

**Oracle Communications® ASAP™ Cartridge 1.0
GA Release for Sonus SGX 5-2-X**

Sonus SGX 5-2-X Cartridge Guide

Third Edition
July 2008

ORACLE®

Copyright and Trademark Information

Copyright © 1992, 2008, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited. The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

1. Cartridge Overview	1
Cartridge content	2
Prerequisites	2
About this guide	2
Services, features, and options	3
Hardware and software requirements	3
Network element (NE) interface	3
ASAP version	4
Connecting to the NE	4
Related documentations	4
2. Installing and Testing the Cartridge	5
Downloading the cartridge	5
Starting ASAP	6
Installing the cartridge using scripts	7
Uninstalling the cartridge using scripts	7
Testing the cartridge installation	8
Configuring loopback and live mode parameters	8
Modifying T_SONUS-SGX_5-2-X_HOST.xml	9
Testing the installation	10
Installation and deployment of the cartridge using Studio	11
Uninstallation and Undeployment of the cartridge using Studio	12
3. Atomic Service Description Layer (ASDL) Commands	13
ASDL commands	15
A_SONUS-SGX_5-2-X_ADD_CC-CLIENT	15
A_SONUS-SGX_5-2-X_ADD_ROUTE-SET	17
A_SONUS-SGX_5-2-X_DEL_CC-CLIENT	18
A_SONUS-SGX_5-2-X_DEL_ROUTE-SET	20
A_SONUS-SGX_5-2-X_DISABLE_ROUTE-SET	21
A_SONUS-SGX_5-2-X_ENABLE_ROUTE-SET	22
A_SONUS-SGX_5-2-X_MOD_ROUTE-SET	23
A_SONUS-SGX_5-2-X_QRY_ACTIVE-CLIENT	25
A_SONUS-SGX_5-2-X_QRY_CC-CLIENT	27
A_SONUS-SGX_5-2-X_QRY_CLIENT	29
A_SONUS-SGX_5-2-X_QRY_ROUTE-SET	31
User exit types	36
Understanding user exit type XML files	37
User defined ASDL exit types	38
UserExitType.xml	41
4. Service Definition	49
CSDL commands	51
C_SONUS-SGX_5-2-X_ADD_CC-CLIENT	51

C_SONUS-SGX_5-2-X_ADD_ROUTE-SET	52
C_SONUS-SGX_5-2-X_ADD_TRUNK-GROUP	54
C_SONUS-SGX_5-2-X_DEL_CC-CLIENT	56
C_SONUS-SGX_5-2-X_DEL_ROUTE-SET	57
C_SONUS-SGX_5-2-X_DEL_TRUNK-GROUP	57
C_SONUS-SGX_5-2-X_DISABLE_ROUTE-SET	59
C_SONUS-SGX_5-2-X_ENABLE_ROUTE-SET	59
C_SONUS-SGX_5-2-X_MOD_ROUTE-SET	60
C_SONUS-SGX_5-2-X_QRY_ACTIVE-CLIENT	61
C_SONUS-SGX_5-2-X_QRY_CC-CLIENT	61
C_SONUS-SGX_5-2-X_QRY_CLIENT	62
C_SONUS-SGX_5-2-X_QRY_ROUTE-SET	62
5. Configuring ASAP to Support Additional NE Instances	65
Extracting source files	67
Loading a new XML file	67
6. Appendix	69
Login and Logout of Termhandler	69

Cartridge Overview

ASAP cartridges are discrete software components that are developed for the ASAP product. An ASAP cartridge offers specific domain behavior on top of the core ASAP software, and provides the configuration that supports a set of services on a network element (NE).

An ASAP cartridge is not a stand-alone component, but operates in conjunction with the ASAP core product. ASAP cartridges offer the following benefits:

- ◆ **Reduced Time to Market** - time to market of new services is reduced through simplified development, implementation, and extension of cartridges on customer sites.
- ◆ **Extendable** - cartridges can be extended to include additional services and components that deliver business value, without requiring changes to the original cartridge.
- ◆ **Simplified Effort** - the effort and technical knowledge that is required to perform customizations is reduced.
- ◆ **Ease of Installation** - cartridges can be installed into an ASAP environment without interfering with the existing install base.

An ASAP cartridge can be used to configure ASAP to provision the following:

- ◆ NEs from a specific vendor, such as Nortel or Lucent.
- ◆ Technologies, such as Asynchronous Transfer Mode (ATM) and Frame Relay switches, or Internet Protocol (IP) routers.
- ◆ Services that are supported on the NE, such as ATM, IP Virtual Private Networks (VPN), Wireless, or Optical.



Cartridges are designed for a specific technology, software load, and service.

An ASAP cartridge supports a particular set of services on an NE. These services are independent of customer-specific service definitions. Professional Services or systems integrators can perform extensions to the cartridge to support customer-specific requirements.

For more information on extending a cartridge, refer to the *ASAP Cartridge Development Guide for Service Activation*.

Cartridge content

An ASAP cartridge contains the following:

- ◆ An interface to the NE
- ◆ A set of scripts, such as State Tables or Java methods
- ◆ A set of atomic actions in the form of Atomic Service Description Layer (ASDL) commands
- ◆ A set of Common Service Description Layer (CSDL) commands that form meaningful services
- ◆ Sample work orders
- ◆ Installation scripts

Prerequisites

System integrators such as managers, designers, programmers, and testers who are responsible for the adaptation and integration of ASAP-based solutions should use this manual as a reference. It assumes that readers possess the following skills:

- ◆ A knowledge of ASAP programming concepts
- ◆ A good working knowledge of the UNIX operating system
- ◆ A thorough understanding of service and network provisioning
- ◆ Familiarity with telecommunications

About this guide

This guide provides a detailed description of the Sonus SGX cartridge. It contains overview and technical information to assist with extending and integrating the cartridge into a customer environment.

The scope of this guide includes ASAP as it pertains to the use of this cartridge. It is not intended to be a complete ASAP reference guide.

For additional information when using this cartridge, refer to the following supporting documentation:

- ◆ **Activation documentation set**—for detailed information on the ASAP component.
- ◆ **ASAP Cartridge Development Guide for Service Activation**—for information on how to extend a cartridge.

The Sonus SGX cartridge provides the ASAP service configuration and network element (NE) interface to activate provisioning services on T_SONUS-SGX_5-2-X_HOST NEs.

Services, features, and options

This cartridge supports the following services:

Table 1: Supported services

Service	Description
Add Trunk Group	This service adds a trunk group.
Delete Trunk Group	This service deletes a trunk group.
Add Route Set	This service adds a route set.
Modify Route Set	This service modifies a route set.
Delete Route Set	This service deletes a route set.
Query Route Set	This service queries a route set.
Block Route Set	This service blocks a route set.
Un Block Route Set	This service unblocks a route set.
Add CC Client	This service adds a CC client.
Delete CC Client	This service deletes a CC client.
Query CC Client	This service queries a CC client.
Query Client	This service queries a client.
Query Active Clients	This service queries the active clients.

Hardware and software requirements

The following sections contain the high-level software and hardware environment requirements for provisioning services including:

- ◆ Network element (NE) interface
- ◆ ASAP version

Network element (NE) interface

The following database tables in SARM are configured to support the NE configuration:

- ◆ tbl_host_clli
- ◆ tbl_clli_route

- ◆ tbl_comm_param
- ◆ tbl_resource_pool
- ◆ tbl_ne_config

ASAP version

This cartridge was developed and tested using ASAP Activation 4.7.1.

For more information on the operating environment of this ASAP version, refer to the ASAP version 4.7.1 Release Record.

Connecting to the NE

This cartridge uses Telnet protocol to connect to the NE.

Related documentations

This cartridge was developed according to the following Network Element Provisioning Specifications:

- ◆ 550-00990_SGX_5.2.0_Operations_Guide_1.pdf

Installing and Testing the Cartridge

This chapter describes the following procedures related to installing and testing the cartridge:

- ◆ [Downloading the cartridge](#)
- ◆ [Installing the cartridge using scripts](#)
- ◆ [Uninstalling the cartridge using scripts](#)
- ◆ [Testing the cartridge installation](#)
- ◆ [Installation and deployment of the cartridge using Studio](#)
- ◆ [Uninstallation and Undeployment of the cartridge using Studio](#)

Downloading the cartridge

Before you install the cartridge, you must use the internet to download the cartridge's TAR file from Oracle's Customer Portal.

Use the following instructions to download, then unTAR the TAR file.

To download the TAR file

1. Login to Oracle MetaLink internet home page (<http://www.metalink.oracle.com>).
2. Download the cartridge patch to your workstation.

To unTAR the TAR file

1. On your workstation, create a repository directory—the naming of which is your choice.

```
mkdir <repository dir>
```

2. Untar SonusSGX_5_2_X_R1_0_0.<build number>.tar.

```
tar xvf SonusSGX_5_2_X_R1_0_0.<build number>.tar
```

The directory structure in the repository directory should look like the following illustration. (this illustration describes the minimum required structure; you can enhance this directory structure with additional directories based on your requirements and deliverables).

```
<repository_directory>  
  /README.txt  
  /installCartridge  
  /uninstallCartridge  
  /SONUS_SGX_5_2_X.sar
```

Starting ASAP

Before installing the cartridge, ensure that ASAP is running.

To start ASAP

1. To start ASAP, execute the following script:

```
start_asap_sys -d
```

2. Ensure the ASAP Daemon (DAEM\$ENV_ID) is running by checking the ASAP status using the ASAP script “status”.
3. Check whether the WebLogic instance for this ASAP environment is running. If not, start the WebLogic instance.

The *ASAP Administration Guide* contains more information on starting ASAP, the ASAP Daemon, and WebLogic.

Installing the cartridge using scripts

Run the installation script *installCartridge* to install the cartridge. The script executes the following tasks:

- ◆ Configures the Sonus SGX-specific NE using the SACT.
- ◆ Deploys the Sonus SGX cartridge service model (only if the Sonus SGX service model is not yet deployed) using the Service Activation Deployment Tool (SADT).
- ◆ Copies the Sonus SGX-specific jar files and the cpp library file to the ASAP environment.
- ◆ Loads the sample work orders to the SRP database.

For information on the SACT and the SADT, refer to the *ASAP Administration Guide*.

To install the cartridge

1. Run the *installCartridge* script. At the prompt, type:

```
installCartridge SONUS_SGX_5_2_X.sar
```

2. The script prompts you for the values of the following WebLogic login parameters:

- ◆ WebLogic Hostname
- ◆ WebLogic HTTP Port
- ◆ WebLogic Login User ID
- ◆ WebLogic Login Password

The script loads the NEP-NE configuration and the CSDL-ASDL configuration to the SARM database, and loads sample work orders to the SRP database. The script also copies the cartridge-specific jar files and cpp library file to the ASAP environment.

3. Copy `studio_2_6_0.jar` file to the `$ASAP_BASE/lib` directory.
4. Add `${ASAP_BASE}/lib/studio_2_6_0.jar` to the CLASSPATH in the JInterpreter file under `$ASAP_BASE/programs` directory.
5. Restart ASAP to upload the cartridge configuration into ASAP.

Uninstalling the cartridge using scripts

Run the uninstallation script *uninstallCartridge* to uninstall the Sonus SGX cartridge. The script executes the following tasks:

- ◆ Unconfigures Sonus SGX-specific NEs using the SACT.
- ◆ Undeploys the Sonus SGX cartridge service model (only if the Sonus SGX service model is already deployed) using the Service Activation Deployment Tool (SADT).
- ◆ Removes the Sonus SGX-specific jar files and cpp library file from the ASAP environment.

