Oracle Communications® ASAP™ Cartridge 1.0
GA Release for Gemplus OTA 2.6

Gemplus OTA (Over-The-Air Activation platform) 2.6 Cartridge Guide

Eight Edition
July 2008

Copyright and Trademark Information

Contents

1

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.

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 Gemplus OTA 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.

The Gemplus OTA cartridge provides the ASAP service configuration and network element (NE) interface to activate subscriber services on GEMP-OTA_2-6-HOST NEs.

Services, features, and options

This cartridge supports the following services:

Table 1: Supported services

Service	Description
Add subscriber on the OTA	This service creates a subscriber account on the Gemplus Over-The-Air Activation (OTA) platform.
Delete subscriber from the OTA	This service deletes a subscriber account from the Gemplus OTA platform.
Modify subscriber on the OTA	This service updates a subscriber account on the Gemplus OTA platform.
Activate subscriber on the OTA	This service activates a subscriber account on the Gemplus OTA platform.
Deactivate subscriber on the OTA	This service deactivates a subscriber account on the Gemplus OTA platform.
Query subscriber on the OTA	This service queries a subscriber account on the Gemplus OTA platform.
Add subscriber SIM card on the OTA	This service creates a subscriber SIM card on the Gemplus OTA platform.
Delete subscriber SIM card from the OTA.	This service deletes a subscriber SIM card from the Gemplus OTA platform.
Modify subscriber SIM card on the OTA.	This service updates a subscriber SIM card on the Gemplus OTA platform.
Activate subscriber SIM card on the OTA.	This service activates a subscriber SIM card on the Gemplus OTA platform.
Deactivate subscriber SIM card on the OTA.	This service deactivates a subscriber SIM card on the Gemplus OTA platform.
Query subscriber SIM card on the OTA.	This service queries a subscriber SIM card on the Gemplus OTA platform. This service returns the status of the subscriber's SIM card.
Add MSISDN to SIM Card	This service adds the MSISDN to the SIM card on the Gemplus OTA platform.

Table 1: Supported services

Service	Description
Delete MSISDN from SIM card	This service deletes the MSISDN from the SIM card on the Gemplus OTA platform.
Delete SIM card Security Data	This service deletes the SIM card security data from the Gemplus OTA platform.
Upload SIM card with SIM card Loader	This service uploads a file that consists SIM card settings
Add SIM card with SIM card Loader	This service creates a set of SIM cards
Update SIM card with SIM card Loader	This service updates a set of SIM cards
Delete SIM card with SIM card Loader	This service deletes a set of SIM card settings
Query All files with SIM card Loader	This service queries all the files that exist on the repository.
Upload SIM card security bundle with SIM card security loader	This service uploads a file that consists SIM card security bundle settings.
Add SIM card security bundle with SIM card security Loader	This service creates new SIM card security bundle settings information
Update SIM card security bundle with SIM card security Loader	Update existing SIM card security data settings information
Query All files with SIM card security Loader	This service queries all the files on the repository

Hardware and software requirements

The following sections contain the high-level software and hardware environment requirements for provisioning subscriber services on authentication center 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 Activation 5.

For more information on the operating environment of this ASAP version, refer to the ASAP Activation 5 Release Record.

Connecting to the NE

This cartridge uses CORBA protocol for connecting to the NE.

Related documentation

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

- OTAManager40Developingclientapps.pdf

2

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 can 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 you workstation, create a repository directory—the naming of which is your choice.

```
mkdir <repository dir>
```

2. Copy the TAR file into the repository directory.
3. Untar GEMPLUSOTA_2_6_X_R1_0_0.<buildId>.tar.

```
tar xvf GEMPLUSOTA_2_6_X_R1_0_0.<buildId>.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
/gemp_ota.properties
/GemplusOTA_2_6.sar
/NEP_GOTA_jinterpreter_template
```

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
   ```

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. You will find this script under /Gemplus_ota_2_6. The script executes the following tasks:

- Creates and configures a new customized NEP named NEP_GOTA.
- Configures the Gemplus OTA-specific NE using the SACT.
- Deploys the Gemplus OTA cartridge service model (only if the Gemplus OTA service model is not yet deployed) using the Service Activation Deployment Tool (SADT).
- Copies the Gemplus OTA-specific jar files 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

- Copy the tar file, which contains all the details of this cartridge to the user repository.
- Extract archive file using following command.

