can consist of zero or more fields. Each Field may contain one or more data elements separated by commas.

Parameters within each field can be specified as either Positional or Name-defined

- Positional parameters have a fixed number of components. Empty components are specified by omitting the data, but inserting the delimiter in its place to show the data as missing. The rules are basically the same as that used when using CSV. All positional components must be specified even if the data is not present.
- Named-defined parameters specify their components using a NAME=VALUE approach. The order of components is not enforced and can be omitted if desired.

ATTENTION

For this feature, all commands defined contain either zero or one parameter in the payload block. Only Name-defined parameters are used and contain a variable number of components (depending on the command). All tag names and enumerated values are case insensitive. All other values are treated as case sensitive.

TL1 responses

There are two types of TL1 response message formats used in this feature: (note that spaces are represented by the ^ (caret) character).

Typical response containing header and optional parameter fields

Success response

```
<cr> <lr> <lf>
^^^<af>^<date>^<time> <cr> <lf>
M^^<ctag>^<cs> <cr> <lf>
^^^["<Optional comma-delimited parameters>"]
<cr> <lf>;
```

where

`<aft>` is the address format

where

`<lf>` is the line feed

where

<cr> is the carriage return

where

cs is the command state (COMPLD =success, DENY = error_

Error message format

```
<cr><lf><lf>
^^^<af>^<date>^<time> <cr> <lf>
M^^<ctag>^DENY <cr><lf>
^^^<TL1 ERROR CODE><cr><lf>
^^^[<optional comma-delimited error parameters>]<cr><lf>
^^^/*<GR-834 error code description>*/<cr><lf>
^^^/*<System-specific error description>*/<cr><lf>
;
```

Error message example

```
> CONN-SLACC-MET:DN:1418420217:CTAG2::TAP=2,BW=NB,
DIR=O,sz=1;
   DN 04-06-17 14:13:51
M  CTAG2 DENY
   IPMS
   DN=1418420217,TYPE="SINGLE PARTY LINE",LEN="UAIP  00 2
02 17",OPTIONS=DGT,GW_NAME=UAIP000-0-2,EP_NAME=tp/
02/17,GW_NAME=UAIP000-0-2,NE_NUM=2,LF=UE9000MG,LS=
IDL,CLLI=COMPACT3,BW=NB,DIR=O,sz=1
   /* PARAMETER MISSING */
   /* Missing command parameter "FA". */
;
```

Acknowledgement

This response type is used by the application to construct TL1 acknowledgements to user commands. This type of TL1 response contains no data. It is used to simply ACK specific queries. In this implementation, it is returned as a response to the REPT-STAT command.

```
OK^<CTAG><cr><lf>
<
> rept-state::8420217:ctag2;
OK ctag2
<
```

Concurrent TL1 operations

The TL1 Interface for MG 9000 line test restricts the number of concurrent TL1 commands to 5. Although the number of TL1 login users may exceed this value, the TL1 server will limit the number of TL1 commands that can execute simultaneously on the CS2M. When all resources are busy, a response similar to the following will be returned to indicate that resource exhaustion has (temporarily) occurred.

```
DNLEN 04-10-13 09:51:49
```

```
M  UNKNWN DENY
   ESPG
    /* SOFTWARE PROGRAM */
    /* Error creating command instance for command
"TL1Transaction"  pool.ResourceUnavailableException: The
maximum number of concurrent    instances of this command
has been reached, please try later. */
;
```

This problem typically occurs when more than 5 TL1 OSS clients are simultaneously logged in and executing commands at the same time. Normally this does not occur with TL1 login users entering commands manually.

To avoid this condition, it is recommended that the number of TL1 user logins does not exceed the concurrent operations threshold (5).

Connect subscriber loop on the test bus (CONN-SLACC-MET)

The command CONN-SLACC-MET connects the subscriber loop on the test bus. The input, output and the behavior of this command are described in the following sections.

Input fields

Syntax

```
CONN-SLACC-MET:<af>:<DN|LEN>:<ctag>::TAP=<tap>,
BW=<bw>,DIR=<dir>,FA=<fa>,SZ=<sz>;
```

Connect a subscriber line on the test bus (CONN-SLACC-MET)	Description
Input:	
AF (Address Format) (Mandatory input)	For DN access, enter DN. For LEN access, enter LEN.
DN, LEN (Mandatory input)	
DN format (# of digits)	If DN is entered, only numerical digits (0-9) are valid input. Min # of digits: 1 Max # of digits: 30 In order to support both North American and international DN format, the TL1 command parser will not reject the number of digits within MIN and MAX above. If the number of digits is invalid to the call server, it should be rejected by the call server.

Connect a subscriber line on the test bus (CONN-SLACC-MET)	Description
LEN format	This is the traditional Line Equipment Number stored in CS 2000 Core. If LEN is entered, alphanumeric character values (0-9, A-Z, a-z) are valid input. The TL1 server converts all characters to upper case before processing the command. LEN format in call server will be used, i.e. SITE name + LEN number For example: CARY 000 1 01 12 CARY is the physical location where the line is originating. 000: physical frame number in Cary 1: shelf number 01: slot number 12: circuit number leading zero (0) is required. For Multiple Appearance Directory Number (MADN), line test access is via LEN only. ***Note:*** LENs must be enclosed in double quotes “”
CTAG (correlation tag) (Mandatory input)	Correlation tag (max 6 ASCII characters)
TAP (Test Access Port) (0-999) (Mandatory input)	Test access port ID For MG 9000, it is 1 to 8.
BW (Access Bandwidth) (Mandatory input)	
NB	Narrowband. For MG 9000 POTS and SAA card, there is only one test access path. NB or WB will result in the test access on the same path, i.e., no difference. For the 8x8 ADSL card on MG 9000, NB line test access allows for the narrowband test access. The test access point is between the POTS circuit card and LPF (Low Pass Filter), when the test direction is OUT (towards the equipment side or subscriber loop side), the test head sees the subscriber loop through LPF. This test access does not cut off the ADSL service.

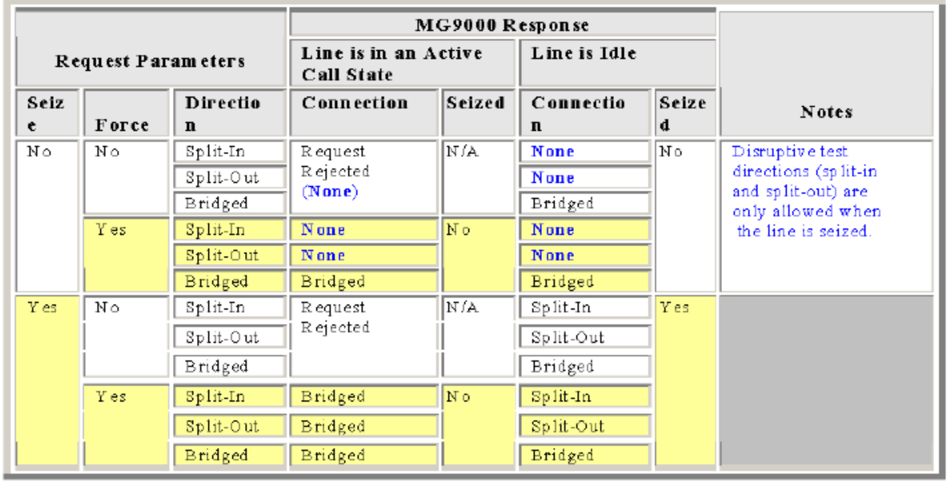
Connect a subscriber line on the test bus (CONN-SLACC-MET)	Description
WB	Wideband Currently is for the 8x8 ADSL card on MG 9000 wideband access. The test access is at the subscriber loop side of the splitter. It cuts off both POTS and ADSL services. In Phase 1.1, the WB connection option will not be rejected, unless it is an ADSL circuit.
DIR (Initial test Direction) (Mandatory input)	
O (OUT)	For lookout towards the subscriber loop. It is also called Equipment side.
B (BRIDGED)	Tap the test pair from the test head to the subscriber line. High impedance isolation is not provided.
I (IN)	For look in towards the circuit.
FA (Forced access flag) (Mandatory input)	

Flag=1, if the line state is CPB or CPD, set the connection up and in BRIDGED mode. Do not allow the user to change the test direction.
Flag=2, if the line state is CPB or CPD, reject the connection state.
See the table below for the detailed description of the behavior of the forced access flag and seize fields.
	Flag=1
	Flag=2

SZ (SeiZe) (Mandatory input)	
1	Seize the line This mode allows the test head to perform the loop testing. However, it does not allow the test head to draw dial tone test. The POTS circuit will not supply the dial tone. In this mode, the voice service is in out-of-service state. The call server will route the caller to this line for a treatment, e.g. supply busy tone or an announcement
2	Do not seize the line. This mode allows the test head to perform not only the loop testing, but draw dial tone test when the test direction is in either bridged state or in state. In this mode, the voice service on the line will behave as follows:

Connect a subscriber line on the test bus (CONN-SLACC-MET)	Description
	The line remains in service. The caller to this line will get the normal ring back tone. Since the called party is under test, the loop is cut off. The called party will not hear the ringing tone

MG 9000 Responses to TL1 Force and Seize Parameters



Request Parameters			MG9000 Response				Notes
			Line is in an Active Call State		Line is Idle		
Seize	Force	Direction	Connection	Seized	Connection	Seized	
No	No	Split-In	Request Rejected (None)	N/A	None	No	Disruptive test directions (split-in and split-out) are only allowed when the line is seized.
		Split-Out			None		
		Bridged			Bridged		
	Yes	Split-In	None	No	None		
		Split-Out	None		None		
		Bridged	Bridged		Bridged		
Yes	No	Split-In	Request Rejected	N/A	Split-In	Yes	
		Split-Out			Split-Out		
		Bridged			Bridged		
	Yes	Split-In	Bridged	No	Split-In		
		Split-Out	Bridged		Split-Out		
		Bridged	Bridged		Bridged		

This table is valid for CONN_SLACC-MET commands. In cases where the initial connection has been made, this table is also valid for CHG_SLACC_DIR commands.