For more information on the SACT and the SADT, refer to the *ASAP Administration Guide*.

To uninstall the cartridge

1. Run the `uninstallCartridge` script. At the prompt, type:

```
uninstallCartridge SONUS_SGX_5_2_X.<timestamp>.sar
```

2. The script prompts you for the values of the following parameters:

- ◆ WebLogic Hostname
- ◆ WebLogic HTTP Port
- ◆ WebLogic Login User ID
- ◆ WebLogic Login Password

The script unloads the NEP-NE configuration and CSDL-ASDL configuration from SARM database. It also removes the cartridge specific jar files and cpp library file from the ASAP environment.

Testing the cartridge installation

To test this cartridge installation, you need to know about the network element (NE), services, and basic Activation configuration. You may need to perform adjustments to provision a service for a specific NE, network, or connectivity configuration.

You can test the cartridge installation using one of the following methods:

- ◆ **Loopback mode**—does not actually connect to or send commands to the NE.
- ◆ **Live mode**—connects to and sends commands to a live NE.

Configuring loopback and live mode parameters

Set the following variables to test the cartridge in loopback or live testing modes.

Loopback mode

Set the following parameter to test the cartridge in loopback mode.

Table 2: Loopback Mode Parameter Settings

Configuration Variable	Parameter Settings	Location
LOOPBACK_ON	1 (default setting)	ASAP.cfg

Live mode

Set the following parameter to test the cartridge in live mode.

Table 3: Live Mode Parameter Settings

Configuration Variable	Parameter Settings	Location
LOOPBACK_ON	0	ASAP.cfg

Communication parameters

The following are the list of parameters for the sample NE configuration XML used by SACT.

Table 4: Communication parameters

param_label	param_value	param_desc
HOST_IPADDR	12.120.202.44	Host IP address.
PORT	23	Telnet port used by the switch.
HOST_USERID	omni	Login user name.
HOST_PASSWORD	sonus	Password for the user.
OPEN_TIMEOUT	20	Connection timeout in seconds.
READ_TIMEOUT	30	Read timeout in seconds.
HOST_LOGIN_PROMPT	login:	Login prompt from the NE. Default value: login
HOST_PASSWORD_PROMPT	Password:	Password prompt from the NE. Default value: Password
PROMPT	>	Unix prompt.
RESPONSELOG	true	Flag to turn off or on the response log.
LOGIN_SLEEP	2	Sleep in seconds during login to GSX before getting the prompt.
PROVISIONING_PROMPT] #	Provisioning prompt.

Modifying T_SONUS-SGX_5-2-X_HOST.xml

Use the following procedure to modify T_SONUS-SGX_5-2-X_HOST.xml.

To modify T_SONUS-SGX_5-2-X_HOST.xml

1. Create a new source directory. You can give this directory any appropriate, meaningful name you want to.

```
mkdir <new_source_directory>
```

2. Copy SONUS_SGX_5_2_X.sar to this new source directory.

```
cp SONUS_SGX_5_2_X.sar ./<new_source_directory>
```

3. Change directory to <new_source_directory>.

```
cd <new_source_directory>
```

4. Un-jar SONUS_SGX_5_2_X.sar. This extracts the contents of the sar file.

```
jar xvf SONUS_SGX_5_2_X.sar
```

5. Edit T_SONUS-SGX_5-2-X_HOST.xml with the appropriate changes, which is available under <new_source_directory>/NetworkElements.

6. Create a new sar file at the <new_source_directory> level.

```
CreateSar $PWD
```

7. Uninstall the cartridge using SONUS_SGX_5_2_X.sar. (That is, use the original sar file that you copied in [Step 2](#) above—see “[Uninstalling the cartridge using scripts](#)” on [page 7](#) for uninstallation instructions).

8. After you uninstall the cartridge, rename the sar file, so you have a backup copy of it.

9. Copy the new sar file from <new_source_directory>.

10. Reinstall the cartridge (see “[Installing the cartridge using scripts](#)” on [page 7](#) for installation instructions).

Testing the installation

The following procedure describes the steps required to test the cartridge installation in loopback mode. We recommend that you perform the initial cartridge installation test in loopback mode.

To test in loopback mode

1. Stop ASAP by typing the following command at the UNIX prompt:

```
stop_asap_sys -d
```

2. Ensure loop back mode is on. See “[Loopback mode](#)” on [page 8](#) for a description of how to set the loop back parameter to “On”.

3. Start ASAP by typing:

```
start_asap_sys -d
```

4. Send the sample work orders through the SRP Emulator by typing:

```
run_suite $SRP <ctrl_password> <suite name>
```

You can locate the suite names by typing:

```
grep SUITE * | grep -v END
```

A list of all available suites appears.

For more information on the SRP Emulator, refer to the *ASAP Administration Guide*.

5. Verify the status of the sample work orders by typing:

```
asap_utils -d l
```

All successful work orders returns to the 104 state.

To view the sample work orders provided with this cartridge, refer to the Sonus SGX cartridge source.

Viewing the sample work orders

You can find the sample work orders under the **SampleWorkOrders** directory in the sar file. The following procedure describes how to view the sample work orders.

To view the sample work orders

1. Create a repository directory, copy the sar file to the new directory and un-jar the sar file, as described by [Step 1](#) through [Step 4](#) in “[Modifying T_SONUS-SGX_5-2-X_HOST.xml](#)” on [page 9](#).
2. Locate and view the sample work order files.

Installation and deployment of the cartridge using Studio

Before installing the cartridge, ensure weblogic and ASAP are started and running.

The following are the steps involved:

1. Open Studio in design perspective. Choose **Import** from the **File** menu and select **Activation Archive (SAR)** under **Studio Wizards** to import the sar file. Browse for the path to the sar file and click **Finish**.
2. Create a new **Service Activation Project**.
3. Define a new **NE Entity**, based on the **NE Template** contained in the cartridge provided by Oracle.
4. Ensure that the primary pool of the newly created NE is different from the NE template primary pool. You can modify it, if necessary.
5. Ensure that the test work order provided with the cartridge targets the newly defined NE. If not, then modify the test work orders file(s).

6. Create a new **Activation Environment Project** from the **Studio** menu. (Use Studio help for more information).
7. Create **Activation Environment** inside the **Activation Environment Project** and configure the **Connection Details** tab with your Environment ID, Activation version and weblogic data.
8. Connect to your environment using the **Connect** button.
9. Select the **Cartridge** tab of the **Activation Environment** and click **Add** to add your projects to the environment. The cartridge and the newly created **Service Activation** should appear in the **Cartridges** list.
10. Deploy the **NetworkActivation** (NA) cartridge provided by Oracle. (No NE information is to be deployed with this cartridge, therefore it isn't necessary to deploy the **NEP map** info).
11. Deploy the **Service Activation** (SA) project as follows:
 - On the **Cartridge** tab, select the necessary SA cartridge and click the **Deploy** button.
 - Select the **NEP Map** tab of the **Activation Environment**. Choose the necessary **NEP** server from the drop-down box of the **Network Element Processors** (Use Studio help for more information).
 - Select the SA cartridge from the **Network Element Processor Map** and click the **Deploy** button.
12. Verify the **SADT** console to confirm the installation.
13. Go to ASAP environment.
14. Copy `studio_2_6_0.jar` file to the `$ASAP_BASE/lib` directory.
15. Add `${ASAP_BASE}/lib/studio_2_6_0.jar` to the CLASSPATH in the JInterpreter file under `$ASAP_BASE/programs` directory.
16. Restart **ASAP** in order to start working with the cartridge.

Uninstallation and Undeployment of the cartridge using Studio

The following are the steps involved:

1. Connect to your environment using the **Connect** button.
2. Select the necessary cartridge from the **Environment Cartridge** list in Studio 2.6 and click the **Undeploy** button.
3. Verify the Environment Cartridge list. The Check Box with the name of the cartridge should be in unchecked state.

Atomic Service Description Layer (ASDL) Commands

ASDL commands represent a set of atomic actions that ASAP can perform on a network element (NE). ASAP can combine ASDLs to create meaningful services (CSDLs) within a cartridge.

This chapter presents detailed information on the ASDL parameters that we provide with this cartridge. The following table lists and describes the type of parameter information that is included.

Table 5: ASDL parameter information

Item	Description
Parameter Name	Identifies the parameter that is configured for the stated service.
Description	Describes the parameter.
Range	Describes or lists the range of values that can be used to satisfy this parameter.
Default Value	Configures a default value for the parameter so that it is not mandatory for the upstream system to provide a value.

Table 5: ASDL parameter information

Item	Description
Type	<p>Indicates one of the following parameter types:</p> <ul style="list-style-type: none"> ◆ S—Scalar, specifies the parameter label transmitted on the ASDL command. Scalar parameters are conventional name-value pair parameters. ◆ C—Compound, specifies the base name of the compound parameter transmitted on the ASDL command. A compound parameter contains structures or arrays of information that are represented by a particular structure name or compound parameter name. Each compound parameter can contain a large number of elements. If you use compound parameters, you only require a single entry in the ASAP translation tables to call the compound parameter and all its associated parameter elements. ◆ I—Indexed, identifies a parameter that contains a sequential numerical index value to tell the SARM that it should execute the same operation (for example, an ASDL command) for all occurrences of that index. Consequently, if there are several options on a particular CSDL command (OPT1, OPT2, OPT3, etc.), you can specify the OPT parameter as an indexed parameter. When you specify the OPT parameter as an indexed parameter, the SARM generates several occurrences of that same ASDL command and each command has a different value for the option being transmitted to the NEP. <p>For more information on parameter types, refer to the <i>ASAP Developer Reference</i>.</p>
Class	<p>Indicates one of the following parameter classifications:</p> <ul style="list-style-type: none"> ◆ R—Required scalar parameter ◆ O—Optional scalar parameter ◆ C—Required compound parameter ◆ N—Optional compound parameter ◆ M—Mandatory indexed parameter ◆ I—Optional indexed parameter ◆ S—Parameter count

For a detailed description of the Required and Optional parameter classifications, refer to the *ASAP Administration Guide*.

ASDL commands

This cartridge provides the following ASDL commands:

- ◆ A_SONUS-SGX_5-2-X_ADD_CC-CLIENT
- ◆ A_SONUS-SGX_5-2-X_ADD_ROUTE-SET
- ◆ A_SONUS-SGX_5-2-X_DEL_CC-CLIENT
- ◆ A_SONUS-SGX_5-2-X_DEL_ROUTE-SET
- ◆ A_SONUS-SGX_5-2-X_DISABLE_ROUTE-SET
- ◆ A_SONUS-SGX_5-2-X_ENABLE_ROUTE-SET
- ◆ A_SONUS-SGX_5-2-X_MOD_ROUTE-SET
- ◆ A_SONUS-SGX_5-2-X_QRY_ACTIVE-CLIENT
- ◆ A_SONUS-SGX_5-2-X_QRY_CC-CLIENT
- ◆ A_SONUS-SGX_5-2-X_QRY_CLIENT
- ◆ A_SONUS-SGX_5-2-X_QRY_ROUTE-SET

A_SONUS-SGX_5-2-X_ADD_CC-CLIENT

Adds a call control client.