```
tar xvf GEMPLUSOTA_2_6_X_R1_0_0.b<x>.tar
```

- Before run script installCartridge, start ASAP, check if daemon is running, and weblogic. Both weblogic and daemon should be up.
- Extract the sar file:
  - Create new directory.
  - Copy sar file to that new directory.
  - Run command "jar xvf GemplusOTA_2_6.sar"
- Make sure that ports in gemp_ota.properties are not used, - run the netstat command.
- Run script installCartridge - "installCartridge GemplusOTA_2_6.sar"
- You will be prompted to enter NEP server name to be used, for that, enter the new NEP server's name as NEP_GOTA.
- You will be prompted to enter weblogic values for host, port, userID and password.
- Script will perform:
  - Creates a customized NEP.
  - Setup of ASAP NE configuration.
  - Service deployment.
  - Sample work orders deployment.
- After installation a new sar file will also be created, which will be used for uninstalling the Cartridge.
- The file - NEP_GOTA_jinterpreter_template extracted as part of the SAR file contains the Java properties settings.

- Update the Java properties in the new NEP's jinterpreter file - $ASAP_BASE/programs/<NEP_GOTA>_jinterpreter similar to the Java properties set in the NEP_GOTA_jinterpreter_template file.
- Deploy Visibroker libraries.

  Visibroker 5.2.1 jar and Gemplus OTA library files are not shipped with the cartridge.

  The following Gempus libraries should be copied from the Gemplus OTA platform into $ASAP_BASE/lib :

  - vbjorb.jar
  - vbsec.jar
  - lm.jar

  The following Gempus libraries should be acquired from the Gemplus and copied into $ASAP_BASE/lib :

  - TOPLink.jar
  - TOPLinkX.jar
  - Tools.jar
  - reuse.jar
  - basesystem.jar
  - common.jar
  - xmlwrappers.jar

  Add the above jar files in the CLASSPATH in the $ASAP_BASE/programs/<NEP_GOTA>_jinterpreter file.
- Add the GemplusOTA_2_6.jar found in $ASAP_BASE/lib in the CLASSPATH in the $ASAP_BASE/programs/<NEP_GOTA>_jinterpreter file.
- Copy studio_2_6_0.jar file to the $ASAP_BASE/lib directory.
- Add ${ASAP_BASE}/lib/studio_2_6_0.jar to the CLASSPATH in the JInterpreter file under $ASAP_BASE/programs directory.
- User needs to stop and restart ASAP.
- Then user can run sample work orders from the SampleWorkOrders directory using command "run_suite $SRP $CTRL_USER WO_ID"

Uninstalling the cartridge using scripts

Run the uninstallation script uninstallCartridge to uninstall the Gemplus OTA cartridge. This script is located under /Gemplus_ota_2_6. The script executes the following tasks:

- Removes the customized NEP which was created during installation.
- Unconfigures Gemplus OTA-specific NEs using the SACT.

- Undeploys the Gemplus OTA cartridge service model (only if the Gemplus OTA service model is already deployed) using the Service Activation Deployment Tool (SADT).
- Removes the Gemplus OTA-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

- Go to directory where the new SAR file is located.
- Check the daemon and weblogic are running.
- Run script uninstallCartridge - "uninstallCartridge GemplusOTA_2_6.<timestamp>.sar".