When the MG 9000 seizes a line for testing, the line state at the call server becomes SB.

For the purposes of this table, the call state column headings map to the following call server states:

- Active Call States: CPB, CPD.
- Idle States: IDL, INB, MB, PLO, LO.

Regarding Line States not supported by this feature—DEL, LMB, NEQ, SB: The TL1 interface implementation in SESM checks for these call server line states. For lines in these states, connection requests are not forwarded to the MG 9000.

> ***Note:*** Testing is not supported if SB is the initial state. This does not apply to lines that have gone SB because the MG 9000 has seized them for testing.

The following TL1 parameters map to SNMP MTA request entries:

- The SEIZE parameter maps directly.
- FORCE=Yes in TL1 maps to FORCE=Normal in SNMP.
- FORCE=No in TL1 is new. It will map to the new value ForceDont (6) in SNMP.

The following is a description of the line status and behavior during the line testing:

State	Action	No Seize	Seize
CPB	The CS 2000/MG 9000 must give the test system an indication that it has accessed a line that is in an active call. If desired, the test system can test the line in a non-disruptive bridged state. The CS 2000/MG 9000 must not allow a disruptive test action on a line involved in an active call.	Must allow bridged connections. Must not affect the line state.	Must allow split-in, split-out and bridge connections. Must set the line to an Out of Service state.
CPD	The CS 2000/MG 9000 must give the test system an indication that it has accessed a line that is in an active call. If desired, the test system can test the line in a non-disruptive bridged state. The CS 2000/MG 9000 must not allow a disruptive test action on a line involved in an active call.	Must allow split-in, split-out and bridge connections. Must not affect the line state.	Must allow split-in, split-out and bridge connections. Must set the line to an Out of Service state.
DEL	The CS 2000/MG 9000 will not support testing a line in the DEL state.		
IDL	The CS 2000/MG 9000 must allow testing a line in the IDL state. This includes testing with a non-disruptive bridged connection and testing with disruptive split connections.	Must allow split-in, split-out and bridge connections. Must not affect the line state.	Must allow split-in, split-out and bridge connections. Must set the line to an Out of Service state.

State	Action	No Seize	Seize
INB	The CS 2000/MG 9000 must allow testing a line in the INB state. This includes testing with a non-disruptive bridged connection and testing with disruptive split connections.	Must allow split-in, split-out and bridge connections. Must not affect the line state.	Must allow split-in, split-out and bridge connections. Must set the line to an Out of Service state.
LMB	The CS 2000/MG 9000 will not support testing a line in the LMB state.		
LO	The CS 2000/MG 9000 must allow testing a line in the LO state. This includes both non-disruptive bridged connections and disruptive split connections.	Must allow split-in, split-out and bridge connections. Must not affect the line state.	Must allow split-in, split-out and bridge connections. Must set the line to an Out of Service state.
MB	The CS 2000/MG 9000 must allow testing a line in the MB state. This includes both non-disruptive bridged connection and disruptive split connections.	Must allow split-in, split-out and bridge connections. Must not affect the line state.	Must allow split-in, split-out and bridge connections. Must set the line to an Out of Service state.
NEQ	The CS 2000/MG 9000 will not support testing a line in the NEQ state.	Must allow split-in, split-out and bridge connections. Must not affect the line state.	Must allow split-in, split-out and bridge connections. Must set the line to an Out of Service state.
PLO	The CS 2000/MG 9000 must allow testing a line in the PLO state. This includes both non-disruptive bridged connections and disruptive split connections.	Must allow split-in, split-out and bridge connections. Must not affect the line state.	Must allow split-in, split-out and bridge connections. Must set the line to an Out of Service state.
SB	The CS 2000/MG 9000 will not support testing a line in the SB state.		

Input format and examples

Syntax

```
CONN-SLACC-MET:<af>:<DN or LEN>:<ctag>::
TAP=<tap>,BW=<bw>,DIR=<dir>,FA=<fa>,SZ=<sz>;
```

Access though DN example

```
CONN-SLACC-MET:DN:4166213000:CTAG1::  TAP=1,BW=NB,DIR=O,F
A=1,SZ=2;
```

Access through LEN example

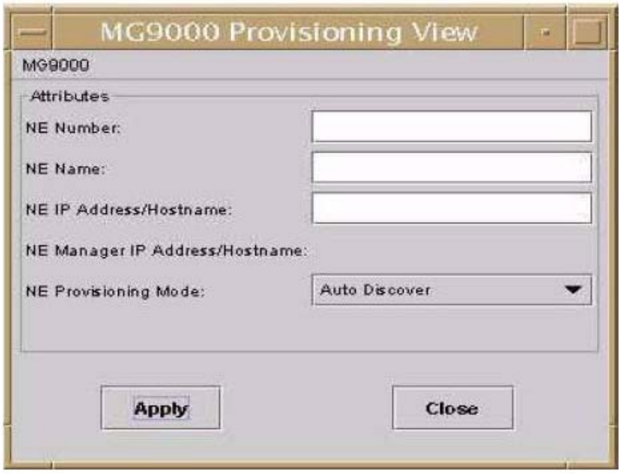
```
CONN-SLACC-MET:LEN:”CARY 000 1 01 12”:CTAG1::TAP=1,BW=NB
,DIR=O,FA=1,SZ=2;
```

Within a parameter field, when there is more than one field, a comma ‘,’ is placed between the two adjacent fields as separator. There must not be space before and after the comma.

Output fields

Connect a subscriber line on the test bus (CONN-SLACC-MET)	Description
Output:	
CTAG	Correlation tag
CS (Command Status)	COMPLD: command is executed successfully. DENY: command is executed unsuccessfully.
ACTUALDIR (The actual connection direction) (In, Out, Bridged)	MG 9000: I (IN), O (OUT), B (BRIDGED) ***Note:*** This parameter is populated only if the actual test direction set by the MG 9000 is different than requested.
NE Number	The NE NUMBER is a numeric digit defined for each MG 9000 node. This ID is unique to all MG 9000 nodes associated with the same call server. It is limited to 3 digits. It is not enforced to make sure that it is in sequential order. For example, if the NE NUMBER for the first installed MG 9000 node is defined as 0, the NE NUMBER of the next installed MG 9000 node does not have to be 1, as long as it is not 0 (it has to be unique). This NE NUMBER is displayed with label: NE Number with the MG 9000 node name (NE Name) (refer to the following screen shot). It is required for the subsequent command request, NE Number and Tap are the mandatory input parameters for the commands after the test connection is set up.
GW_NAME (NE name)	Gateway name defined for MG 9000. ***Note:*** This is the same as the NE name.
CS 2000 CLLI	
LS (Line State)	The Line state before the test access request.
LEN (Equipment Number)	Line Equipment Number (Site name and equipment number). Example: CARY 000 1 01 09

Connect a subscriber line on the test bus (CONN-SLACC-MET)	Description
TYPE (Line Type) (POTS, ISDN, P-phone, etc.)	The Line Type as specified by the QDN/QLEN command
LF (Line Frame type)	For MG 9000 return UE9000MG. This field is populated with the gateway type defined in the CMT database.
LCC (Line Class Code)	Line class code of the line as defined in the call server
CARDCODE	Line card type as defined in the CS2000. Examples: GWLPOT, RDTLSG, 6X21AC...



MG9000 Provisioning View

MG9000

Attributes

NE Number:

NE Name:

NE IP Address/Hostname:

NE Manager IP Address/Hostname:

NE Provisioning Mode: Auto Discover

WARNING

If the test connection is in the bridged or in state, Nortel recommends that the test controller may only execute passive commands, i.e., those which do not require applying voltage, current, or tones to the loop or line card (such as AC/DC voltage measurements) to subscriber test functions. For example, do not perform the 12 metallic loop test function. The reason is that MG 9000 POTS card has a hazard voltage protection feature to protect the POTS circuit from foreign voltage. If the voltage applied on the line in BRIDGED (or IN state) is above the hazardous voltage threshold, the subscriber line will be cut off. This may cut off an active call.

Output format and examples

Syntax

```
<cr> <lf> <lf>
^^^<af>^<data>^<time> <cr> <lf>
M^^<ctag>^<cs> <cr> <lf>
^^^"ACTUALDIR=<dir>"
<cr> <lf>;
```

Example

```
DN 03-12-12 03:23:43 M CTAG2 COMPLD ”ACTUALDIR= B”;
```

Example

```
DN 03-12-12 03:23:43 M CTAG2 COMPLD;
```

Dependencies

This command must be rejected, if a connection has already been setup for the same session.

Change test direction - CHG-SLACC-DIR

Command CHG-SLACC-DIR allows a user to set the test direction to IN, OUT and BRIDGED. IN means connect the tip-ring test pair to the circuit on a line card. OUT means connect the tip-ring test pair to a subscriber loop. BRIDGED means connect the tip-ring test pair to the line. In the BRIDGED state, the high impedance isolation is not provided by the MTA at test bridged configuration for the Nortel product. This command is valid if CONN-SLACC-MET was executed successfully. The input, output and the behavior of this command are described in the following sections.