It is implemented by the following Java method:

```
com.mslv.activation.cartridge.sonus.sgx.x5_2_x.cc_client.add.generated.AddCcClientProxy.execute
```

Table 6: A_SONUS-SGX_5-2-X_ADD_CC-CLIENT

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	Remote network element name.			S	R
HOST_NAME	Name of the host.	Should be in the format gAAANNa.		S	R
ALT_HOST_NAME	Name of the alternate host.	Should be in the format gAAANNa.		S	O
NODE	Termhandler node.			S	R
DUAL	Dual.	Default: Y		S	O
SERVICE	Name of the service.	Default: A7_ISUP_CC		S	O
CIC_START	Starting value of CIC.	Default: 0		S	R

Table 6: A_SONUS-SGX_5-2-X_ADD_CC-CLIENT

Parameter Name	Description	Range	Default Value	Type	Class
CIC_END	End value of CIC.	Default: 16383		S	R
REMOTE_PC	Remote point code.	Should be in the formate NNN-NNN-NNN.		S	R
ARGS	ARGS.	Default: "ac 1 act 5"		S	O
NODE_NAME	Name of the call control node.			S	O
UID	User ID.	Default: zero		S	O
GID	Group ID.	Default: zero		S	O
PROTOCOL	Protocol value TCP or UDP.	min value, max value, valid values, etc		S	O

MML Commands

```
CREATE-CC-CLIENT:HOST="<HOST_NAME>",UID=[UID],GID=[GID],SERVICE=[SERVICE],
NODE="[NODE_NAME]",CIC_START=<CIC_START>,CIC_END=<CIC_END>,RPC=<REMOTE_PC>,A
LTHOST="[ALT_HOST_NAME]",DUAL=[DUAL],PROTOCOL=[PROTOCOL],ARGS="[ARGS]";
```

Output Parameters

Success case:

- ◆ SGX_ADD_CC_CLIENT_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_ADD_CC_CLIENT_RETCODE,value <User defined exit type>.
```

- ◆ SGX_ADD_CC_CLIENT_RETURN_INFO as INFO parameter to display error description:

```
SGX_ADD_CC_CLIENT_RETURN_INFO,value <Description>.
```

Failure case:(ProvCartridgeException, IOException, TelnetException, Exception and any other failures)

- ◆ SGX_ADD_CC_CLIENT_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_ADD_CC_CLIENT_RETCODE,value <User defined exit type>.
```

- ◆ SGX_ADD_CC_CLIENT_RETURN_INFO as INFO parameter to display error description:

```
SGX_ADD_CC_CLIENT_RETURN_INFO,value <NE Response>.
```

A_SONUS-SGX_5-2-X_ADD_ROUTE-SET

Adds a route set.

It is implemented by the following Java method:

```
com.mslv.activation.cartridge.sonus.sgx.x5_2_x.route_set.add.generated.AddRouteSetP  
roxy.execute
```

Table 7: A_SONUS-SGX_5-2-X_ADD_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	Remote network element name.			S	R
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R
ROUTE1	Name of the first route.	A list of up to 4 identifiers, separated by ampersands (&).		S	O
NODE	Termhandler node.			S	R
ROUTE2	Name of the second route.	A list of up to 4 identifiers, separated by ampersands (&).		S	O
ROUTE3	Name of the third route.	A list of up to 4 identifiers, separated by ampersands (&).		S	O
ROUTE4	Name of the fourth route.	A list of up to 4 identifiers, separated by ampersands (&).		S	O
LOAD_SHARE	Specifies the load sharing between two routes in the route set.	YES or NO		S	O
REMOTE_PC	Remote point code.	Should be in the format NNN-NNN-NNN.		S	R

MML Commands

A7 nodes:

```
CREATE-RSET:RSET=<ROUTE_SET_NAME>,PC=<REMOTE_PC>,RTES=<ROUTE1>&
[ROUTE2] & [ROUTE3] & [ROUTE4];
```

For other SS7 logical nodes variants:

```
CREATE-RSET:RSET=<ROUTE_SET_NAME>,PC=<REMOTE_PC>,RTES=<ROUTE1>&
[ROUTE2] & [ROUTE3] & [ROUTE4],
LOADSHR=[LOAD_SHARE];
```

For all the IP logical nodes variants:

```
CREATE-RSET:RSET=<ROUTE_SET_NAME>,PC=<REMOTE_PC>;
```

Output Parameters

Success case:

- ◆ SGX_ADD_ROUTE_SET_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_ADD_ROUTE_SET_RETCODE,value <User defined exit type>.
```

- ◆ SGX_ADD_ROUTE_SET_RETURN_INFO as INFO parameter to display error description:

```
SGX_ADD_ROUTE_SET_RETURN_INFO,value <Description>.
```

Failure case: (ProvCartridgeException, IOException, TelnetException, Exception and any other failures)

- ◆ SGX_ADD_ROUTE_SET_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_ADD_ROUTE_SET_RETCODE,value <User defined exit type>.
```

- ◆ SGX_ADD_ROUTE_SET_RETURN_INFO as INFO parameter to display error description:

```
SGX_ADD_ROUTE_SET_RETURN_INFO,value <NE Response>.
```

A_SONUS-SGX_5-2-X_DEL_CC-CLIENT

Deletes an existing call control client.

It is implemented by the following Java method:

com.mslv.activation.cartridge.sonus.sgxx5_2_x.cc_client.del.generated.DelCcClientProxy.execute

Table 8: A_SONUS-SGX_5-2-X_DEL_CC-CLIENT

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	Remote network element name.			S	R
HOST_NAME	Name of the host.	Should be in the format gAAANNa.		S	R
REMOTE_PC	Remote point code.	Should be in the formate NNN-NNN-NNN.		S	R
CIC_START	Starting value of CIC.	Default: 0		S	R
CIC_END	End value of CIC.	Default: 16383		S	R
NODE	Termhandler node.			S	R
NODE_NAME	Name of the call control node.	Logical node name for call control services.		S	O
SERVICE	Name of the service.	Default: A7_ISUP_CC		S	O
UID	User ID.	Default: zero		S	O
GID	Group ID.	Default: zero		S	O

MML Commands

```
DELETE-CC-CLIENT:HOST="<HOST_NAME>",UID=[UID],GID=[GID],SERVICE=[SERVICE],
NODE=" [NODE_NAME]",CIC_START=<CIC_START>,CIC_END=<CIC_END>,RPC=<REMOTE_PC>;
```

Output Parameters

Success case:

- ◆ SGX_DEL_CC_CLIENT_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:


```
SGX_DEL_CC_CLIENT_RETCODE,value <User defined exit type>.
```
- ◆ SGX_DEL_CC_CLIENT_RETURN_INFO as INFO parameter to display error description:

```
SGX_DEL_CC_CLIENT_RETURN_INFO,value <Description>.
```

Failure case: (ProvCartridgeException, IOException, TelnetException, Exception and any other failures)

- ◆ SGX_DEL_CC_CLIENT_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_DEL_CC_CLIENT_RETCODE,value <User defined exit type>.
```

- ◆ SGX_DEL_CC_CLIENT_RETURN_INFO as INFO parameter to display error description:

```
SGX_DEL_CC_CLIENT_RETURN_INFO,value <NE Response>.
```

A_SONUS-SGX_5-2-X_DEL_ROUTE-SET

Deletes an existing route set.

It is implemented by the following Java method:

```
com.mslv.activation.cartridge.sonus.sgx.x5_2_x.route_set.del.generated.DelRouteSetProxy.execute
```

Table 9: A_SONUS-SGX_5-2-X_DEL_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	Remote network element name.			S	R
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R
NODE	Termhandler node.			S	R
NONCHECK	Ignores the dependence in switch software while deleting the routeset.	YES/NO		S	O

MML Commands

```
DELETE-RSET:RSET=<ROUTE_SET_NAME>, [NOCHECK];
```

Output Parameters

Success case:

- ◆ SGX_DEL_ROUTE_SET_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_DEL_ROUTE_SET_RETCODE,value <User defined exit type>.
```

- ◆ SGX_DEL_ROUTE_SET_RETURN_INFO as INFO parameter to display error description:

```
SGX_DEL_ROUTE_SET_RETURN_INFO,value <Description>.
```

Failure case: (ProvCartridgeException, IOException, TelnetException, Exception and any other failures)

- ◆ SGX_DEL_ROUTE_SET_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_DEL_ROUTE_SET_RETCODE,value <User defined exit type>.
```

- ◆ SGX_DEL_ROUTE_SET_RETURN_INFO as INFO parameter to display error description:

```
SGX_DEL_ROUTE_SET_RETURN_INFO,value <NE Response>.
```

A_SONUS-SGX_5-2-X_DISABLE_ROUTE-SET

Locks an existing route set.

It is implemented by the following Java method:

```
com.mslv.activation.cartridge.sonus.sgx.x5_2_x.route_set.disable.generated.DisableRouteSetProxy.execute
```

Table 10: A_SONUS-SGX_5-2-X_DISABLE_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	Remote network element name.			S	R
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R
NODE	Termhandler node.			S	R

MML Commands

```
INHIBIT-RSET:RSET=<ROUTE_SET_NAME>;
```

Output Parameters

Success case:

- ◆ SGX_DIS_ROUTE_SET_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_DIS_ROUTE_SET_RETCODE,value <User defined exit type>.
```

- ◆ SGX_DIS_ROUTE_SET_RETURN_INFO as INFO parameter to display error description:

```
SGX_DIS_ROUTE_SET_RETURN_INFO,value <Description>.
```

Failure case: (ProvCartridgeException, IOException, TelnetException, Exception and any other failures)

- ◆ SGX_DIS_ROUTE_SET_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_DIS_ROUTE_SET_RETCODE,value <User defined exit type>.
```

- ◆ SGX_DIS_ROUTE_SET_RETURN_INFO as INFO parameter to display error description:

```
SGX_DIS_ROUTE_SET_RETURN_INFO,value <NE Response>.
```

A_SONUS-SGX_5-2-X_ENABLE_ROUTE-SET

Unblocks an existing route set.

It is implemented by the following Java method:

```
com.mslv.activation.cartridge.sonus.sgx.x5_2_x.route_set.enable.generated.EnableRouteSetProxy.execute
```

Table 11: A_SONUS-SGX_5-2-X_ENABLE_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	Remote network element name.			S	R
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R
NODE	Termhandler node.			S	R

MML Commands

```
ALLOW-RSET:RSET=<ROUTE_SET_NAME>;
```

Output Parameters

Success case:

- ◆ SGX_ENA_ROUTE_SET_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_ENA_ROUTE_SET_RETCODE,value <User defined exit type>.
```

- ◆ SGX_ENA_ROUTE_SET_RETURN_INFO as INFO parameter to display error description:

```
SGX_ENA_ROUTE_SET_RETURN_INFO,value <Description>.
```

Failure case: (ProvCartridgeException, IOException, TelnetException, Exception and any other failures)

- ◆ SGX_ENA_ROUTE_SET_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_ENA_ROUTE_SET_RETCODE,value <User defined exit type>.
```

- ◆ SGX_ENA_ROUTE_SET_RETURN_INFO as INFO parameter to display error description:

```
SGX_ENA_ROUTE_SET_RETURN_INFO,value <NE Response>.
```

A_SONUS-SGX_5-2-X_MOD_ROUTE-SET

Modifies an existing route set.

It is implemented by the following Java method:

```
com.mslv.activation.cartridge.sonus.sgx.x5_2_x.route_set.mod.generated.ModRouteSetProxy.execute
```

Table 12: A_SONUS-SGX_5-2-X_MOD_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	Remote network element name.			S	R
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R
NODE	Termhandler node.			S	R
LINK_SET_NAME	Name of the link set.	Should be a character string.		S	O

Table 12: A_SONUS-SGX_5-2-X_MOD_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
PRIORITY	Specifies the priority of the route being added.	Numeric value from 1 to 4, with 1 being the highest priority.		S	O
LOAD_SHARE	Specifies the load sharing between two routes in the route set.	YES or NO		S	O

MML Commands

- ◆ If the user wants to add a linkset to the existing routeset using mod routeset service, the MML is:


```
CHANGE-RSET:RSET=<ROUTE_SET_NAME>,ADD=<LINK_SET_NAME>,PRIORITY=<PRIORITY>;
```
- ◆ If the user wants to delete a linkset from the existing routeset using mod routeset service, the MML is :


```
CHANGE-RSET:RSET=<ROUTE_SET_NAME>,DEL=<LINK_SET_NAME>;
```
- ◆ If the LOAD_SHARE parameter is present, the MML command is (For non-A7 variants only):


```
CHANGE-RSET:RSET=<ROUTE_SET_NAME>,LOADSHR=[LOAD_SHARE];
```

Note:

- ◆ User should provide either LINE_SET_NAME with/without PRIORITY for A7 or LOAD_SHARE for Non-A7 along with ROUTE_SET_NAME.
- ◆ User should disable the route set before modifying the route set.

Output Parameters

Success Case:

- ◆ SGX_MOD_ROUTE_SET_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:


```
SGX_MOD_ROUTE_SET_RETCODE,value <User defined exit type>.
```
- ◆ SGX_MOD_ROUTE_SET_RETURN_INFO as INFO parameter to display error description:


```
SGX_MOD_ROUTE_SET_RETURN_INFO,value <Description>.
```

Failure case: (ProvCartridgeException, IOException, TelnetException, Exception and any other failures)

- ◆ SGX_MOD_ROUTE_SET_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_MOD_ROUTE_SET_RETCODE,value <User defined exit type>.
```

- ◆ SGX_MOD_ROUTE_SET_RETURN_INFO as INFO parameter to display error description:

```
SGX_MOD_ROUTE_SET_RETURN_INFO,value <NE Response>
```

A_SONUS-SGX_5-2-X_QRY_ACTIVE-CLIENT

Queries an existing active client.

It is implemented by the following Java method:

```
com.mslv.activation.cartridge.sonus.sgx.x5_2_x.active_client.qry.generated.QryActiveClientProxy.execute
```

Table 13: A_SONUS-SGX_5-2-X_QRY_ACTIVE-CLIENT

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	Remote network element name.			S	R
NODE	Termhandler node.			S	R

MML Commands

```
DISPLAY-ACTIVE-CLIENT;
```

Output Parameters

Success case:

- ◆ SGX_QRY_ACT_CLIENT_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_QRY_ACT_CLIENT_RETCODE,value <User defined exit type>.
```

- ◆ SGX_QRY_ACT_CLIENT_RETURN_INFO as INFO parameter to display error description:

```
SGX_QRY_ACT_CLIENT_RETURN_INFO,value <Description>.
```

Failure case: (ProvCartridgeException, IOException, TelnetException, Exception and any other failures)

- ◆ SGX_QRY_ACT_CLIENT_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

SGX_QRY_ACT_CLIENT_RETCODE,value <User defined exit type>.

- ◆ SGX_QRY_ACT_CLIENT_RETURN_INFO as INFO parameter to display error description:

SGX_QRY_ACT_CLIENT_RETURN_INFO,value <NE Response>.

Return CSDL Parameters:

AC_SERVER_PROCESS[++] - For Server value(server) i.e. SERVER00, SERVER01

AC_CLIENT_HOST_NAME[++] - For Client host value(clienthost)

AC_ALT_HOST_NAME[++] - For Alt Host value(althost)

AC_UID[++] - For UID value(uid) – i.e 0

AC_GID[++] - For GID value(gid) – i.e 0

AC_SERVICE[++] - For Service value(service) – i.e A7_ISUP_CC

AC_CE0[++] - For Ce0 value(ce0) – i.e p is value

AC_CE1[++] - For Ce1 value(ce1) – i.e b is value

AC_SSN[++] - For ssn value(ssn) - i.e 0 is value

MML example and output:

Sent MML command #1 to labnetra3.PM, cmd[DISPLAY-ACTIVE-CLIENT;]

starting 5 min. timer . . .

1 [14 Feb 200-1 16:46:00]

DISPLAY-ACTIVE-CLIENT:

Active CSF Server Connections, 16:46: 14-Feb-2001

server

process client host alt host ssn uid gid service ce0 ce1

SERVER00 0 0 0 A7_ISUP_CC p b

CLIENT1:00 mns11 mns12 0 0 0 A7_ISUP_CC uu uu

CLIENT2:00 mns21 mns22 0 0 0 A7_ISUP_CC uu uu

or

server

process client host uid gid service ce0 ce1

SERVER00 newt.dgms.com 342 101 MML pc --

SERVER01 newt.dgms.com 342 101 EVENT pc --

SERVER02 newt.dgms.com 342 101 C7_TCAP pd bd

SERVER03 newt.dgms.com 342 101 A7_TCAP pc TU

Legend:

Primary/Backup State (per server CE)

p - primary copy

b - backup copy

T - transient primary/backup state

- - server process not running on this CE

LAN connectivity state (associate with clientHostName)

u - connected via LAN path

D - LAN connection Down

-- No Connection established for this LAN

Usage: Filename: - to execute from a file

MML command: - series of mml command(s)

OMNI [25 Jan 2000 14:51:23] #1:quit

A_SONUS-SGX_5-2-X_QRY_CC-CLIENT

Queries an existing call control client.

It is implemented by the following Java method:

```
com.mslv.activation.cartridge.sonus.sgx.x5_2_x.cc_client.qry.generated.QryCcClientPro
xy.execute
```

Table 14: A_SONUS-SGX_5-2-X_QRY_CC-CLIENT

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	Remote network element name.			S	R
NODE	Termhandler node.			S	R

MML Commands

```
DISPLAY-CC-CLIENT;
```

Output Parameters**Success case:**

- ◆ SGX_QRY_CC_CLIENT_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_QRY_CC_CLIENT_RETCODE,value <User defined exit type>.
```

- ◆ SGX_QRY_CC_CLIENT_RETURN_INFO as INFO parameter to display error description:

```
SGX_QRY_CC_CLIENT_RETURN_INFO,value <Description>.
```

Failure case: (ProvCartridgeException, IOException, TelnetException, Exception and any other failures)

- ◆ SGX_QRY_CC_CLIENT_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_QRY_CC_CLIENT_RETCODE,value <User defined exit type>.
```

- ◆ SGX_QRY_CC_CLIENT_RETURN_INFO as INFO parameter to display error description:

```
SGX_QRY_CC_CLIENT_RETURN_INFO,value <NE Response>.
```

Return CSDL Parameters:

- CC_CLIENT_HOST_NAME[++] - For Host value(HOST)
- CC_CLIENT_UID[++] - For UID value(UID)
- CC_CLIENT_GID[++] - For GID value(GID)
- CC_CLIENT_SERVICE[++] - For Service value(SERVICE)
- CC_CLIENT_NODE_NAME[++] - For Node value(NODE)
- CC_CLIENT_CIC_START[++] - For CIC start value(CIC_START)
- CC_CLIENT_CIC_END[++] - For CIC end value(CIC_END)
- CC_CLIENT_REMOTE_PC[++] - For Remote PC value(RPC)
- CC_CLIENT_ARGS[++] - For Args value (ARGS)

MML example and output:

```
Sent MML command #1 to cslab.PM, cmd[DISPLAY-CC-CLIENT;]
starting 5 min. timer...
1 [25 Jan 2000 14:51:13]
DISPLAY-CC-CLIENT;
Displayed file: orCCclient.1001
```

orCCclient.1001 content

```
HOST UID GID SERVICE NODE CIC_START CIC_END RPC ARGS
```

```
-----
mns11 0 0 A7_ISUP_CC a7n1 3401 3424 254-240-0 -ac 1 -act 5
mns21 0 0 A7_ISUP_CC a7n1 3401 3424 254-240-0 -ac 1 -act 5
mns11 0 0 A7_ISUP_CC a7n1 3401 3424 254-241-0 -ac 1 -act 5
mns21 0 0 A7_ISUP_CC a7n1 3401 3424 254-241-0 -ac 1 -act 5
mns11 0 0 A7_ISUP_CC a7n1 3401 3424 254-244-31 -ac 1 -act 5
mns21 0 0 A7_ISUP_CC a7n1 3401 3424 254-244-31 -ac 1 -act 5
```

```

mns11 0 0 A7_ISUP_CC a7n1 3401 3424 254-252-26 -ac 1 -act 5
mns21 0 0 A7_ISUP_CC a7n1 3401 3424 254-252-26 -ac 1 -act 5
mns11 0 0 A7_ISUP_CC a7n5 0 4095 254-241-0 -ac 1 -act 5

```

Legend

Primary/Backup State (per server CE)

p - primary copy

b - backup copy

T - transient primary/backup state

-- server process not running on this CE

LAN connectivity state (associate with clientHostName)

u - connected via LAN path

D - LAN connection Down

-- No Connection established for this LAN

Usage: Filename: - to execute from a file

MML command: - series of mml command(s)

OMNI [25 Jan 2000 14:51:23] #1:quit

A_SONUS-SGX_5-2-X_QRY_CLIENT

Queries an existing client.

It is implemented by the following Java method:

```

com.mslv.activation.cartridge.sonus.sgx.x5_2_x.client.qry.generated.QryClientProxy.execute

```

Table 15: A_SONUS-SGX_5-2-X_QRY_CLIENT

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	Remote network element name.			S	R
NODE	Termhandler node.			S	R

MML Commands

```

DISPLAY-CLIENT;

```

Output Parameters

Success case:

- ◆ SGX_QRY_CLIENT_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:
`SGX_QRY_CLIENT_RETCODE,value <User defined exit type>.`
- ◆ SGX_QRY_CLIENT_RETURN_INFO as INFO parameter to display error description:
`SGX_QRY_CLIENT_RETURN_INFO,value <Description>.`

Failure case:(ProvCartridgeException, IOException, TelnetException, Exception and any other failures)

- ◆ SGX_QRY_CLIENT_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:
`SGX_QRY_CLIENT_RETCODE,value <User defined exit type>.`
- ◆ SGX_QRY_CLIENT_RETURN_INFO as INFO parameter to display error description:
`SGX_QRY_CLIENT_RETURN_INFO,value <NE Response>.`

Return CSDL Parameters:

- CLIENT_HOST_NAME[++] - For Host value(HOST)
- CLIENT_UID[++] - For UID value(UID)
- CLIENT_GID[++] - For GID value(GID)
- CLIENT_SERVICE[++] - For Service value(SERVICE)
- CLIENT_ALT_HOST_NAME[++] - For Alt Host value(ALT_HOST)
- CLIENT_DUAL[++] - For Dual value(DUSL)
- CLIENT_PROTOCOL[++] - For Protocol value(PROTO)
- CLIENT_AUTHORIZATION[++] - For Authorization factor value(AUTH)
- CLIENT_ARGS[++] - For Args value(ARGS)

MML example and output:

```
Sent MML command #5 to labnetra3.PM, cmd[DISPLAY-CLIENT;]
starting 5 min. timer...
5 [09 Apr 2001 14:25:16]
DISPLAY-CLIENT;
Displayed file: smClientDB.2010
```

smClientDB.2010 file content

HOST UID GID SERVICE ALT_HOST DUAL PROTO AUTH ARGS

MNS11 0 0 A7_ISUP_CC MNS12 Y TCP SYS -ac 1 -act 5
MNS21 0 0 A7_ISUP_CC MNS22 Y TCP SYS -ac 1 -act 5

Legend:

Primary/Backup State (per server CE)

p - primary copy

b - backup copy

T - transient primary/backup state

- - server process not running on this CE

LAN connectivity state (associate with clientHostName)

u - connected via LAN path

D - LAN connection Down

-- No Connection established for this LAN

Usage: Filename: - to execute from a file

MML command: - series of mml command(s)

OMNI [25 Jan 2000 14:51:23] #5:quit

A_SONUS-SGX_5-2-X_QRY_ROUTE-SET

Queries route set details.

It is implemented by the following Java method:

com.mslv.activation.cartridge.sonus.sgx.x5_2_x.route_set.qry.generated.QryRouteSetProxy.execute

Table 16: A_SONUS-SGX_5-2-X_QRY_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	Remote network element name.			S	R
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R
NODE	Termhandler node.			S	R
REMOTE_PC	Remote point code.	Should be in the format NNN-NNN-NNN.		S	O

Table 16: A_SONUS-SGX_5-2-X_QRY_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
PRINT_OPTION	Specifies whether you want a printed copy of the Route Set display on the system printer or a selected device.	YES or NO		S	O

MML Commands

- ◆ If the user wants to get the current configured information for a route set using route set name, the MML is:

```
DISPLAY-RSET[:RSET=ROUTE_SET_NAME],PRT=[PRINT_OPTION];
```

- ◆ If the user wants to get the current configured information for a route set using remote point code, the MML is:

```
DISPLAY-RSET[:PC=REMOTE_PC],PRT=[PRINT_OPTION];
```

Note: When the parameter PC or RSET is omitted an identifier of ALL is assume and all route sets are displayed.

Output Parameters

Success case:

- ◆ SGX_QRY_ROUTE_SET_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_QRY_ROUTE_SET_RETCODE,value <User defined exit type>.
```

- ◆ SGX_QRY_ROUTE_SET_RETURN_INFO as INFO parameter to display error description:

```
SGX_QRY_ROUTE_SET_RETURN_INFO,value <Description>.
```

Failure case:(ProvCartridgeException, IOException, TelnetException, Exception and any other failures)

- ◆ SGX_QRY_ROUTE_SET_RETCODE as CSDL parameter to the SARM table TBL_SRQ_PARM:

```
SGX_QRY_ROUTE_SET_RETCODE,value <User defined exit type>.
```

- ◆ SGX_QRY_ROUTE_SET_RETURN_INFO as INFO parameter to display error description:

```
SGX_QRY_ROUTE_SET_RETURN_INFO,value <NE Response>.
```

Return CSDL Parameters:

ROUTESET[+].ROUTE_SET_NAME - For RouteSet name value(Name)
 ROUTESET[+].REMOTE_PC - For Point code value(DPC)
 ROUTESET[+].ROUTE_SET_STATE - For RouteSet State value(State)
 ROUTESET[+].ROUTE_SET_STATUS - For RouteSet Status value(Status)
 ROUTESET[+].ROUTING_METHOD - For Routing method value(Routing Method)
 ROUTESET[+].ROUTE[+] - For Route value(Name)
 ROUTESET[+].ROUTE_STATUS[+] - For Route Status value(Status)
 ROUTESET[+].CLS_NAME[+] - For Route Status value(Status)

MML example and output:

Sent MML command #22 to mtjnsgx01.PM, cmd[DISPLAY-RSET:RSET=ALL;]
 22 [15 Feb 2007 12:13:09]
 DISPLAY-RSET:RSET=ALL;

--- ROUTE SET ---

Name DPC State Status Routing Method

ROUTESET1 254-248-000 ACTIVE ac0 STP

--- ROUTES ---

Name CLS Name Status

LINKSET1 CLSET1 ax
 LINKSET2 CLSET1 Ax

--- ROUTE SET ---

Name DPC State Status Routing Method

More? [Yes/No] y
 ROUTESET2 254-249-000 ACTIVE ac0 STP

--- ROUTES ---

Name CLS Name Status

LINKSET2 CLSET1 Ax
 LINKSET1 CLSET1 ax

--- ROUTE SET ---

Name DPC State Status Routing Method

DCOSSRS 254-244-032 ACTIVE Ac0 Member

--- ROUTES ---

Name CLS Name Status

DCOSSLS Ax

More? [Yes/No] y

DCOSSLS Ax

--- ROUTE SET ---

Name DPC State Status Routing Method

ITNGFOURE 254-244-031 ACTIVE ac0 Member

--- ROUTES ---

Name CLS Name Status

LINKSET1 CLSET1 ax

LINKSET2 CLSET1 Ax

--- ROUTE SET ---

Name DPC State Status Routing Method

ROUTESET4 254-252-222 ACTIVE ac0 Member

More? [Yes/No] y

--- ROUTES ---

Name CLS Name Status

LINKSET1 CLSET1 ax

LINKSET2 CLSET1 Ax

--- ROUTE SET ---

Name DPC State Status Routing Method

ROUTESET5 229-253-222 BLOCKED Ac0 Member

--- ROUTES ---

Name CLS Name Status

LINKSET1 CLSET1 ax
LINKSET2 CLSET1 Ax

More? [Yes/No] y

--- ROUTE SET ---

Name DPC State Status Routing Method

MACN02T 254-255-255 BLOCKED Ac0 Member

--- ROUTES ---

Name CLS Name Status

LINKSET1 CLSET1 Ax
LINKSET2 CLSET1 Ax

--- ROUTE SET ---

Name DPC State Status Routing Method

MACN03T 1-001-001 BLOCKED Ac0 Member

--- ROUTES ---

More? [Yes/No] y

Name CLS Name Status

LINKSET1 CLSET1 Ax
LINKSET2 CLSET1 Ax

--- ROUTE SET ---

Name DPC State Status Routing Method

MACN04T 254-052-144 BLOCKED Ac0 Member

--- ROUTES ---

Name CLS Name Status

LINKSET1 CLSET1 Ax

LINKSET2 CLSET1 Ax

--- ROUTE SET STATUS LEGEND ---

More? [Yes/No] y

a - PC accessible A - PC inaccessible

c0 - route set not congested Cx - route set congested to level x

--- ROUTE STATUS LEGEND ---

a - link set available A - link set not available

x - transfer allowed X - transfer prohibited

Usage: Filename: - to execute from a file

MML command: - series of mml command(s)

OMNI [15 Feb 2007 12:11:33] #22:exit

User exit types

User exit types allow cartridge developers and systems administrators to map ASDL exit codes to one of the predefined base exit types. Base exit types determine the product behavior. Cartridges map return codes and status values from a network element to a user defined exit type.

Regular expressions (regex) are used to perform pattern searches on responses from network elements. The pattern is stored in "tbl_user_err" in the SARM database. The user exit type contains a regex pattern that is applied at runtime.

Regular expressions enable users to associate a series of responses to a specific base type. For example, a regular expression "6" can identify a pattern where any response with the character "6" followed by any number of characters will translate to base type of FAIL.

Regular expressions can also allow very specific searches within a response from a network element. Regular expressions are typically compiled before being executed. Compilation produces a binary version of the expression and ensures that the syntax of the regular expression is correct. This compilation occurs using SACT\SADT when user exit types are deployed into ASAP. If the syntax is deemed to be incorrect during compilation, SADT displays an error message and the deployment of the user exit type will fail.

For more information on pattern matching, refer to the *ASAP Developer Reference* and the *ASAP Administration Guide*.

Understanding user exit type XML files

```

...
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>DYNAMIC_SL</softwareLoad>
    <technology>DYNAMIC_VENDOR-DYNAMIC_TECH</technology>
  </neDescriptor>
  <searchPattern>SUCCESS.</searchPattern>1
  <userType>U_SUCCEED</userType>2
  <baseType>SUCCEED</baseType>3
  <description>The ASDL provisioning was successful</description>
</userDefinedExitType>
<userDefinedExitType>
  <searchPattern>90.</searchPattern>
  <userType>U_FAIL</userType>
  <baseType>FAIL</baseType>
  <description>The ASDL failed - fail the current order
    and stop processing.</description>
</userDefinedExitType>
<userDefinedExitType>
  <searchPattern>101-110[201-215]</searchPattern>4
  <userType>U_SOFT_FAIL</userType>
  <baseType>SOFT_FAIL</baseType>
  <description>The ASDL has encountered a soft failure. Processing will
    continue.</description>
</userDefinedExitType>
<userDefinedExitType>
  <searchPattern>801-850</searchPattern>5
  <userType>U_MINOR_ERROR</userType>
  <baseType>SOFT_FAIL</baseType>
  <description>The ASDL has encountered a soft failure. Processing will
    continue.</description>

```

1. Pattern searches accommodate situations in which responses from the device contain small variants that represent the same meaning. The user type contains an associated search pattern that is applied at runtime. Using regular expressions, you can default a series of responses. For example a regular expression "90." can specify a pattern where any response with the character "90" followed by any character will translate to base type of FAIL. If the regular expression is defined as "90*", then any response with the character "90" followed by any number of characters will translate to base type of FAIL
2. The user type that the search pattern maps to.
3. The base type that maps to the user type.
4. 101 to 110 and 201 to 215 will translate to a base type of SOFT_FAIL
5. 801-850 will translate to a base type of SOFT_FAIL. Note that the user type differs from the previous range.

```

</userDefinedExitType>
<userDefinedExitType>
  <searchPattern>251-275&&[^261-265]</searchPattern>1
  <userType>U_DELAYED_FAIL</userType>
  <baseType>DELAYED_FAIL</baseType>
  <description>The ASDL has failed during provisioning.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>BCS36</softwareLoad>
    <technology>NORTEL_DMS</technology>
    <neVendor>Nortel</neVendor>
  </neDescriptor>
  <searchPattern>*.</searchPattern>
  <userType>U_MAINTAIN</userType>
  <baseType>MAINTENANCE</baseType>
  <description>The ASDL will Wait until the NE comes out of
    Maintenance Mode</description>
</userDefinedExitType>

```

The previous code sample shows some typical search pattern examples. Some additional examples follow:

- ◆ `^.*\b(one|two|three)\b.*$` = matches a complete line of text that contains any of the words "one", "two" or "three"
- ◆ `^(?=.*?\bone\b)(?=.*?\btwo\b)(?=.*?\bthree\b).*` matches a complete line of text that contains all of the words "one", "two" and "three"
- ◆ `"[^\r\n]*"` matches a single-line string that does not allow the quote character to appear inside the string.
- ◆ `\bd{1,3}\.d{1,3}\.d{1,3}\.d{1,3}\b` matches any IP address.

For more information on search patterns, refer to <http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html>.

For more information on user exit types, refer to chapter 3 of the *ASAP Developer Reference*.

User defined ASDL exit types

The following table lists the user defined ASDL exit types.

Table 17: User defined ASDL exit types

Search pattern	User_type	Base_type	Description
<code>((?s).)*DENY((?s).)*EN EQ((?s).)*</code>	SGX_ENEQ_NOT_EXISTS	FAIL	Equipment not equipped.

1. 251 to 275 but not 261 to 265 will translate to a base type of DELAYED_FAILURE.