- You will be prompted to enter NEP server name.
- You will be prompted to enter weblogic values for host, port, userID and password.
- Script will perform:
  - Remove ASAP NE configuration details.
  - Remove service related information.
  - Remove sample workorders
  - Remove NEP_GOTA

Note:

- The Java properties specified in $ASAP_BASE/programs/<NEP_GOTA>_jinterpreter file,
  - Dvbroker.agent.addr
  - Dvbroker.agent.port
  - Ddefault.user.name
  - Ddefault.user.pwd

  values should match the Communication parameters,

  - HOST_IPADDR
  - PORT
  - HOST_USERID
  - HOST_PASSWORD

  values respectively.
- The Java properties specified in $ASAP_BASE/programs/<NEP_GOTA>_jinterpreter file,
  - Dgxs.product.name
  - Dgxs.invocationmanager.name

  values should match the Communication parameter - PRODUCT value.
- The Java property - "-Dvbroker.orb.gatekeeper.ior" should have the correct URL value, other Java properties should not be altered or removed.

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

dev_type	host	device	param_label	param_value	param_desc
G	GEMP-OTA_2-6-HOST	COMMON_DEVICE_CFG	HOST_IPADDR		The IP address of the host on which the naming service is running on.
G	GEMP-OTA_2-6-HOST	COMMON_DEVICE_CFG	PORT		The port of the naming service is listening on.
G	GEMP-OTA_2-6-HOST	COMMON_DEVICE_CFG	HOST_USERID		User name for login.
G	GEMP-OTA_2-6-HOST	COMMON_DEVICE_CFG	HOST_PASSWORD		Password for login.
G	GEMP-OTA_2-6-HOST	COMMON_DEVICE_CFG	PRODUCT	RCA1	Product name.
G	GEMP-OTA_2-6-HOST	COMMON_DEVICE_CFG	RESPONSELOG	TRUE	Flag to turn ON or OFF the response log.

Modifying gemplus_ota_2_6_ne_config.xml

Use the following procedure to modify gemplus_ota_2_6_ne_config.xml.

To modify gemplus_ota_2_6_ne_config.xml

1. Create a new source directory under /Gemplus_ota_2_6. You can give this directory any appropriate, meaningful name you want to.

   ```
   mkdir <new_source_directory>
   ```

2. Copy GemplusOTA_2_6.sar to this new source directory.

   ```
   cp GemplusOTA_2_6.sar ./<new_source_directory>
   ```

3. Change directory to <new_source_directory>.

   ```
   cd <new_source_directory>
   ```

4. Un-jar GemplusOTA_2_6.sar. This extracts the contents of the sar file.

   ```
   jar xvf GemplusOTA_2_6.sar
   ```

5. Edit <new_source_directory>/Gemplus_ota_2_6/common/application_config/gemplus_ota_2_6_ne_config.xml in with the appropriate changes.
6. Create a new sar file at the <new_source_directory> level.

   ```
   CreateSar $PWD
   ```

7. Uninstall the cartridge using GemplusOTA_2_6.sar in /Gemplus_ota_2_6 (That is, use the original sar file that you copied in Step 2 above—see "Uninstalling the cartridge using scripts" on page 10 for uninstallation instructions).
8. After you uninstall the cartridge, rename the sar file in /Gemplus_ota_2_6 so you have a backup copy of it.
9. Copy the new sar file from <new_source_directory> to /Gemplus_ota_2_6.
10. Reinstall the cartridge (see "Installing the cartridge using scripts" on page 9 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
   ```