Input fields

Change test access direction (CHG-SLACC-DIR)	Description
Input	
AF (Address Format) (Mandatory input)	For DN access, enter DN. For LEN access, enter LEN.
DN, LEN (Mandatory input)	

Change test access direction (CHG-SLACC-DIR)	Description
DN format (# of digits)	If DN is entered, only numerical digits (0-9) are valid input. Min # of digits: 1 Max # of digits: 30 In order to support both North American and international DN format, the TL1 command parser should not reject the number of digits within MIN and MAX above. If the number of digits are invalid to the call server, it should be rejected by the call server.
LEN	This is the traditional Line Equipment Number stored in the CS 2000 Core. If LEN is entered, alphanumeric character values (0-9, A-Z, a-z) are valid input. The TL1 server converts all characters to upper case before processing the command. The LEN format in the call server will be used, i.e. SITE name + LEN number For example: CARY 000 1 01 12 CARY is the physical location where the line originates. 000: physical frame number in Cary 1: shelf number 01: slot number 12: circuit number leading zero (0) is required. For the Multiple Appearance Directory Number (MADN), line test access is via LEN only. ***Note:*** LENs must be enclosed in double quotes “”.
CTAG (Mandatory input)	Correlation tag (max 6 ASCII characters)
DIR (Test direction) (Mandatory input)	The test direction change will be rejected, if, for example, the line is in the CPB (Call Processing Busy) state, the test direction is in the BRIDGED state. Changing the test direction to OUT or IN will be rejected, since cut-off of the subscribers’ conversation is not allowed. If the test access direction change is rejected, the response must not produce an error message. It should response with COMPLD and with no error message. This behavior is consistent with the current

Change test access direction (CHG-SLACC-DIR)	Description
	no-test trunk interface behavior. And Access Care does not expect an error message, if the desired test access direction cannot be changed.
O (OUT)	Toward the subscriber loop. If the line is busy and the original state is BRIDGED, changing the line test direction to OUT is not allowed.
B (BRIDGED)	Tap the test pair from the test head to the subscriber line. High impedance isolation is not provided.
I (IN)	Toward the line circuit. If the line is busy and the original state is BRIDGED, changing the line test direction to IN is not allowed.

Input format and examples

Syntax

```
CHG-SLACC-DIR::<DN>|<LEN>:<ctag>::DIR=<dir>;
```

Example

```
CHG-SLACC-DIR::4166213000:CTAG2::DIR=I;
```

Output fields

Change test access direction (CHG-SLACC-DIR)	Description
Output	
CS (Command Status)	COMPLD: command is executed successfully. DENY: command is executed unsuccessfully.
CTAG	Correlation tag
ACTUALDIR (The actual connection direction)	MG 9000: I (IN), O (OUT), B (BRIDGED) ***Note:*** This parameter is populated only if the actual test direction set by the MG 9000 is different than requested.

Output format and examples

Syntax

```
<cr> <lf> <lf>
^^^<af>^<date>^<time> <cr> <lf>
M^^<ctag>^<cs> <cr> <lf>
^^^"ACTUALDIR=<dir>"
<cr> <lf>;
```

Example

```
DN 03-12-12 03:23:43 M CTAG2 COMPLD ”ACTUALDIR= B”
```

Example

```
DN 03-12-12 03:23:43 M CTAG2 COMPLD
```

Syntax

```
<cr><lf><lf>
^^^<af>^<date>^<time> <cr><lf>
M^^<ctag>^DENY <cr><lf>
^^^<TL1 ERROR CODE><cr><lf>
^^^[‘’<optional comma-delimited error parameters>"]<cr><lf
>
^^^/*<TL1 error description><cr><lf>*/
^^^/*<System-specific error description*/<cr><lf>
;
```

Change Test Access Bandwidth – CHG-SLACC-BW

Command CHG-SLACC-BW allows for changing the test access between narrowband access and wideband access. In MG 9000, the POTS and SAA cards have only one physical test access. It supports wideband testing. CHG-SLACC-BW will not physically change the test access relays. The 8x8 ADSL card has different test access. If the narrowband test access is requested, the test access point will be between the voice circuit and LPF (Low Pass Filter). In this case, the voice service will be cutoff and the ADSL service will still be available to the customer. If the wideband test access is requested, the cutoff relay is on the customer loop side. Both the voice and the ADSL service will be cutoff. The CHG-SLACC-BW command is independent from the current test direction. For example, if the current test direction is narrowband OUT, CHG-SLACC-BW will set the test access to wideband OUT and vise-versa. This logic applies to test direction IN and BRIDGED as well. If the line is voice busy during the initial test access, the test direction will be set to BRIDGED. CHG-SLACC-BW is allowed to switch between wideband BRIDGED and narrowband BRIDGED. However, changing CHG-SLACC-DIR to IN or OUT is not allowed in both cases.

Change test access direction (CHG-SLACC-BW)	Description
Input	
AF (Address Format) (Mandatory input)	For DN access, enter DN. For LEN access, enter LEN
DN, LEN (Mandatory input)	

Change test access direction (CHG-SLACC-BW)	**Description**
Input	
DN format (# of digits)	If DN is entered, only numerical digits (0-9) are valid input. Min # of digits: 1 Max # of digits: 30 In order to support both North American and international DN format, the TL1 command parser should not reject the number of digits within MIN and MAX above. If the number of digits is invalid to the call server, it should be rejected by the call server.
LEN format	This is the traditional Line Equipment Number stored in the CS 2000 Core. If LEN is entered, alphanumeric character values (0-9, A-Z, a-z) are valid input. The TL1 server converts all characters to upper case before processing the command. The LEN format in the call server will be used, for example, SITE name + LEN number For example: CARY 000 1 01 12 CARY is the physical location where the line originates. 000: physical frame number in Cary 1: shelf number 01: slot number 12: circuit number leading zero (0) is required. For Multiple Appearance Directory Number (MADN), line test access via LEN only. ***Note:*** LENs must be enclosed in double quotes “”
CTAG (Mandatory input)	Correlation tag (max 6 ASCII characters)
BW (Access Bandwidth) (Mandatory input)	

Change test access direction (CHG-SLACC-BW)	Description
Input	
NB	Narrowband. For the MG 9000 POTS and SAA card, there is only one test access path. NB or WB will result in the test access on the same path, i.e., no difference. For the 8x8 ADSL card on MG 9000, NB line test access allows for the narrowband test access. The test access point is between the POTS circuit card and LPF (Low Pass Filter), when the test direction is OUT (subscriber loop side), the test head sees the subscriber loop through LPF. This test access does not cut off the ADSL service.
WB	Wideband Currently this is for the 8x8 ADSL card on MG 9000 wideband access. The test access is at the subscriber loop side of the splitter. It cuts off both the POTS and ADSL services.

Input format and examples

Syntax

```
CHG-SLACC-BW::<DN>|<LEN>:<ctag>::BW=<bw>;
```

Example

```
CHG-SLACC-BW::4166213000:CTAG3::BW=NB;
```

Change test access direction (CHG-SLACC-BW)	Description
Output	
CTAG	Correlation tag
CS (Command Status)	COMPLD: command is executed successfully. DENY: command is executed unsuccessfully.
BW (Access Bandwidth)	N for Narrowband. W for Wideband. Note: For the MG 9000 POTS and SAA card, there is only one test access path. NB or WB will result in the test access on the same path, or, no difference. For the 8x8 ADSL card on MG 9000, NB line test access allows for the narrowband test access. The test access point is between the POTS circuit card and LPF (Low Pass Filter), when the test direction is

Change test access direction (CHG-SLACC-BW)	Description
Output	
	OUT (towards the equipment side or the subscriber loop side), the test head sees the subscriber loop through LPF. This test access does not cut off the ADSL service. Currently this is for the 8x8 ADSL card on MG 9000 wideband access. The test access is at the subscriber loop side of the splitter. It cuts off both POTS and ADSL services.

Output format and examples

Syntax

```
<cr> <lf> <lf>
^^^<af>^<date>^<time> <cr> <lf>
M^^<ctag>^<cs>
<cr> <lf>;
```

Example

```
DN 03-12-12 03:23:43 M CTAG2 COMPLD ;
```

Error response

Syntax

```
<cr><lf><lf>
<^^^<af>^<date>^<time> <cr><lf>
M^^<ctag>^DENY <cr><lf>
^^^<TL1 ERROR CODE><cr><lf>
^^^["<optional quoted, comma-delimited error parameters
>"]<cr><lf>
^^^/*<TL1 error description><cr><lf>*/
^^^/*<System-specific error description>*/<cr><lf>
```

Dependencies

This command is valid after the successful execution of CONN-SLACC-MET.

Report status - REPT-STAT

Command REPT-STAT is used for refreshing the 180-second access disconnect timer kept by the gateway. If the gateway does not receive a valid command within the timeout period, it disconnects test access from the subscriber circuit that was specified in the initial CONN-SLACC-MET command.

Input fields

Report status (used for timeout refresh) (REPT-STAT)	Description
Input	
AF (Address Format) (Mandatory input)	For DN access, enter DN. For LEN access, enter LEN
DN, LEN, EPI (Mandatory input)	
DN format (# of digits)	If DN is entered, only numerical digits (0-9) are valid input. Min # of digits: 1 Max # of digits: 30 In order to support both North American and international DN format, the TL1 command parser should not reject the number of digits within MIN and MAX above. If the number of digits is invalid to the call server, it should be rejected by the call server.
LEN format	This is the traditional Line Equipment Number stored in the CS 2000 Core. If LEN is entered, alphanumeric character values (0-9, A-Z, a-z) are valid input. The TL1 server converts all characters to upper case before processing the command. The LEN format in the call server will be used, such as SITE name + LEN number For example: CARY 000 1 01 12 CARY is the physical location where the line originates. 000: physical frame number in Cary 1: shelf number 01: slot number 12: circuit number leading zero (0) is required. For Multiple Appearance Directory Number (MADN), line test access is via LEN only. ***Note:*** LENs must be enclosed in double quotes “”.
CTAG (Mandatory input)	Correlation tag (max 6 ASCII characters)

Input format and examples

Syntax

```
REPT-STAT::<DN>|<LEN>:<ctag>:;
```

Example

```
REPT-STAT::4166213000:CTAG4:;
```

Output fields

Report status (used for timeout refresh) (REPT-STAT)	Description
Output	
OK	
CTAG	Correlation tag

Output format and examples

Syntax

```
OK <ctag> <cr> <lf>
<
```

Example

```
OK CTAG2 <cr><lf>
<
```

Error response

Syntax

```
<cr><lf><lf>
^^^<af>^<date>^<time> <cr><lf>
M^^<ctag>^DENY <cr><lf>
^^^<TL1 ERROR CODE><cr><lf>
^^^["<optional comma-delimited error parameters>"]<cr><lf>
^^^/*<TL1 error description><cr><lf>*/
^^^/*<System-specific error description>*/<cr><lf>
```

Dependencies

This command is valid after the successful execution of CONN-SLACC-MET.