Table 17: User defined ASDL exit types

Search pattern	User_type	Base_type	Description
((?s).)*DENY((?s).)*ICNV((?s).)*	SGX_IPCMD_NOT_VAL	FAIL	Input command not valid.
((?s).)*DENY((?s).)*IISP((?s).)*	SGX_ERR_IISP	FAIL	Input invalid syntax.
((?s).)*DENY((?s).)*IDRG((?s).)*	SGX_ERR_OUT_OF_RANGE	FAIL	Input data with invalid range.
((?s).)*DENY((?s).)*IDNV((?s).)*	SGX_ERR_INVALID_VAL	FAIL	Input data not valid.
((?s).)*DENY((?s).)*IIFM((?s).)*	SGX_ERR_IIFM	FAIL	Invalid data format.
((?s).)*DENY((?s).)*SROF((?s).)*	SGX_ERR_SROF	FAIL	Status requested operation failed.
((?s).)*Ms*DENY((?s).)*	SGX_ERR_DENY	FAIL	Action was denied.
((?s).)*Error((?s).)*out of range((?s).)*	SGX_DATA_ERR	FAIL	Input data out of range error.
((?s).)*CE Hostname cannot start with an integer((?s).)*	SGX_CMD_ERR	FAIL	Command error with invalid input data.
((?s).)*Service((?s).)*unknown((?s).)*	SGX_ERR_CC_SRC_DATA	FAIL	Invalid data for provisioned service.
((?s).)*ERR_INTERNAL((?s).)*	SGX_ERR_INTERNAL	FAIL	This is an internal client API error.
((?s).)*Error((?s).)*cannot determine((?s).)*	SGX_ERR_IO	FAIL	Input data error.
((?s).)*TelnetException((?s).)*	SGX_TELEXCEPT	RETRY_DISS	Telnet exception on the NE.
((?s).)*IOException((?s).)*	SGX_IOEXCEPT	RETRY_DISS	IO exception on the NE.
((?s).)*MCOMPLETED((?s).)*	SGX_COMPLD	SUCCEED	Successfully provisioned route set CLI command on the NE.

Table 17: User defined ASDL exit types

Search pattern	User_type	Base_type	Description
((?s.)*Command accepted(?s.)*	SGX_ACCPTD	SUCCEED	Successfully provisioned call control client CLI command on the NE.
((?s.)*DENY(?s.)*ID NV(?s.)*already provisioned(?s.)*	SGX_RS_ADD_SOFT_FAIL	SOFT_FAIL	The NE command failed because the provisioning action tried adding an element that already exists, but continue processing.
((?s.)*not found in CC client DB(?s.)*	SGX_CC_DEL_SOFT_FAIL	SOFT_FAIL	The NE command failed because the provisioning action tried deleting an element that already deleted, but continue processing.
((?s.)*Usage:(?s.)*	SGX_QRY_SUCCEED	SUCCEED	Successfully provisioned query CLI command on the NE.
((?s.)*CMD OK(?s.)*	SGX_LB_SUCCEED	SUCCEED	The NE command succeeded in Loop Back mode.
((?s.)*ProvCartridgeException(?s.)*	SGX_PROVEXCEP	FAIL	The NE command was denied due to provisioning cartridge exception.
((?s.)*USER_EXIT_NOT_FOUND(?s.)*	SGX_NO_MATCH	FAIL	No match found.
((?s.)*already exists in CC client DB(?s.)*	SGX_CC_ADD_SOFT_FAIL	SOFT_FAIL	The NE command failed because the provisioning action tried adding an element that already exists, but continue processing.
((?s.)*DENY(?s.)*ID NV(?s.)*not provisioned(?s.)*	SGX_RS_DEL_SOFT_FAIL	SOFT_FAIL	The NE command failed because the provisioning action tried deleting an element that already deleted, but continue processing.
((?s.)*DENY(?s.)*IC NV(?s.)*already(?s.)*	SGX_SOFT_FAIL	SOFT_FAIL	The NE command failed because the provisioning action tried modifying an element that already modified, but continue processing.

Table 17: User defined ASDL exit types

Search pattern	User_type	Base_type	Description
((?s.)*Error((?s.)*CIC_START value > CIC_END value((?s.)*	SGX_GEN_ERR	FAIL	Data validation failed.

UserExitType.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<serviceModel xmlns:ude="http://www.mslv.com/studio/activation/model/
userDefinedExitType" xmlns:sm="http://www.metasolv.com/ServiceActivation/
2003/ServiceModel" xmlns="http://www.metasolv.com/ServiceActivation/2003/
ServiceModel" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <userDefinedExitType>
    <neDescriptor>
      <softwareLoad>5-2-X</softwareLoad>
      <technology>SGX</technology>
      <neVendor>SONUS</neVendor>
    </neDescriptor>
    <searchPattern>((?s.)*DENY((?s.)*ENEQ((?s.)*</searchPattern>
    <userType>SGX_ENEQ_NOT_EXISTS</userType>
    <baseType>FAIL</baseType>
    <description>Equipment not equipped</description>
  </userDefinedExitType>
  <userDefinedExitType>
    <neDescriptor>
      <softwareLoad>5-2-X</softwareLoad>
      <technology>SGX</technology>
      <neVendor>SONUS</neVendor>
    </neDescriptor>
    <searchPattern>((?s.)*DENY((?s.)*ICNV((?s.)*</searchPattern>
    <userType>SGX_IPCMD_NOT_VAL</userType>
    <baseType>FAIL</baseType>
    <description>Input command not valid.</description>
  </userDefinedExitType>
  <userDefinedExitType>
    <neDescriptor>
      <softwareLoad>5-2-X</softwareLoad>
      <technology>SGX</technology>
      <neVendor>SONUS</neVendor>
    </neDescriptor>
    <searchPattern>((?s.)*DENY((?s.)*IISP((?s.)*</searchPattern>
    <userType>SGX_ERR_IISP</userType>
    <baseType>FAIL</baseType>
    <description>Input invalid syntax.</description>
  </userDefinedExitType>
  <userDefinedExitType>
    <neDescriptor>
      <softwareLoad>5-2-X</softwareLoad>

```

```

        <technology>SGX</technology>
        <neVendor>SONUS</neVendor>
    </neDescriptor>
    <searchPattern>((?s).)*DENY((?s).)*IDRG((?s).)*</searchPattern>
    <userType>SGX_ERR_OUT_OF_RANGE</userType>
    <baseType>FAIL</baseType>
    <description>Input data with invalid range.</description>
</userDefinedExitType>
<userDefinedExitType>
    <neDescriptor>
        <softwareLoad>5-2-X</softwareLoad>
        <technology>SGX</technology>
        <neVendor>SONUS</neVendor>
    </neDescriptor>
    <searchPattern>((?s).)*DENY((?s).)*IDNV((?s).)*</searchPattern>
    <userType>SGX_ERR_INVALID_VAL</userType>
    <baseType>FAIL</baseType>
    <description>Input data not valid.</description>
</userDefinedExitType>
<userDefinedExitType>
    <neDescriptor>
        <softwareLoad>5-2-X</softwareLoad>
        <technology>SGX</technology>
        <neVendor>SONUS</neVendor>
    </neDescriptor>
    <searchPattern>((?s).)*DENY((?s).)*IIFM((?s).)*</searchPattern>
    <userType>SGX_ERR_IIFM</userType>
    <baseType>FAIL</baseType>
    <description>Invalid data format</description>
</userDefinedExitType>
<userDefinedExitType>
    <neDescriptor>
        <softwareLoad>5-2-X</softwareLoad>
        <technology>SGX</technology>
        <neVendor>SONUS</neVendor>
    </neDescriptor>
    <searchPattern>((?s).)*DENY((?s).)*SROF((?s).)*</searchPattern>
    <userType>SGX_ERR_SROF</userType>
    <baseType>FAIL</baseType>
    <description>Status requested operation failed.</description>
</userDefinedExitType>
<userDefinedExitType>
    <neDescriptor>
        <softwareLoad>5-2-X</softwareLoad>
        <technology>SGX</technology>
        <neVendor>SONUS</neVendor>
    </neDescriptor>
    <searchPattern>((?s).)*M\s*DENY((?s).)*</searchPattern>
    <userType>SGX_ERR_DENY</userType>
    <baseType>FAIL</baseType>
    <description>Action was denied.</description>
</userDefinedExitType>

```

```

<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*Error((?s).)*out of range((?s).)*
</searchPattern>
  <userType>SGX_DATA_ERR</userType>
  <baseType>FAIL</baseType>
  <description>Input data out of range error.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*CE Hostname cannot start with an
integer((?s).)*</searchPattern>
  <userType>SGX_CMD_ERR</userType>
  <baseType>FAIL</baseType>
  <description>Command Error with invalid input data.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*Service((?s).)*unknown((?s).)*</searchPattern>
  <userType>SGX_ERR_CC_SRC_DATA</userType>
  <baseType>FAIL</baseType>
  <description>Invalid data for provisioned service.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*ERR_INTERNAL((?s).)*</searchPattern>
  <userType>SGX_ERR_INTERNAL</userType>
  <baseType>FAIL</baseType>
  <description>This is an internal client API errorr. </description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>

```

```
<searchPattern>((?s).)*Error((?s).)*cannot determine((?s).)*
</searchPattern>
<userType>SGX_ERR_IO</userType>
<baseType>FAIL</baseType>
<description>Input data error.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*TelnetException((?s).)*</searchPattern>
  <userType>SGX_TELEXCEPT</userType>
  <baseType>RETRY_DIS</baseType>
  <description>Telnet exception on the NE.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*IOException((?s).)*</searchPattern>
  <userType>SGX_IOEXCEPT</userType>
  <baseType>RETRY_DIS</baseType>
  <description>IO Exception on the NE.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*M_COMPLETED((?s).)*</searchPattern>
  <userType>SGX_COMPLD</userType>
  <baseType>SUCCEED</baseType>
  <description>Successfully provisioned route set CLI command on the
  NE.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*Command accepted((?s).)*</searchPattern>
  <userType>SGX_ACCPTD</userType>
  <baseType>SUCCEED</baseType>
  <description>Successfully provisioned call control client CLI command
  on the NE.</description>
</userDefinedExitType>
```

```

<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*DENY((?s).)*IDNV((?s).)*already provisioned
  ((?s).)*</searchPattern>
  <userType>SGX_RS_ADD_SOFT_FAIL</userType>
  <baseType>SOFT_FAIL</baseType>
  <description>The NE command failed because the provisioning action
  tried adding an element that already exists, but continue processing.
  </description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*not found in CC client DB((?s).)*
  </searchPattern>
  <userType>SGX_CC_DEL_SOFT_FAIL</userType>
  <baseType>SOFT_FAIL</baseType>
  <description>The NE command failed because the provisioning action
  tried deleting an element that already deleted, but continue
  processing.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*Usage:((?s).)*</searchPattern>
  <userType>SGX_QRY_SUCCEED</userType>
  <baseType>SUCCEED</baseType>
  <description>Successfully provisioned query CLI command on the NE.
  </description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*CMD OK((?s).)*</searchPattern>
  <userType>SGX_LB_SUCCEED</userType>
  <baseType>SUCCEED</baseType>
  <description>The NE command succeeded in Loop Back mode.</description>
</userDefinedExitType>
</userDefinedExitType>

```

```
<neDescriptor>
  <softwareLoad>5-2-X</softwareLoad>
  <technology>SGX</technology>
  <neVendor>SONUS</neVendor>
</neDescriptor>
<searchPattern>((?s).)*ProvCartridgeException((?s).)*</searchPattern>
<userType>SGX_PROVEXCEP</userType>
<baseType>FAIL</baseType>
<description>The NE command was denied due to provisioning cartridge
exception.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*USER_EXIT_NOT_FOUND((?s).)*</searchPattern>
  <userType>SGX_NO_MATCH</userType>
  <baseType>FAIL</baseType>
  <description>No match found.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*already exists in CC client DB((?s).)*
</searchPattern>
  <userType>SGX_CC_ADD_SOFT_FAIL</userType>
  <baseType>SOFT_FAIL</baseType>
  <description>The NE command failed because the provisioning action
tried adding an element that already exists, but continue
processing.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
    <softwareLoad>5-2-X</softwareLoad>
    <technology>SGX</technology>
    <neVendor>SONUS</neVendor>
  </neDescriptor>
  <searchPattern>((?s).)*DENY((?s).)*IDNV((?s).)*not provisioned
((?s).)*</searchPattern>
  <userType>SGX_RS_DEL_SOFT_FAIL</userType>
  <baseType>SOFT_FAIL</baseType>
  <description>The NE command failed because the provisioning action
tried deleting an element that already deleted, but continue
processing.</description>
</userDefinedExitType>
<userDefinedExitType>
  <neDescriptor>
```