2. Ensure loop back mode is on. See "Loopback mode" on page 12 for a description of how to set the loop back parameter to "On".
3. Start ASAP by typing:

   ```
   start_asap_sys
   ```

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 in /Gemplus_ota_2_6/sample_wo 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 l
   ```

All successful work orders returns to the 104 state.

To view the sample work orders provided with this cartridge, refer to the Gemplus OTA cartridge source.

Viewing the sample work orders

You 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. If necessary, create a repository directory under /Gemplus_ota_2_6, copy the sar file to the new directory and un-jar the sar file, as described by Step 1 through Step 4 in "Modifying gemplus_ota_2_6_ne_config.xml" on page 13.
2. Locate and view the sample work order files under /Gemplus_ota_2_6/ota_2_6/subscriber/sample_wo.

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. Create and configure a new customized NEP.
2. Extract the sar file:
   - Create new directory.
   - Copy sar file to that new directory.
   - Run command "jar xvf GemplusOTA_2_6.sar"
3. The file - NEP_GOTA_jinterpreter_template extracted as part of the SAR file contains the Java properties settings.
4. Update the Java properties in the new NEP's jinterpreter file - $ASAP_BASE/programs/<NEP_ID>_jinterpreter similar to the Java properties set in the NEP_GOTA_jinterpreter_template file.
5. 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**.
6. Create a new **Service Activation Project**.
7. Define a new **NE Entity**, based on the **NE Template** contained in the cartridge provided by Oracle.

8. Ensure that the primary pool of the newly created NE is different from the NE template primary pool. You can modify it, if necessary.
9. 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).
10. Create a new **Activation Environment Project** from the **Studio** menu. (Use Studio help for more information).
11. Create **Activation Environment** inside the **Activation Environment Project** and configure the **Connection Details** tab with your Environment ID, Activation version and weblogic data.
12. Connect to your environment using the **Connect** button.
13. 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.
14. First Deploy the **Service Activation** (SA) project as follows:
    - On the **Cartridge** tab, select the necessary SA cartridge and press the **Deploy** button.
    - Select the **NEP Map** tab of the **Activation Environment**. Choose or type the new **NEP** server in 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.
15. 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).
16. Select the **NEP Map** tab of the **Activation Environment**. Choose or type the new **NEP** server in the drop-down box of the **Network Element Processors**. (Use Studio help for more information).
17. Verify the **SADT** console to confirm the installation.
18. Go to ASAP environment.
19. Copy studio_2_6_0.jar file to the $ASAP_BASE/lib directory.
20. Add ${ASAP_BASE}/lib/studio_2_6_0.jar to the CLASSPATH in the JInterpreter file under $ASAP_BASE/programs directory.
21. Deploy Visibroker libraries.

    Visibroker 5.2.1 jar and Gemplus OTA library files are not shipped with the cartridge.

    The following Gempus libraries should be copied from the Gemplus OTA platform into $ASAP_BASE/lib :

    - vbjorb.jar

- vbsec.jar
- lm.jar

The following Gempus libraries should be acquired from the Gemplus and copied into $ASAP_BASE/lib :

- TOPLink.jar
- TOPLinkX.jar
- Tools.jar
- reuse.jar
- basesystem.jar
- common.jar
- xmlwrappers.jar

Add the above jar files in the CLASSPATH in the $ASAP_BASE/programs/<NEP_ID>_jinterpreter file.

22. Add the GemplusOTA_2_6.jar found in $ASAP_BASE/lib in the CLASSPATH in the $ASAP_BASE/programs/<NEP_ID>_jinterpreter file.
23. 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 that is disabled should be unchecked.

3

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	Indicates one of the following parameter types: ◆ 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. For more information on parameter types, refer to the *ASAP Developer Reference*.
Class	Indicates one of the following parameter classifications: ◆ 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_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER
- A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER-SIM-CARD
- A_GEMP-OTA_2-6_ADD_SIM-CARDS
- A_GEMP-OTA_2-6_ADD_SIM-CARDS-RB
- A_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN
- A_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN-RB
- A_GEMP-OTA_2-6_ADD_SIM-CARD-SECUR-BUNDLE
- A_GEMP-OTA_2-6_ADD_SUBSCRIBER
- A_GEMP-OTA_2-6_ADD_SUBSCRIBER-RB
- A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD
- A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD-RB
- A_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER
- A_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER-SIM-CARD
- A_GEMP-OTA_2-6_DEL_SIM-CARDS
- A_GEMP-OTA_2-6_DEL_SIM-CARD-SECUR-BUNDLE-RB
- A_GEMP-OTA_2-6_DEL_SIM-CARD-MSISDN
- A_GEMP-OTA_2-6_DEL_SIM-CARD-SECURITY-DATA
- A_GEMP-OTA_2-6_DEL_SUBSCRIBER
- A_GEMP-OTA_2-6_DEL_SUBSCRIBER-SIM-CARD
- A_GEMP-OTA_2-6_MOD_SIM-CARDS
- A_GEMP-OTA_2-6_MOD_SIM-CARD-SECUR-BUNDLE
- A_GEMP-OTA_2-6_MOD_SUBSCRIBER
- A_GEMP-OTA_2-6_MOD_SUBSCRIBER-RB
- A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD
- A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD-RB
- A_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARDS
- A_GEMP-OTA_2-6_QRY_SUBSCRIBER
- A_GEMP-OTA_2-6_QRY_SUBSCRIBER-RB
- A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD
- A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD-RB
- A_GEMP-OTA_2-6_QRY_SIM-CARD-MSISDN-RB
- A_GEMP-OTA_2-6_QRY_SIM-CARDS-RB
- A_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARD-SECUR-BUNDLE

- A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS
- A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE

A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER

Activates a subscriber on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.activateSubscriber.**

Table 6: A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
MSISDN	Mobile station ISDN.	5 - 15 digits		S	R

MML commands