Disconnect the Test Connection – DISC-SLACC

The command DISC-SLACC disconnects the test connection. This command restores the circuit under test to its through state and opens all TAP connections to that access point. The input, output and the behavior of this command are described in the following sections.

Input fields

Disconnect a subscriber line from the test bus (DISC-SLACC)	Description
Input	
AF (Address Format) (Mandatory input)	For DN access, enter DN. For LEN access, enter LEN
DN, LEN (Mandatory input)	
DN format (# of digits)	If DN is entered, only numerical digits (0-9) are valid input. Min # of digits: 1 Max # of digits: 30 In order to support both North American and international DN format, the TL1 command parser should not reject the number of digits within MIN and MAX above. If the number of digits is invalid to the call server, it should be rejected by the call server.
LEN format	This is the traditional Line Equipment Number stored in CS 2000 Core. If LEN is entered, alphanumeric character values (0-9, A-Z, a-z) are valid input. The TL1 server converts all characters to upper case before processing the command. LEN format in call server will be used, such as SITE name + LEN number For example: CARY 000 1 01 12 CARY is the physical location where the line originates. 000: physical frame number in Cary 1: shelf number 01: slot number 12: circuit number leading zero (0) is required. For Multiple Appearance Directory Number (MADN), line test access via LEN only. ***Note:*** LENs must be enclosed in double quotes “”.
CTAG (Mandatory input)	Correlation tag (max 6 ASCII characters)

Input Format and Examples

Syntax

```
DISC-SLACC::<DN>|<LEN>:<ctag>:;
```

Example

```
DISC-SLACC::4166213000:CTAG4:;
```

Output fields

Disconnect a subscriber line from the test bus (DISC-SLACC)	Description
Output	
CTAG	Correlation tag Phase 1.1
CS (Command Status)	COMPLD: command is executed successfully. DENY: command is executed unsuccessfully. Phase 1.1

Output format and examples

Syntax

```
<cr> <lf> <lf>
^^^<af>^<date>^<time> <cr> <lf>
M^^<ctag>^<cs> <cr> <lf>
<cr> <lf>;
```

Example

```
DN 03-12-12 03:23:43 M CTAG4 COMPLD
```

Error response

Syntax

```
<cr><lf><lf>
^^^<af>^<date>^<time> <cr><lf>
M^^<ctag>^DENY <cr><lf>
^^^<TL1 ERROR CODE><cr><lf>
^^^["<optional comma-delimited error parameters>"]<cr><lf>
^^^/*<TL1 error description><cr><lf>
^^^<System-specific error description>*/<cr><lf>
```

Retrieve Line Information – RTRV-SLACC-INFO

Command RTRV-SLACC-INFO queries the CS 2000 to obtain information related to a line. This command does not operate the test access relays, such as the ones in MG 9000 MTA and line circuit. It can be executed without a connection up. The input, output and the behavior of this command are described in the following sections.

Input fields

Query line info (RTRV-SLACC-INFO)	Description
Input:	
AF (Address Format) (Mandatory input)	For DN access, enter DN. For LEN access, enter LEN
DN, LEN (Mandatory input)	
DN format (# of digits)	If DN is entered, only numerical digits (0-9) are valid input. Min # of digits: 1 Max # of digits: 30 In order to support both North American and international DN format, the TL1 command parser should not reject the number of digits within MIN and MAX above. If the number of digits are invalid to the call server, it should be rejected by the call server.
LEN format	If LEN is entered, alphanumeric character values (0-9, A-Z, a-z) are valid input. The TL1 server converts all characters to upper case before processing the command. The LEN format in the call server will be used, such as SITE name + LEN number For example: CARY 000 1 01 12 CARY is the physical location where the line originates 000: physical frame number in Cary 1: shelf number 01: slot number 12: circuit number leading zero (0) is required. For Multiple Appearance Directory Number (MADN), line test access via LEN only. ***Note:*** LENs must be enclosed in double quotes “”.
CTAG (correlation tag) (Mandatory input)	Correlation tag (max 6 ASCII characters)

Input format and examples

Syntax

```
RTRV-SLACC-INFO:<af>:<DN|LEN>:<ctag>;
```

DN access example

```
RTRV-SLACC-INFO:DN:4166213000:CTAG6;
```

LEN access example

```
RTRV-SLACC-INFO:LEN:”CARY 000 1 01 12”:CTAG6;
```

Output fields

Query line info (RTRV-SLACC-INFO)	Description
Output:	
CTAG	Correlation tag
CS (Command Status)	COMPLD: command is executed successfully. DENY: command is executed unsuccessfully.
NE Number	The NE Number is a numeric digit defined for each MG 9000 node. This ID is unique to all MG 9000 nodes associated with the same call server. It is limited to 3 digits. It is not enforced to make sure that it is in sequential order. For example, if the NE Number for the first installed MG 9000 node is defined as 0, the NE Number of the next installed MG 9000 node does not have to be 1, as long as it is not 0 (it has to be unique). This NE Number is displayed with label: NE Number with the MG 9000 node name (NE Name) (See the following screen shot). It is required that for the subsequent command request, NE Number and Tap are the mandatory input parameters for the commands after the test connection is setup.
GW_NAME (NE name)	Gateway name defined for the MG 9000. ***Note:*** This is the same as the NE name.
CLLI	CS 2000 CLLI code
LS (Line State)	Line status before the test access request.
LEN (Equipment Number)	Line Equipment Number (Site name and equipment number) Example: CARY 000 1 01 09
TYPE (Line Type)	Line Type as specified by the QDN/QLEN command Examples: POYS, ISDN, P-phone...
LF (Line Frame type)	For MG 9000 return UE9000MG This field is populated with the gateway type defined in the CMT database.
CARDCODE	Line card type as defined in the CS 2000 Examples: GWLPOT, RDTLSG, 6X21AC...

Query line info (RTRV-SLACC-INFO)	Description
Output:	
EP_NAME	End Point name on the gateway
OPTIONS	Line options on Call Server

Output format and examples

Syntax

```
<cr> <lf> <lf>
^^^<af>^<date>^<time> <cr> <lf>
M^^<ctag>^<cs> <cr> <lf>
^^^DN=<FQDN>, TYPE=<LineType>,LCC=<Line Class Code>,CARDCO
DE=<card code>,LEN=<LEN>,GW_NAME=<gatewayName>,EP_NAME=
<endpointName>,OPTIONS=<options>,CLLI=<CS2000CLLI>,
CLLI=<CS 2000 CLLI>,NE_NUM=<ne number>,LF=
<frame Type>NE_NAME=<ne name>,LS=<line state><cr> <lf>;
```

Output example

Example

```
DN 04-06-30 20:25:40 M  CTAG5 COMPLD
 "DN=1418400207,TYPE="SINGLE PARTY
LINE",LCC=IBN,CARDCODE=GWLPOT,LEN="UAIP  00 0 02
07",GW_NAME=UAIP000-0-0, EP_NAME=tp/02/07, OPTIONS=DGT,
CLLI=COMPACT3, NE_NUM=1,LF=UE9000MG,LS=IDL" ;
```

Error response

Syntax

```
<cr><lf><lf>
^^^<af>^<date>^<time> <cr><lf>
M^^<ctag>^DENY <cr><lf>
^^^<TL1 ERROR CODE><cr><lf>
^^^["<optional comma-delimited error parameters>"]<cr><lf>
^^^/*<TL1 error description><cr><lf>*/
^^^/*<System-specific error description>*/<cr><lf>
```

Dependencies

This command has no dependencies and can be executed at any time. The input DN or LEN can be the same as the DN or LEN of the active connection, but it is not required

Error code listings

The following tables list the TL1 error codes and descriptions generated by the TL1 Server and the MG 9000.

TL1 error code	TL1 category	TL1 error text	Description
ESPG	EQUIPAGE	SOFTWARE PROGRAM	An error occurred in the application software while processing a request. All accumulated line information up to the point of error will be displayed. A comment line will be displayed in the TL1 response message indicating the source of failure.
ICNV	INPUT	COMMAND NOT VALID	The command code entered is invalid for this interface.
IDNV	INPUT	DATA NOT VALID	One of the input fields contains invalid data. A comment line will be displayed in the TL1 response message indicating the source of failure.
IIAC	INPUT	INVALID ACCESS IDENTIFIER (AID)	An invalid DN or LEN has been found in the access identifier field of the request message. A comment line will be displayed in the TL1 response message indicating the source of failure.
IICM	INPUT	INVALID COMMAN D	The command entered is missing required fields. A comment line will be displayed in the TL1 response message indicating the source of failure.
IICT	INPUT	INVALID CORRELA TION TAG	The correlation tag entered violates the one of the following constraints: • must be non-null • must be alpha-numeric • must be between 1-6 characters in length A comment line will be displayed in the TL1 response message indicating the source of failure.
IISP	INPUT	INVALID SYNTAX OR PUNCTUATION	Missing or invalid command terminator.