```
        <softwareLoad>5-2-X</softwareLoad>
        <technology>SGX</technology>
        <neVendor>SONUS</neVendor>
    </neDescriptor>
    <searchPattern>((?s).)*DENY((?s).)*ICNV((?s).)*already((?s).)*
</searchPattern>
    <userType>SGX_SOFT_FAIL</userType>
    <baseType>SOFT_FAIL</baseType>
    <description>The NE command failed because the provisioning action
    tried modifying an element that already modified, but continue
    processing.</description>
</userDefinedExitType>
<userDefinedExitType>
    <neDescriptor>
        <softwareLoad>5-2-X</softwareLoad>
        <technology>SGX</technology>
        <neVendor>SONUS</neVendor>
    </neDescriptor>
    <searchPattern>((?s).)*Error((?s).)*CIC_START value &gt; CIC_END
    value((?s).)*</searchPattern>
    <userType>SGX_GEN_ERR</userType>
    <baseType>FAIL</baseType>
    <description>Data validation failed.</description>
</userDefinedExitType>
</serviceModel>
```


Service Definition

The Sonus SGX cartridge contains a set of CSDLs that map to one or more ASDL commands. You can also create additional CSDLs that map to existing and newly-created ASDLs. An upstream system can assemble any of these CSDL commands onto a work order for provisioning.

This chapter presents detailed information on the CSDL parameters that we provide in this cartridge. The following table lists and describes the type of parameter information that is included.

Table 18: ASDL parameter information

Item	Description
Parameter Name	Identifies the parameter that is configured for the stated service.
Description	Describes the parameter.
Range	Describes or lists the range of values that can be used to satisfy this parameter.
Default Value	Configures a default value for the parameter so that it is not mandatory for the upstream system to provide a value.

Table 18: ASDL parameter information

Item	Description
Type	<p>Indicates one of the following parameter types:</p> <ul style="list-style-type: none"> ◆ S—Scalar, specifies the parameter label transmitted on the ASDL command. Scalar parameters are conventional name-value pair parameters. ◆ C—Compound, specifies the base name of the compound parameter transmitted on the ASDL command. A compound parameter contains structures or arrays of information that are represented by a particular structure name or compound parameter name. Each compound parameter can contain a large number of elements. If you use compound parameters, you only require a single entry in the ASAP translation tables to call the compound parameter and all its associated parameter elements. ◆ I—Indexed, identifies a parameter that contains a sequential numerical index value to tell the SARM that it should execute the same operation (for example, an ASDL command) for all occurrences of that index. Consequently, if there are several options on a particular CSDL command (OPT1, OPT2, OPT3, etc.), you can specify the OPT parameter as an indexed parameter. When you specify the OPT parameter as an indexed parameter, the SARM generates several occurrences of that same ASDL command and each command has a different value for the option being transmitted to the NEP. <p>For more information on parameter types, refer to the <i>ASAP Developer Reference</i>.</p>
Class	<p>Indicates one of the following parameter classifications:</p> <ul style="list-style-type: none"> ◆ R—Required scalar parameter ◆ O—Optional scalar parameter ◆ C—Required compound parameter ◆ N—Optional compound parameter ◆ M—Mandatory indexed parameter ◆ I—Optional indexed parameter ◆ S—Parameter count

For a detailed description of the Required and Optional parameter classifications, refer to the *ASAP Administration Guide*.

CSDL commands

This cartridge provides the following CSDL Commands:

- ◆ C_SONUS-SGX_5-2-X_ADD_CC-CLIENT
- ◆ C_SONUS-SGX_5-2-X_ADD_ROUTE-SET
- ◆ C_SONUS-SGX_5-2-X_ADD_TRUNK-GROUP
- ◆ C_SONUS-SGX_5-2-X_DEL_CC-CLIENT
- ◆ C_SONUS-SGX_5-2-X_DEL_ROUTE-SET
- ◆ C_SONUS-SGX_5-2-X_DEL_TRUNK-GROUP
- ◆ C_SONUS-SGX_5-2-X_DISABLE_ROUTE-SET
- ◆ C_SONUS-SGX_5-2-X_ENABLE_ROUTE-SET
- ◆ C_SONUS-SGX_5-2-X_MOD_ROUTE-SET
- ◆ C_SONUS-SGX_5-2-X_QRY_ACTIVE-CLIENT
- ◆ C_SONUS-SGX_5-2-X_QRY_CC-CLIENT
- ◆ C_SONUS-SGX_5-2-X_QRY_CLIENT
- ◆ C_SONUS-SGX_5-2-X_QRY_ROUTE-SET

C_SONUS-SGX_5-2-X_ADD_CC-CLIENT

Adds a call control client.

Table 19: C_SONUS-SGX_5-2-X_ADD_CC-CLIENT

Parameter Name	Description	Range	Default Value	Type	Class
ALT_HOST_NAME	Name of the alternate host.			S	O
ARGS	ARGS.	Default: "ac 1 act 5"		S	O
CIC_END	End value of CIC.	Default: 16383		S	R
CIC_START	Starting value of CIC.	Default: 0		S	R
DUAL	Dual.	Default: Y		S	O
GID	Group ID.	Default: zero.		S	O
HOST_NAME	Name of the host.	Should be in the format gAAANNa.		S	R
NE_ID_SONUS_SGX	Remote network element name.			S	R

Table 19: C_SONUS-SGX_5-2-X_ADD_CC-CLIENT

Parameter Name	Description	Range	Default Value	Type	Class
NODE	Termhandler node.			S	R
NODE_NAME	Name of the call control node.			S	O
PROTOCOL	Protocol value TCP or UDP.	min value, max value, valid values, etc		S	O
REMOTE_PC	Remote point code.	Should be in the format NNN-NNN-NNN.		S	R
SERVICE	Name of the service.	Default: A7_ISUP_CC		S	O
UID	User ID.	Default: zero.		S	O

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 20: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_ADD_CC-CLIENT	A_SONUS-SGX_5-2-X_ADD_CC-CLIENT

C_SONUS-SGX_5-2-X_ADD_ROUTE-SET

Adds a route set.

Table 21: C_SONUS-SGX_5-2-X_ADD_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
REMOTE_PC	Remote point code.	Should be in the format NNN-NNN-NNN.		S	R
LOAD_SHARE	Specifies the load sharing between two routes in the route set.	YES or NO		S	O

Table 21: C_SONUS-SGX_5-2-X_ADD_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
NE_ID_SONUS_SGX	Remote network element name.			S	R
NODE	Termhandler node.			S	R
ROUTE1	Name of the first route.	A list of up to 4 identifiers, separated by ampersands (&).		S	O
ROUTE2	Name of the second route.	A list of up to 4 identifiers, separated by ampersands (&).		S	O
ROUTE3	Name of the third route.	A list of up to 4 identifiers, separated by ampersands (&).		S	O
ROUTE4	Name of the fourth route.	A list of up to 4 identifiers, separated by ampersands (&).		S	O
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 22: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_ADD_ROUTE-SET	A_SONUS-SGX_5-2-X_ADD_ROUTE-SET

C_SONUS-SGX_5-2-X_ADD_TRUNK-GROUP

Adds a trunk group to the end point.

Table 23: C_SONUS-SGX_5-2-X_ADD_TRUNK-GROUP

Parameter Name	Description	Range	Default Value	Type	Class
ALT_HOST_NAME	Name of the alternate host.	Should be in the format gAAANNa.		S	O
ARGS	ARGS.	Default: "ac 1 act 5"		S	O
CIC_END	End value of CIC.	Default: 16383		S	R
CIC_START	Starting value of CIC.	Default: 0		S	R
REMOTE_PC	Remote point code.	Should be in the format NNN-NNN-NNN.		S	R
DUAL	Dual.	Default: Y		S	O
GID	Group ID.	Default: zero.		S	O
HOST_NAME	Name of the host.	Should be in the format gAAANNa.		S	R
LOAD_SHARE	Specifies the load sharing between two routes in the route set.			S	O
NE_ID_SONUS_SGX	Remote network element name.			S	R
NODE	Termhandler node.			S	R
NODE_NAME	Name of the call control node.			S	O
PROTOCOL	Protocol value TCP or UDP.	min value, max value, valid values, etc		S	O
REMOTE_PC	Remote point code.	Should be in the format NNN-NNN-NNN.		S	R
ROUTE1	Name of the first route.	A list of up to 4 identifiers, separated by ampersands (&).		S	O

Table 23: C_SONUS-SGX_5-2-X_ADD_TRUNK-GROUP

Parameter Name	Description	Range	Default Value	Type	Class
ROUTE2	Name of the second route.	A list of up to 4 identifiers, separated by ampersands (&).		S	O
ROUTE3	Name of the third route.	A list of up to 4 identifiers, separated by ampersands (&).		S	O
ROUTE4	Name of the fourth route.	A list of up to 4 identifiers, separated by ampersands (&).		S	O
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R
SERVICE	Name of the service.	Default: A7_ISUP_CC		S	O
UID	User ID.	Default: zero.		S	O

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 24: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_ADD_TRUNK-GROUP	A_SONUS-SGX_5-2-X_ADD_ROUTE-SET
	A_SONUS-SGX_5-2-X_ENABLE_ROUTE-SET
	A_SONUS-SGX_5-2-X_ADD_CC-CLIENT

C_SONUS-SGX_5-2-X_DEL_CC-CLIENT

Deletes an existing call control client.

Table 25: C_SONUS-SGX_5-2-X_DEL_CC-CLIENT

Parameter Name	Description	Range	Default Value	Type	Class
CIC_END	End value of CIC.	Default: 16383		S	R
CIC_START	Starting value of CIC.	Default: 0		S	R
GID	Group ID.	Default: zero.		S	O
HOST_NAME	Name of the host.	Should be in the format gAAANNa.		S	R
NE_ID_SONUS_SGX	Remote network element name.			S	R
NODE	Termhandler node.			S	R
NODE_NAME	Name of the call control node.			S	O
REMOTE_PC	Remote point code.	Should be in the formate NNN-NNN-NNN.		S	R
SERVICE	Name of the service.	Default: A7_ISUP_CC.		S	O
UID	User ID.	Default: zero.		S	O

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 26: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_DEL_CC-CLIENT	A_SONUS-SGX_5-2-X_DEL_CC-CLIENT

C_SONUS-SGX_5-2-X_DEL_ROUTE-SET

Deletes an existing route set.

Table 27: C_SONUS-SGX_5-2-X_DEL_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
NE_ID_SONUS_SGX	Remote network element name.			S	R
NODE	Termhandler node.			S	R
NONCHECK	Ignores the dependence in switch software while deleting the routeset.	YES/NO		S	O
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 28: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_DEL_ROUTE-SET	A_SONUS-SGX_5-2-X_DEL_ROUTE-SET

C_SONUS-SGX_5-2-X_DEL_TRUNK-GROUP

Removes a trunk group from an existing end point.

Table 29: C_SONUS-SGX_5-2-X_DEL_TRUNK-GROUP

Parameter Name	Description	Range	Default Value	Type	Class
CIC_END	End value of CIC.	Default: 16383		S	R
CIC_START	Starting value of CIC.	Default: 0		S	R
GID	Group ID.	Default: zero.		S	O

Table 29: C_SONUS-SGX_5-2-X_DEL_TRUNK-GROUP

Parameter Name	Description	Range	Default Value	Type	Class
HOST_NAME	Name of the host.	Should be in the format gAAANNa.		S	R
NE_ID_SONUS_SGX	Remote network element name.			S	R
NODE	Termhandler node.			S	R
NODE_NAME	Name of the call control node.			S	O
NONCHECK	Ignores the dependence in switch software while deleting the routeset.	YES/NO		S	O
REMOTE_PC	Remote point code.	Should be in the formate NNN-NNN-NNN.		S	R
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R
SERVICE	Name of the service.	Default: A7_ISUP_CC.		S	O
UID	User ID.	Default: zero.		S	O

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 30: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_DEL_TRUNK-GROUP	A_SONUS-SGX_5-2-X_DISABLE_ROUTE-SET
	A_SONUS-SGX_5-2-X_DEL_ROUTE-SET
	A_SONUS-SGX_5-2-X_DEL_CC-CLIENT

C_SONUS-SGX_5-2-X_DISABLE_ROUTE-SET

Blocks an existing route set.

Table 31: C_SONUS-SGX_5-2-X_DISABLE_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
NE_ID_SONUS_SGX	Remote network element name.			S	R
NODE	Termhandler node.			S	R
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 32: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_DISABLE_ROUTE-SET	A_SONUS-SGX_5-2-X_DISABLE_ROUTE-SET

C_SONUS-SGX_5-2-X_ENABLE_ROUTE-SET

Unblocks an existing route set.

Table 33: C_SONUS-SGX_5-2-X_ENABLE_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
NE_ID_SONUS_SGX	Remote network element name.			S	R
NODE	Termhandler node.			S	R
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 34: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_ENABLE_ROUTE-SET	A_SONUS-SGX_5-2-X_ENABLE_ROUTE-SET

C_SONUS-SGX_5-2-X_MOD_ROUTE-SET

Modifies an existing route set.

Table 35: C_SONUS-SGX_5-2-X_MOD_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
LINK_SET_NAME	Name of the link set	Should be a character string.		S	O
LOAD_SHARE	Specifies the load sharing between two routes in the route set.	YES or NO		S	O
NE_ID_SONUS_SGX	Remote network element name.			S	R
NODE	Termhandler node.			S	R
PRIORITY	Specifies the priority of the route being added.	Numeric value from 1 to 4, with 1 being the highest priority.		S	O
ROUTE_SET_NAME	Name of the route set.	Should be 7 character string.		S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 36: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_MOD_ROUTE-SET	A_SONUS-SGX_5-2-X_MOD_ROUTE-SET

C_SONUS-SGX_5-2-X_QRY_ACTIVE-CLIENT

Queries an existing active client.

Table 37: C_SONUS-SGX_5-2-X_QRY_ACTIVE-CLIENT

Parameter Name	Description	Range	Default Value	Type	Class
NE_ID_SONUS_SGX	Remote network element name.			S	R
NODE	Termhandler node.			S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 38: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_QRY_ACTIVE-CLIENT	A_SONUS-SGX_5-2-X_QRY_ACTIVE-CLIENT

C_SONUS-SGX_5-2-X_QRY_CC-CLIENT

Queries an existing call control client.

Table 39: C_SONUS-SGX_5-2-X_QRY_CC-CLIENT

Parameter Name	Description	Range	Default Value	Type	Class
NE_ID_SONUS_SGX	Remote network element name.			S	R
NODE	Termhandler node.			S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 40: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_QRY_CC-CLIENT	A_SONUS-SGX_5-2-X_QRY_CC-CLIENT

C_SONUS-SGX_5-2-X_QRY_CLIENT

Queries an existing client.

Table 41: C_SONUS-SGX_5-2-X_QRY_CLIENT

Parameter Name	Description	Range	Default Value	Type	Class
NE_ID_SONUS_SGX	Remote network element name.			S	R
NODE	Termhandler node.			S	R

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 42: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_QRY_CLIENT	A_SONUS-SGX_5-2-X_QRY_CLIENT

C_SONUS-SGX_5-2-X_QRY_ROUTE-SET

Queries route set information.

Table 43: C_SONUS-SGX_5-2-X_QRY_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
NE_ID_SONUS_SGX	Remote network element name.			S	R

Table 43: C_SONUS-SGX_5-2-X_QRY_ROUTE-SET

Parameter Name	Description	Range	Default Value	Type	Class
NODE	Termhandler node.			S	R
ROUTE_SET_NAME	Name of the route set.			S	R
REMOTE_PC	Remote point code.	Should be in the formate NNN-NNN-NNN.		S	O
PRINT_OPTION	Specifies whether you want a printed copy of the Route Set display on the system printer or a selected device.	YES or NO		S	O

Mapping to ASDLs

The following table illustrates the CSDL to ASDL mapping for this service.

Table 44: CSDL to ASDL Mapping

CSDL	ASDL
C_SONUS-SGX_5-2-X_QRY_ROUTE-SET	A_SONUS-SGX_5-2-X_QRY_ROUTE-SET

Configuring ASAP to Support Additional NE Instances

You can configure ASAP to support the T_SONUS-SGX_5-2-X_HOST - NEP configuration using the Service Activation Configuration Tool (SACT). Refer to the *ASAP Administration Guide* for more information.

Below is an example of the Activation.Configuration.XML file for the Sonus SGX cartridge.