TL1 error code	TL1 category	TL1 error text	Description
IITA	INPUT	INVALID TARGET IDENTIFIER	An invalid address format has been found in the target identifier field of the request message. A comment line will be displayed in the TL1 response message indicating the source of failure.
IPMS	INPUT	PARAMETER, MISSING	A required parameter has not been specified in the command entered. A comment line will be displayed in the TL1 response message indicating the source of failure.
IPNC	INPUT	PARAMETER NOT CONSISTENT	An entered parameter has both Positional and Named components. As per GR-831 syntax rules, a parameter components must be of the same type. A comment line will be displayed in the TL1 response message indicating the source of failure.
IPNV	INPUT	PARAMETER NOT VALID	An illegal parameter tag or value has been found in the command string. A comment line will be displayed in the TL1 response message indicating the source of failure
IPC	INPUT	SYNTAX PUNCTU ATION	A syntax error has been detected in the command string. An example would be a missing “around a quoted string. A comment line will be displayed in the TL1 response message indicating the source of failure
ITSN	INPUT	INVALID/INACTIVE TEST SESSION NUMBER	The test session for the DN or LEN entered does not exist. Either there has been no test session established yet, or the test session has been terminated due to an inactivity time-out. A comment line will be displayed in the TL1 response message indicating the source of failure

TL1 error code	TL1 category	TL1 error text	Description
SCSN	STATUS	INVALID COMMAN D SEQUENCE	A TL1 command has been entered for which a required requisite command has not yet been executed. For example, CHG-SLACC-DIR cannot be executed successfully until CONN-SLACC-MET has been run to establish a test session. A comment line will be displayed in the TL1 response message indicating the source of failure
SDBE	STATUS	INTERNAL DATABASE ERROR	An internal database error has been detected during command execution. A comment line will be displayed in the TL1 response message indicating the source of failure
SNVS	STATUS	NOT IN VALID STATE	The target circuit is not in a state valid for line test. A comment line will be displayed in the TL1 response message indicating the source of failure
SROF	STATUS	REQUESTED OPERATION FAILED	This is a general error indicating that the request could not be processed successfully. A comment line will be displayed in the TL1 response message indicating the source of failure.
SRQN	STATUS	INVALID REQUEST	The operation cannot be performed on the specified target. Possible reasons include the following: • Unsupported gateway type • NLT option on line (no line test) • line is specified by DN and is member of a MADN group.

TL1 error code	TL1 category	TL1 error text	Description
			A comment line will be displayed in the TL1 response message indicating the source of failure

TL1 error code	TL1 error text	MG 9000 error text (user visible)	MG 9000 error description
RCBY	CIRCUIT BUSY	“Line Busy, Try Again Later”	The MG sends this if the line appears to be in a call.
SNVS	NOT IN VALID STATE	“Line can’t be accessed. Check Line at Gateway Manager.”	The MG sends this if the linecard has been pulled, the line circuit is busy, or otherwise inaccessible.
ENEQ	NOT EQUIPPED	“Gateway Not Equipped for This Request. Check Gateway Manager”	The MG sends this when the required equipment is missing. Currently for MTA access, this may never be returned by the MG 9000. It might be a good return code if the EM finds that there is not a corresponding tap provisioned for the request.
RANB	ACCESS NETWORK BUSY	“A Gateway Test Resource is Busy. Try Again Later.”	This is returned when internal resources are all in use, such as verticals and shelf test buses, or if there is another test on the same linecard (there is only one test bus per line card)
SROF	REQUESTED OPERATION FAILED	“Problem with a Gateway Test Resource. Check Gateway Manager”	The MG sends this when there is nothing wrong with the request, but there is something wrong with required equipment on the MG 9000. Could not find a better match in the list.
EFON	FEATURE OPTION NOT PROVIDED	“The request or one of its requested options is not supported.”	For example, request for full-split would return not_supported.
SRQN	INVALID REQUEST	“Invalid Request. Check MTAPT and line exists, and for conflicting parameters.”	MG sends this for scenarios such as conflicting parameters, e.g., a test-out connection requested when seize is false (test-out interferes with call-p, so the line must be seized for test-out).

TL1 error code	TL1 error text	MG 9000 error text (user visible)	MG 9000 error description
SABT	ABORTED	“Test session abandoned, connection dropped. Check Test System.”	Line test session inactivity time-out expired on the gateway.
SABT	ABORTED	“The gateway encountered a problem. Try Again. If problem recurs, check line and test Resources at Gateway Manager.”	Line test session terminated by the gateway due to a maintenance event or failure.
SROF	REQUESTED OPERATION FAILED	“Try Again. If problem recurs, check line and test Resources at Gateway Manager.”	Miscellaneous internal system error.

Limitations and restrictions

As stated earlier, directory numbers must be specified in their fully qualified form (FQDN) for test commands that manipulate test connections on the MG 9000. This includes the following commands:

- CONN-SLACC-MET
- CHG-SLACC-DIR
- CHG-SLACC-BW
- REPT-STAT

Upgrade dependencies

This feature introduces a dependency in the upgrade sequence of the Access Care system with the rest of the Carrier VoIP components.

When upgrading from releases prior to (I)SN07 to (I)SN07 and greater, it is required that Access Care be upgraded ONLY after the CMT, MG 9000 Manager and MG 9000 upgrades are complete. Failing to do so will disable Line test capabilities between Access care and the MG 9000.

Glossary

Terminology and description of method parameters

gwcName

the external or public name to a GWC (Gateway Controller), e.g. GWC7 or CallServerName, GWC7. TYPE String.

mgName

the external name of the MG (Media Gateway), e.g. PVG03, PVG11. TYPE String.

RC

the return code, a numeric value indicating the result of the attempted operation. TYPE String.

MsgText

the appropriate text string that describes the result of the operation.

<Component> Error Code

the error code supplied by the component which generated the error. (For example, <Oracle-RC>17002</Oracle-RC>

<Component> ErrorText

the text string supplied by the component which generated the error.

TN

terminal number. The valid range is 1-n, where n = 24 for North American carriers (DS1) and n = 31 for International carriers (E1).

TID

Terminal Identifier - consisting of node number (1-4094) and terminal number (1-4094). This numeric identifier is allocated (one per endpoint) and managed by the GWC EM. The GWC EM also maintains a mapping of endpoints to TIDs. TYPE Integer

carrierName

Name of the carrier, following the appropriate endpoint naming convention for the chosen protocol. TYPE String.

endPointName

name of this endpoint; defined using naming convention for the associated protocol. TYPE String.

Definitions and abbreviations

Acronym	Definition
ADSL	Asymmetric Digital Subscriber Line
AFC	Advanced Fibre Communications
BCM	Business Communications Manager
CMT	CS 2000 Management Tools
CPB	Call Processing Busy
CS	Communication Server
CS 2000	Communication Server 2000
CORBA	Common Object Request Broker Architecture
DDMS	DMS Data Management System
DMS	Digital Multiplex System
DQoS	Dynamic quality of service
DS	Differentiated services field. An 8 bit field containing the 6 bit DSCP subfield.
DSCP	Diffserv Code Point. A 6 bit subfield of the DS field in every IP packet which identifies the Diffserv Per Hop Behavior. In IP version 4, the TOS byte is redefined to be the DSCP. In IP version 6, the Traffic Class octet is used as the DSCP.
DTD	Document Type Definition
DTC	Digital Trunk Controller
DTCI	ISDN Digital Trunk Controller
EM	Element Manager
EPID	End Point Identifier
GUI	Graphical User Interface
GW	Gateway
GWC	Gateway Controller
GWC EM	Gateway Controller Element Manager
HTML	HyperText Markup Language
IDL	Interface Definition Language
IP	Internet Protocol
LBL	Low Bandwidth Length
LPF	Low Pass Filter
LTC	Line/Trunk Controller
MG 9000	Media Gateway 9000
MG 9000 EM	Media Gateway 9000 Element Manager

Acronym	Definition
MG	Media Gateway
MGC	Media Gateway Controller
MP	Media Proxy
MTA	Multimedia Terminal Adapter
MTAP	Metallic Test Access Port on MG 9000
MTC	Maintenance
MTM	Metallic test access Module
NAT	Network Address Translator
NV	Network View
NN	Node Number
OSS	Operation Support System
OSSDI	OSS Data Interface
PEP	Policy Enforcement Point
PRI	Primary Rate Interface
RTU	Remote Test Unit
RU	Resource Usage
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSPFS	Succession Solaris Platform Foundation Software
SS7	Signaling System 7
TDM	Time Division Multiplexing
TID	Terminal Identifier
TL1	Transaction Language 1
TMM	Trunk Maintenance Manager
TN	Terminal Number
TSC	Test System Controller
UPC	User Privilege Code
USN	Unique Sequence Number
XML	Extensible Markup Language
XMLCP	XML Command Processor
XSD	XML Schema Definition
XSL	Extensible Style Sheet Language

SSH Product Comparison

OSSGate supports SSH to enable secure data transport. The table below provides a comparison of several SSH products, which are available in the market. This table is provided as a guidance for selection if needed.

Name	Platform	License	Protocol	Server	Authentication	KeyGen	Contact	Forwarding
AmigaSSH	Amiga	GNU Public License	SSH-1	No	Password, Public-key	ssh-keygen	www.lysator.liu.se/~lilja/amigassh	No
SSH	BeOS	Freeware	SSH-1	No	Password, Public-key, trusted-Host	ssh-keygen	www.bebits.com/app/703	Port, X
Java SSH	Java	Freely Distributable	SSH-1	No	Password, Public-key	N/A	www.cl.cam.ac.uk/~fapp2/software/ java-ssh	No
Mind Term	Java	GNU publicLicense	SSH-1	No	Password, Public-key, Trusted-host, TIS, sid-token	Yes	www.mindbright.se	Port, X
F-Secure SSH client	Mac, Windows	Commercial	SSH-1, SSH-2	No	Password, Public-key	Yes	www.f-secure.com	Port, X
SSH DOS	MS-DOS	GNU publicLicense	SSH-1	No	Password	No	www.vein.hu/ ~nagyd	No

Name	Platform	License	Protocol	Server	Authentication	KeyGen	Contact	Forwarding
SSH OS2	OS/2	Info Not Available	SSH-1	No	Password, Public-key, Trusted-host	Yes	ftp://ftp.cs.hut.fi/pub/ssh/old/os2	Port, X
Top Gun SSH	PalmOS	Free	SSH-1	No	Password	No	www.isac.cs.berkeley.edu/pilot	No
Ossh	Unix	BSD License	SSH-1	sshd	Password, Public-key, Trusted-host	ssh-keygen	ftp://ftp.nada.kth.se/pub/krypto/ossh	Port, X
FISH	VMS	Freeware	SSH-1	No	Password, Public-key, Trusted-host	Yes	www.free.lp.se/fish	No
sshexec.com	VMS	Freeware	SSH-1	Yes	Password, Public-key	Yes	www.er6.eng.ohio-state.edu/~jonesd/ssh	
AppGate	Windows, Unix, Mac	Commercial					www.appgate.com	
Metro State SSH	Windows	GNU public license	SSH-1	No	Password	No	http://csi.mscd.edu/MSSH	Port
Okhapkin Port	Windows	Free for non-commercial use	SSH-1, SSH-2	Sshd	Password, public-key, trusted-host	ssh-keygen	http://miracle.geol.msu.ru/sos	Port, X

Name	Platform	License	Protocol	Server	Authentication	KeyGen	Contact	Forwarding
PenguinNet	Windows	Shareware	SSH-1	No	Password, public-key, rhosts	Yes	www.siliconcircus.com	No
SSH Secure Shell	Windows	Free for non commercial use	SSH-2	No	Password, public-key	Yes	www.ssh.com	Port, X
SecureCRT	Windows	Commercial	SSH-1, SSH-2	No	Password, public-key, TIS	RSA, DSA	www.vandyke.com	Port, X
SecureFX	Windows	Commercial	SSH-2	No	Password, public-key	Yes	www.vandyke.com	No
Secure KoalaTerm	Windows	Shareware	SSH-1, SSH-2	No	Password, public-key	Yes	www.midasoft.com	No
TTSSH	Windows	Freely distributable	SSH-1	No	Password, public-key, trusted-host, TIS	No	www. zip.com/au/~roca/ttssh.html	No
Zoc	Windows	Commercial	SSH-1	No	Password	No	www.emtec.com/zoc	No
SshCE	Windows	Freeware	SSH-1	No	Password	No	www.movsoftware.com/sshce.htm	No

Provisioning Conventions for GWC, Media Gateways, VMG, and Endpoints

Media gateways

Capacities of MGs are defined in the profiles available for selection when the MG is associated to the GWC. Correct selection of capacity is important to get the maximum utilization of your GWC. For example, a trunking GWC supports a maximum of 4094 datafilled ports, however, some MG VSP configurations support a maximum of 1120 ports. Therefore to get maximum utilization of the GWC, one would provision 3 1120 port VSPs and 1 624 port VSPs. Changing capacity is supported. To get better utilization of the GWC, one could change reserved ports for a GWC to be lower than what is defined in the profile.

In (I)SN05 and later releases, MGs are also used to host anchor packet resources. We call this an Anchor Packet Gateway or APG. APGs can only be associated with GWCs that are APG capable.

Profile Name	Service type	Reserved port capacity	Protocol	OSSGate protocol	Protocol version
PVG15K_ ASPEN	trunk	1120	ASPEN	2	2.1
PVG15K_ MEGACO	trunk	1120	MEGACO	4	1.0
PVG15K_ 1000_ASPEN	trunk	1000	ASPEN	2	2.1
PVG15K_ 1000_ MEGACO	trunk	1000	MEGACO	4	1.0
PVG15K_ PARTIAL_ ASPEN	trunk	624	ASPEN	2	2.1
PVG15K_ PARTIAL_ MEGACO	trunk	624	MEGACO	4	1.0
PVG7K_ ASPEN	trunk	1008	ASPEN	2	2.1
PVG7K_ MEGACO	trunk	1008	MEGACO	4	1.0
PVG_VSP3_ ASPEN	trunk	2016	ASPEN	2	2.1
PVG_VSP3_ MEGACO	trunk	2016	MEGACO	4	1.0

Profile Name	Service type	Reserved port capacity	Protocol	OSSGate protocol	Protocol version
PVG_APG_ ASPEN	APG	1120	ASPEN	2	2.1
PVG_APG_ MEGACO	APG	1120	MEGACO	4	1.0
PVG_APG_ VSP3_ ASPEN	APG	2016	ASPEN	2	2.1
PVG_APG_ VSP3_ MEGACO	APG	2016	MEGACO	4	1.0

Profile name	MG category	GWC connectivity profile number
PVG15K_ ASPEN	large	46
PVG15K_ MEGACO	large	58
PVG15K_ 1000_ASPEN	large	46
PVG15K_ 1000_MEGACO	large	58
PVG15K_ PARTIAL_ASPEN	large	46
PVG15K_ PARTIAL_ MEGACO	large	58
PVG7K_ASPEN	large	46
PVG7K_MEGACO	large	58
PVG_VSP3_ASPEN	large	46
PVG_VSP3_MEGACO	large	58
PVG_APG_ASPEN	large	46
PVG_APG_MEGACO	large	58
PVG_APG_VSP3_ASPEN	large	46
PVG_APG_VSP3_MEGACO	large	58

Profile name	Tone data	{[Exec, Term],...}	{[Type, Capacity], ...}
TRUNKNA	NORTHAM	[(’DTCEX’,’PRAB’), (’DTCEX’, ’ABTRK’)]	[(trunks(2),4094), (largeGWs(8), 24)]
TRUNK250	NORTHAM	[(’UTR250’,’PRAB’), (’DTCEX’, ’ABTRK’)]	[(trunks(2),4094), (largeGWs(8), 24)]

Profile name	Tone data	{[Exec, Term],...}	{[Type, Capacity], ...}
TRUNKINTL	UKADSI	[(’DTCEX’,’PRAB’), (’DTCEX’, ’ABTRK’)]	[(trunks(2),4094), (largeGWs(8), 24)]
V52TRUNK	UKADSI	[(’POTSEX’,’POTS’), (’DTCEX’, ’ABTRK’)]	[(trunks(2),3968), (largeGWs(8), 24), (v52(17), 0)]

Profile name	Tone data	{[Exec, Term],...}	{[Type, Capacity], ...}
APG_WITH_RA	NORTHAM	[(’DTCEX’, ’ABTRK’)]	[(apgp(4),6048), (apgGWs(10), 3) (ra(13),0)]
APG	NORTHAM	[(’DTCEX’, ’ABTRK’)]	[(apg(4),6048), (apgGWs(10), 3)]

Media Gateway name

Up to 8 upper case characters starting with an alphabetical character, followed by alphas and numbers

Suggested format

```
PVG <n...n><lp>
```

where

<nn> is device number of the MG, which value should be any character in the range 0-9 or a-z or A-Z, or with ‘-’ or ‘_’ embedded in it.

where

<lp> is the number of the slot of the logical processor card for the VSP in the MG (recommended slots 6-11)

Example - PVG046

H.323 gateways

The maximum port or endpoint capacity for H.323 gateways varies depending on that vendor gateway type. Consult vendor documentation for details.

Profile name	Service type	Protocol	OSSGate protocol	Protocol version
NORTEL_BCM	H.323, ITRANS	H.323	6	4.0
SUCCESSION_1000	H.323, ITRANS	H.323	6	4.0
CISCO_2600	H.323, ITRANS	H.323	6	4.0
CISCO_3600	H.323, ITRANS	H.323	6	4.0
CISCO_AS5300	H.323, ITRANS	H.323	6	4.0

Profile name	Service type	Protocol	OSSGate protocol	Protocol version
WESTELL	H.323, ITRANS	H.323	6	4.0
H.323_PROXY	H.323, ITRANS	H.323	6	4.0

Profile name	MG catego ry	GWC connectivity profile number	Protocol	Protocol version (internal storage)
NORTEL_BCM	Large	61	H.323	4.0
SUCCESSION_1000	Large	62	H.323	4.0
CISCO_2600	Large	63	H.323	4.0
CISCO_3600	Large	64	H.323	4.0
CISCO_AS5300	Large	65	H.323	4.0
WESTELL	Large	66	H.323	4.0

Profile name	Tone data	([Exec, Term]...)	([Type, Capacity],...)
H.323_NA	NORTHAM	[('DTCEX', 'PRAB)] [('DTCEX', 'ABTRK')]	[(h323(21),1032), (largegws(8), 200)]
H.323_INTL	UKADSI	[('DTCEX','PRAB'), ('DTCEX', 'ABTRK')]	[(h323(21),1024), (largeGWs(8), 200)]

CVX gateways

Profile name	Service type	Reserved port capac ity	Protocol	OSSGate protocol	Protocol version
CVX1800_2688	Trunk	2688	DSMCC	3	5.2
CVX600_612	Trunk	612	DSMCC	3	5.2

Profile name	MG categor y	GWC connectivity profile number	Protocol	Protocol version (internal storage)
CVX1800_2688	Large	47	3	5.2
CVX600_612	Large	47	3	5.2

Other trunk gateways and Audio Servers

Profile Name	Service Type	Reserved Port Capacity	GUI Protocol	OSSGATE Protocol	Protocol Version
UAS	audio	0 (undefined)	MEGACO	4	1.0
AUDIOCODES	trunk	280	MEGACO	4	1.0
NUERA_GX_ ASPEN	trunk	2108	ASPEN	2	2.1
NUERA_GX_ MEGACO	trunk	2108	MEGACO	4	1.0
TGCP	trunk	-	TGCP	6	1.0

Profile name	MG category	GWC connectivity profile number	Protocol	Protocol version (internal storage)
UAS	audio	48	4	1.0
AUDIOCODES	large	54	4	1.0
NUERA_GX_ASPEN	large	59	2	2.1
NUERA_GX_ MEGACO	large	59	4	1.0
TGCP	large	60	6	1.0

Profile name	Tone data	([Exec, Term]...)	([Type,Capacity],...)
AUDCNTL	NORTHAM	[(’DTCEX’, ’ABTRK’)]	[(audio(3),4096), (audioGWs(9),16), (bct(14),0), (ipsec(15),0) (conferences(18),0), (announcements(19),0)]
AUDCNTLINTL	UKADSI	[(’DTCEX’, ’ABTRK’)]	[(audio(3),4096), (audioGWs(9),16), (bct(14),0),(ipsec(15),0) (conferences(18),0), (announcements(19),0)]

Media Gateway name

A domain name of the Media Gateway in the form of an absolute domain name including the hostname of the device and suitable for lookup using Directory Name Service (DNS). The name must no longer than 32 characters. An example would be:

```
uas1.ral5.vendor.net
```

Endpoint names

Endpoints are not specified during provisioning of Universal Audio Servers.