```
<?xml version="1.0" encoding="UTF-8"?>
<activationConfig xmlns="http://www.metasolv.com/ServiceActivation/2003/ActivationConfig" xmlns:cfg="http://www.mslv.com/studio/activation/model/config" xmlns:route="http://www.mslv.com/studio/activation/model/routing" xmlns:sm="http://www.metasolv.com/ServiceActivation/2003/ServiceModel" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <connectionPool name="SGX52XPL">
    <device name="SGX52X_dev1">
      <environment/>
      <lineType>TELNET_CONNECTION</lineType>
    </device>
  </connectionPool>
  <element name="T_SONUS-SGX_5-2-X_HOST">
    <vendor>SONUS</vendor>
    <technology>SGX</technology>
    <softwareLoad>5-2-X</softwareLoad>
    <nepServerName>$NEP</nepServerName>
    <primaryPool>SGX52XPL</primaryPool>
    <maximumConnections>1</maximumConnections>
    <dropTimeout>2</dropTimeout>
    <spawnThreshold>6</spawnThreshold>
    <killThreshold>3</killThreshold>
    <routingElement name="T_SONUS-SGX_5-2-X_HOST"/>
    <communicationParameter>
      <label>HOST_IPADDR</label>
      <value>
        <value>12.120.202.44</value>
      </value>
      <description>Host IP address</description>
      <lineType>TELNET_CONNECTION</lineType>
    </communicationParameter>
    <communicationParameter>
      <label>PORT</label>
      <value>
        <value>23</value>
      </value>
    </communicationParameter>
  </element>
</activationConfig>
```

```

        <description>Telnet Port used by the switch</description>
        <lineType>TELNET_CONNECTION</lineType>
    </communicationParameter>
    <communicationParameter>
        <label>HOST_USERID</label>
        <value>
            <value>omni</value>
        </value>
        <description>Login User Name</description>
        <lineType>TELNET_CONNECTION</lineType>
    </communicationParameter>
    <communicationParameter>
        <label>HOST_PASSWORD</label>
        <value>
            <value>sonus</value>
        </value>
        <description>Password for the User</description>
        <lineType>TELNET_CONNECTION</lineType>
    </communicationParameter>
    <communicationParameter>
        <label>OPEN_TIMEOUT</label>
        <value>
            <value>20</value>
        </value>
        <description>Connection timeout in seconds</description>
        <lineType>TELNET_CONNECTION</lineType>
    </communicationParameter>
    <communicationParameter>
        <label>READ_TIMEOUT</label>
        <value>
            <value>30</value>
        </value>
        <description>Read timeout in seconds</description>
        <lineType>TELNET_CONNECTION</lineType>
    </communicationParameter>
    <communicationParameter>
        <label>HOST_LOGIN_PROMPT</label>
        <value>
            <value>login:</value>
        </value>
        <description>Login prompt from the NE, default value is
        login</description>
        <lineType>TELNET_CONNECTION</lineType>
    </communicationParameter>
    <communicationParameter>
        <label>HOST_PASSWORD_PROMPT</label>
        <value>
            <value>Password:</value>
        </value>
        <description>Password prompt from the NE, default value is
        Password</description>
        <lineType>TELNET_CONNECTION</lineType>
    </communicationParameter>
    <communicationParameter>
        <label>PROMPT</label>
        <value>
            <value>&gt;</value>

```

```

        </value>
        <description>Unix prompt</description>
        <lineType>TELNET_CONNECTION</lineType>
    </communicationParameter>
    <communicationParameter>
        <label>RESPONSELOG</label>
        <value>
            <value>>true</value>
        </value>
        <description>Flag to turn off or on Response log
        </description>
        <lineType>TELNET_CONNECTION</lineType>
    </communicationParameter>
    <communicationParameter>
        <label>LOGIN_SLEEP</label>
        <value>
            <value>2</value>
        </value>
        <description>Sleep in seconds during login to GSX before
        getting prompt</description>
        <lineType>TELNET_CONNECTION</lineType>
    </communicationParameter>
    <communicationParameter>
        <label>PROVISIONING_PROMPT</label>
        <value>
            <value>] #</value>
        </value>
        <description>Provisioning prompt</description>
        <lineType>TELNET_CONNECTION</lineType>
    </communicationParameter>
</element>
</activationConfig>

```

Extracting source files

Before you can access an XML file to modify it, you must extract it from the .sar file. Use the following procedure to extract source files from the sar file. (Use Studio help for more information).

To extract source files

1. Create a repository directory. Copy the .sar file to the new directory and un-jar the sar file.
2. After you un-jar the sar file, you can access the XML files.

Loading a new XML file

When you finish modifying an XML file, you must create a new sar file, then restart the cartridge using the new file.

Follow the instructions in [“Testing the installation”](#) on page 10, for directions on how to load a new XML file. (Use Studio help for more information).

Appendix

Login and Logout of Termhandler

This section describes the login and logout of termhandler method.

Login Termhandler:

This cartridge uses the java method `startTermhandler` (String `sonusSgxInvokeCommand`, String `newNode`) in `Connection` class for handing the start/login of the termhandler.

This method will do the following:

- ◆ Compares the new node with previous node value.
- ◆ If both the node values are same, then it will display the following message and it will not login again.

```
"Already logged in TermHandler node: xxxx, newNode: xxxx ..."
```

For example, if the previous node is "a7n1" and new/current node is "a7n1" the displayed message could be:

```
"Already logged in TermHandler node: a7n1, newNode: a7n1..."
```

- ◆ If the new/current node differs from previous node then it will logout from termhandler by sending "QUIT" command and it will override the previous node value with new node value. And again logs in by sending the command with new node value "termhandler -node xxxx -i"

For example, if the previous node is a7n1 and new node is a7n2, then the command could be:

```
"termhandler -node a7n2 -i"
```

- ◆ If any exceptions are caught while doing the above operations, then it will throw an exception with an error message.

Logout Termhandler:

This cartridge uses the java method `disconnect()` in `Connection` class to logout from termhandler.

This method will do the following:

- ◆ Sends the "QUIT" command for logout from termhandler.

- ◆ If any exceptions are caught while doing the above operation, then it will throw an `DisconnectException` with an error message.