Other GWC profiles

Profile name	Tone data	Service type	Service type, capacity
BICC	NORTHAM	DTCEX, ABTRK	dpt(5), 4096 sipt(11), 0
AUDCNTL_RMGC	NORTHAM	DTCEX, ABTRK	audio (3), 4096; audioGWs(9), 16 bct (14), 0; ipsec (15), 0 kerberos (16), 0 conferences (18), 0 announcements (19), 0 rmgcGWs (20),115,000
AUDCNTL_RMGCINTL	UKADSI	DTCEX, ABTRK	audio (3), 4096; audioGWs(9), 16 bct (14), 0; ipsec (15), 0 kerberos (16), 0 conferences (18), 0 announcements (19), 0 rmgcGWs (20),115,000

Carrier name and carrier endpoint names

Carrier Endpoints are provisioned in groups representing E1 and DS1 levels. Provisioning at this level gives access to all timeslot for service provisioning. Services supported in since (I)SN04 include: ISUP trunking, PRI trunking, V5.2 service and APG service for bearer channel anchoring.

Carrier names are case sensitive on the MG and should be entered in upper case. Descriptions below include both endpoint groups and individual endpoint descriptions. Services can be applied at the DS1/E1 timeslot level.

In addition to the traditional trunking carrier groups, special H.323 carrier groups are supported using carrier operations.

E1

```
E1_<h1><h2>.<g>
```

where

<h1> is the LP (logical processor) number (or slot) of the E1: 1-15 (no padding)

where

<h2> is the E1 (two digit) number: 01-32

where

<g> is the channel number: 1-31 (no leading 0)

Example

```
Carrier name:  E1_220; Carrier Endpoint name (timeslot) 1
```

DS3

`DS3_<b1><b2>.<c>.<d>`

where

`<b1>` is the LP (logical processor) number (or slot) of the DS3 (recommended slots 2-5): 2-5

where

`<b2>` is the DS3 (single digit) port number: 0-1

where

`<c>` is the DS1 number within the DS3: 1-28

where

`<d>` is the channel in the DS1: 1-24

Example

```
Carrier name:  DS3_20.3 - Carrier Endpoint name (timeslot)
DS3_20.3.1
```

STM-1

When using ASPEN protocol:

`SM_<lp><pp><r>_412_<jj><k><l><m>.<ee>`

When using H.248/Megaco protocol

`STM/<lp>/<pp>/<r>/VC4VC12/<jj>/<k>/<l>/<m>/<ee>`

where

`<lp>` is the LP (logical processor) number (or slot) of the STM-1 interface: 1-15 (no padding) (recommended slots 2-5)

where

`<pp>` is the (two digit) port number: 00-03

where

`<r>` is the (one digit) STM level ID, STM-1:1, STM-4:2 (Only STM-1 is supported now, hence valid value is 1)

_412_

identifies the muxing type VC4VC12

where

`<jj>` is the (two digit) AUG number within the STM-n (Valid value 01 only, since STM-1 only is supported now)

where

`<k>` is the (one digit) TUG-3 number within a VC4: 1-3

where

`<m>` is the (one digit) TU number within a TU: 1-3

where

`<ee>` is the 1 or 2 digit number, with VS-12 channel/timeslot: 1-31 (no leading 0)

Example carrier name

```
SM_2011_412_01361
```

Example endpoint name

```
SM_2011_412_01361.1
```

Example

```
STM/2/1/1/VC4VC12/1/3/6/1
```

OC-n/STS-n

When using ASPEN protocol:

```
SS_<lp><pp><r>_VT15_<jj><l><m>.<e>
```

When using H.248/Megaco protocol:

```
STS/<lp>/<pp>/<r>/VT15/<jj>/<l>/<m>/<e>
```

where

`<lp>` is the LP (logical processor) number (or slot) of the OC-3 interface: 1-15 (no padding)(recommended slots 2-5)

where

`<pp>` is the (two digit) port number: 00-03

where

`<r>` is the 1 digit STS level ID, STS-3:3, STS-12:4 etc (valid value is 3 since only OC3 is supported as of now)

_VT15_

identifies the muxing type STS-1/VT15

where

`<jj>` is the (two digit) STS-1 number within the STS-n: (valid range is 01-03, since only OC3 is supported as of now)

where

`<l>` is the (one digit) VT group number within STS-1: 1-7

where

`<m>` is the (one digit) VT number within a VT: 1-4

where

`<e>` is the VT1.5 channel/timeslot: 1-24 (no leading 0)

Carrier name
`SS_2023_VT15_0161`

Endpoint name
`SS_2023_VT15_0161.1`

Example
`STS/2/1/3/VT15/1/6/1`

MG 9000 line gateways

Capacities of MGs are defined in the profiles available for selection when the MG is associated to the GWC. Therefore, selection of capacity is important to get the maximum utilization of your GWC.

For example, a line GWC supports a maximum of 6400 datafilled ports, however, MG 9000 VMGs support 512 circuits. Changing this capacity is not supported in before (I)SN04 A future release will allow lowering the reserved capacity for existing VMGs and to allow the customer to specify reserved port capacity upon entering the association.

Profile name	Service type	Reserved port capacity	GUI proto col	OSSGate protocol	Protocol version
UE9000MG	line	512	MEGACO	4	1.0
UE9000MG_IP (was: MG9K_IP)	large	512	MEGACO	4	1.0

Profile name	MG catego ry	GWC profile number	Protocol	Protocol version (internal storage)
UE9000MG	large	50	5	1.0
UE9000MG_IP (was: MG9K_IP)	large	55	5	1.0

Profile name	Tone data	([Exec, Term]...)	([Type,Capacity],...)
LARGE_LINENA	NORTHAA	[(’POTSEX’,’POTS’), (’KSETEX’, ’KEYSET’)]	[(lines(1),6400), (largeGWs(8),27)]
LARGE_LINEINTL	UKADSI	[(’POTSEX’,’POTS’), (’KSETEX’, ’KEYSET’)]	[(lines(1),6400), (largeGWs(8),27)]

Virtual Media Gateway (VMG) added after (I)SN07

The MG 9000 is comprised of one or more VMGs. This section documents the format for naming of a VMG.

- Format: SSSSXXX-Y-Z

- Where:
  - — SSSS is a 1 to 4 character SITE name (as provisioned in Table SITE on the XA-Core).
  - — XXX is the office frame number (000-511).
  - — Y is the logical frame (0-7 for VMG created prior to (I)SN07, fixed value of zero for VMG created in (I)SN07 and later).
  - — Z is the shelf (0-3).

Example

```
LAKE017-0-2
```

Virtual Media Gateway (VMG) added prior to (I)SN07

Up to 32 characters with the following format:

```
<string>/<frame number>/<shelf number>
```

where

`<string>` is MG9K_<n> where n is a number between 1 and 999 indicating the number of the MG 9000 network element in this call server network. <string> cannot contain a “/”, but can contain alphanumerical and “_”.

where

`<frame>` is a number: 0-7

where

`>shelf>` is a number: 0-3

Line termination endpoint names

```
tp/<slot>/<circuit>
```

where

`<slot>` is a 2 digit number: 02-21

where

`<circuit>` is a 2 digit number: 00-31

The tp must be lower case.

Example

```
tp/03/01
```

CICM or 3rd party large gateways

The characteristics of the CICM and AFC gateways are listed in the following table.

Gateway profile name	GW cate gory	Signal protocol	Protocol version	Protocol port	Service type	Port/EP capacity	GWC profile number
CICM	large	MEGAC O	1.0	2944	Line, ITRANS	3069	57
AFC	large	MEGAC O	1.0	2944	Line	1023	67

Media Gateway name

3rd party large line GW and (I)SN06 CICM gateway naming is a 1 to 32 character alpha-numeric string. In (I)SN07 the names for CICM were changed as shown below. AFC and other 3rd party large lane GW remain unchanged.

(I)SN07 CICM gateway naming convention

```
<name>-<nnn>
```

where

`<name>` is any alpha numeric string

where

`<nnn>` is a zero padded number that ranges from 000 to 511. This number matches the frame number seen in the assigned logical group.

AFC endpoint names

Endpoints for AFC gateways take the following format

```
tp/<nnnn>
```

where

`<nnnn>` is a zero padded number that range from 0000 to 1022.

CICM endpoint names

Endpoints for CICM gateways take the following format:

Format of terminations provisioned in (I)SN06 or earlier releases

```
tp/<nnnn>
```

where

`<nnnn>` is a number that ranges from 0 to 9999999. It is not zero padded.

Format of terminations provisioned in (I)SN07 or later releases

```
tp/<G><nnnn>
```

where

`<G>` is a number that ranges from 0-2

where

`<nnnn>` is a number that ranges from 0000 to 1022. It is zero padded.

MTA line gateways

Capacities of MGs are defined in the profiles available for selection when the MG is associated to the GWC. Therefore, selection of capacity is important to get the maximum utilization of your GWC.

For example, a line GWC supports a maximum of 6400 datafilled ports, however, MTA devices support varying numbers of endpoints. Initial selection of reserved capacity should reflect the maximum number of lines this customer will possible use. This will reserve capacity in the GWC and may therefore impact the number of GWCs that support a given office. This is required to avoid the need for “rehoming” MTAs when unused lines are put into service.

A future release will allow lowering the reserved capacity for existing VMGs and to allow the customer to specify reserved port capacity upon creating the association.

Profile name	Service type	Reserved port capacity	GUI protocol	OSSGate protocol	Protocol version
MOTOROLAMTA_1	line, DQOS	1	NCS	1	1.0
MOTOROLAMTA_2	line, DQOS	2	NCS	1	1.0
MOTOROLAMTA_4	line, DQOS	4	NCS	1	1.0
TOUCHTONE_NN01_1	line, DQOS	1	NCS	1	1.0
TOUCHTONE_NN01_2	line, DQOS	2	NCS	1	1.0
TOUCHTONE_NN01_3	line, DQOS	3	NCS	1	1.0
TOUCHTONE_NN01_4	line, DQOS	4	NCS	1	1.0

Profile name	Service type	Reserved port capacity	GUI protoc ol	OSSGate protocol	Protocol version
ARRIS_TOUCHTO NE_ NN01_4	line, DQO S	4	NCS	1	1.0
ARRIS_TOUCHTO NE_ NN02_4	line, DQO S	4	NCS	1	1.0

Profile name	MG catego ry	GWC connectivity profile number	Protocol	Protocol ver sion (internal storage)
MOTOROLAMTA_1	small	52	1	1.0
MOTOROLAMTA_2	small	52	1	1.0
MOTOROLAMTA_4	small	52	1	1.0
TOUCHTONE_NN01_1	small	45	1	1.0
TOUCHTONE_NN01_2	small	45	1	1.0
TOUCHTONE_NN01_3	small	45	1	1.0
TOUCHTONE_NN01_4	small	45	1	1.0
ARRIS_TOUCHTONE_ NN01_4	small	45	1	1.0
ARRIS_TOUCHTONE_ NN02_4	small	52	1	1.0

Profile name	Tone data	([Exec, Term]...)	([Type,Capacity],...)
SMALL_LINENA	NORTHAA	[('POTSEX','POTS'), ('KSETEX', 'KEYSET')]	[(lines(1),6400), (smallGWs(7),6400), (ipsec(15), 0), (kerberos(16),0), (dqos(6),20)]
SMALL_LINEINTL	UKADSI	[('UTR250','PRAB'), ('DTCEX', 'ABTRK')]	[(lineports(1),6400), (smalllinegws(7),6400), (ipsec(15), 0), (kerberos(16),0), (dqos(6),20)]

Media Gateway name

A domain name of the Media Gateway in the form of an absolute domain name including the hostname of the device and suitable for lookup using Directory Name Service (DNS). The name must contain a "." and be no longer than 32 characters. An example would be:

```
cust34671.rdu.attcable.net
```

Line termination endpoint names

```
aaln/<n>
```

where

<n> is a number: 1-4 depending on the MTA model and market

Note: The size of the media gateway is specified when choosing the media gateway profile. Size values reserve space on selected GWCs.

IAD line gateways

Capacities of MGs are defined in the profiles available for selection when the MG is associated to the GWC. Therefore, selection of capacity is important to get the maximum utilization of your GWC.

For example, a line GWC supports a maximum of 6400 datafilled ports, however, Integrated Access Devices (IADs) support varying numbers of endpoints. Initial selection of reserved capacity should reflect the maximum number of lines this customer will possible use. This will reserve capacity in the GWC and may therefore impact the number of GWCs that support a given office. This is required to avoid the need for “rehoming” IADs when unused lines are put into service.

A future release will allow lowering the reserved capacity for existing VMGs and to allow the customer to specify reserved port capacity upon creating the association.

Profile name	Service type	Reserved port capa city	GUI proto col	OSSGate protocol	GUI pro tocol version	OSSGate protocol version
MEDIATRIX _LINE_GW_4	line	4	MGCP	5	1.0	1.0
MEDIATRIX _LINE_GW_24	line	24	MGCP	5	1.0	1.0
ASKEY_LINE_GW_4	line	4	MGCP	5	1.0	1.0

Profile name	Service type	Reserved port capa city	GUI proto col	OSSGate protocol	GUI pro tocol version	OSSGate protocol version
ASKEY_ LINE_ GW_12	line	12	MGCP	5	1.0	1.0
ASKEY_ LINE_ GW_30	line	30	MGCP	5	1.0	1.0

Profile name	MG categor y	GWC connec tivity profile number	Protocol	Protocol version (internal storage)
MEDIATRIX_LINE_ GW_4	small	49	5	1.0
MEDIATRIX_LINE_ GW_24	small	49	5	1.0
ASKEY_LINE_GW_ 4	small	49	5	1.0
ASKEY_LINE_GW_ 12	small	49	5	1.0
ASKEY_LINE_GW_ 30	small	49	5	1.0
AMBIT_LINE_GW_ 16	small	49	5	1.0
MGCP_LINE_GW_ 1	small	49	5	1.0

Profile name	Tone data	([Exec, Term]..)	([Type,Capacity],.)
SMALL_LINENA	NORTHAA	[(’POTSEX’,’POTS’), (’KSETEX’, ’KEYSET’)]	[(lines),6400), (smallGWs(7),6400), (ipsec(15), 0), (kerberos(16),0), (dqos(6),20)]
SMALL_LINEINTL	UKADSI	[(’UTR250’,’PRAB’), (’DTCEX’, ’ABTRK’)]	[(lines(1),6400), (smallGWs(7),6400), (ipsec(15), 0), (kerberos(16),0), (dqos(6),20)]

Media Gateway name

A domain name of the Media Gateway in the form of an absolute domain name including the hostname of the device and suitable for lookup using Directory Name Service (DNS). The name must contain a "." and be no longer than 32 characters. An example would be:

```
cust34671.rdu.vendor.net
```

Line termination endpoint names

```
aaln/<n>
```

where

`<n>` is a number: 1-4 depending on the model and market

> ***Note:*** The size of the media gateway is specified when choosing the media gateway profile. Size values reserve space on selected GWCs.

New SIP information

A SIP endpoint is composed of a VMG and termination.

VMG format = 32 character free-form string (eg. vmgname-string)
termination format = SITE/FFF/G/TTtt

where

`SITE, FFF, G, and TTtt` values are as specified in the SS LEN format with the following conversion padding rules:

- The LEN's FFF value will always be zero-padded to 3 digits (e.g. FFF value "6" would be converted to endpoint/termination FFF value "006")
- The LEN's individual TT tt values will always be zero-padded to 2 digits when converted to an endpoint/termination name (e.g. TT tt value "2 7" would be converted to an endpoint/termination TTtt value of "0207").

Example

```
vmgname-string   site/000/0/0002
```

Other GWC profiles

SIP information

A SIP endpoint is composed of a VMG and termination.

VMG format = 32 character free-form string (e.g. vmgname-string)

termination format = SITE/FFF/G/TTtt

where

`SITE, FFF, G, and TTtt` value are as specified in the SS LEN format with the following conversion padding rules:

- The LEN's FFF value will always be zero-padded to 3 digits (e.g. FFF value "6" would be converted to endpoint/termination FFF value "006").

- The LEN’s individual TT tt values will always be zero-padded to 2 digits when converted to an endpoint/termination name (e.g. TT tt value “2 7” would be converted to an endpoint/termination TTtt value of “0207”).

Example
vmgname-string site/000/0/0002

VRDN

The following profiles are supported for VRDN Gateway Controllers.

Profile Name	Tone Data	([Exec, Term]..)	([Type,Capacity],.)
VRDN	NORTHAM	[(’DTCEX’, ’ABTRK’)]	[(vrdn(12),0)]
VRDNINTL	UKADSI	[(’DTCEX’, ’ABTRK’)]	[(vrdn(12),0)]

SIP-T

The following profiles are supported for SIP-T Gateway Controllers.

Profile Name	Tone Data	([Exec, Term]..)	([Type,Capacity],.)
SIP-T	NORTHAM	[(’DTCEX’, ’ABTRK’)]	[(dpt(5),4096), (sipt(11),0)]
SIP-TINTL	UKADSI	[(’DTCEX’, ’ABTRK’)]	[(dpt(5),4096), (sipt(11),0)]
SIP-T_APG	NORTHAM	[(’DTCEX’, ’ABTRK’)]	[(apg(4), 2016), (dpt(5),4096), (apgGWs(10),1), (sipt(11),0)]
SIP-T_APGINTL	UKADSI	[(’DTCEX’, ’ABTRK’)]	[(apg(4), 2016),(dpt(5),4096), (apgGWs(10),1), (sipt(11),0)]
SIP-T_APG_RA	NORTHAM	[(’DTCEX’, ’ABTRK’)]	[(apg(4), 2016), (dpt(5),4096), (apgGWs(10),1), (sipt(11),0), (ra(13), 0)]
SIP-T_APG_RAINTL	UKADSI	[(’DTCEX’, ’ABTRK’)]	[(apg(4), 2016), (dpt(5),4096), (apgGWs(10),1), (sipt(11),0), (ra(13), 0)]

Index

Carrier Voice over IP

OSSGate User Guide

Publication: NE10004-512
Document status: Preliminary
Document version: 10.03
Document date: 05/26/2006