```
activateSubscriberAccount(String MSISDN)
```

Output parameters

sRETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ACTIVATE_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER-SIM-CARD

Activate SIM Card subscriber on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.activateSubscriberSimCard.**

Table 7: A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
ICCID	The ICCID of the card.			S	O
MSISDN	Mobile station ISDN number of 5 to 20 digits.			S	O
IMSI	International mobile subscriber identity of the card.			S	O

MML commands

```
IF ICCID is  present then call:
activateSIMCard(ICCID).
   ELSE IF MSISDN is present then call:
      activateSIMCard(MSISDN).
   ELSE IF IMSI is present then call:
      activateSIMCard(IMSI).
ELSE call
Throw ProvCartridgeException with message stating none of the optional
parameters are present on the order.
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER-SIM-CARD_UDET=<user defined exit type>

sRETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER-SIM-CARD_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ACTIVATE_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_ADD_SIM-CARDS

Adds a set of SIM cards to the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.addSimCards.**

Table 8: A_GEMP-OTA_2-6_ADD_SIM-CARDS

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILENAME	The name of the SIM card details file on the Network Element's file repository, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters .		S	R

MML commands

```
This command is used to create a set of Sim Cards from an XML file.

This command used the createSIMCards method of the SIMCardLoader API.

Interface Method : createSIMCards(java.lang.String fileName)

Parameter : fileName - The name of the SIM card details file on the Network
Element's file repository.

Returns : an array of reports.
```

Output parameter

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_ADD_SIM-CARDS_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_ADD_SIM-CARDS_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ADD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_ADD_SIM-CARDS-RB

Rollback functionality for the Add SIM Cards service. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.addSimCardsRB.**

Table 9: A_GEMP-OTA_2-6_ADD_SIM-CARDS-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
OLD_FILENAME	The name of the SIM card details file on the Network Element's file repository, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R

MML commands

```
This command is used to create a set of Sim Cards from an XML file for the
rollback.

This command used the createSIMCards method of the SIMCardLoader API.

Interface Method : createSIMCards(java.lang.String fileName)

Parameter : fileName - The name of the SIM card details file on the Network
Element's file repository.

Returns : an array of reports.
```

Output parameter

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_ADD_SIM-CARDS-RB_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_ADD_SIM-CARDS-RB_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ADD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN

Assigns a MSISDN to a subscriber SIM Card on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.addSimCardMsisdn.**

Table 10: A_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
ICCID	The ICCID of the card.			S	O
MSISDN	Mobile Station ISDN Number of 5 to 20 digits of the card.			S	R
IMSI	International Mobile Subscriber Identity of the card.			S	O

MML commands

```
updateSIMCard(SIMCardImage)
```

Output parameter

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ADD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN-RB

Rollback functionality for the Add SIM Card MSISDN service. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.addSimCardMsisdnRB**

Table 11: A_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
OLD_ICCID	The ICCID of the card.			S	O
OLD_MSISDN	Mobile Station ISDN Number of 5 to 20 digits of the card.			S	R
OLD_IMSI	International Mobile Subscriber Identity of the card.			S	O

MML commands

```
This command is used to set the MSISDN value to the already existing SIMCard
in the Rollback cases.
This command used the updateSIMCard method of the SIMCardManager API.
Interface Method : updateSIMCard(SIMCardImage card)
```

```
Parameter : card - The SIMCardImage to be updated
```

Output parameter

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN-RB_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN-RB_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ADD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_ADD_SIM-CARD-SECUR-BUNDLE

Batch loading of new SIM Card settings information. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.addSimCardSecurityBundle.**

Table 12: A_GEMP-OTA_2-6_ADD_SIM-CARD-SECUR-BUNDLE

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILENAME	The name of the SIM card details file on the Network Element's file repository, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters .		S	R

MML commands

```
This command is used to batch load of new card settings information.
```

```
This command used the createSIMCardSecurityBundle method of the
SIMCardSecurityLoader API.

Interface Method : createSIMCardSecurityBundle(java.lang.String fileName)

Parameter : fileName - The name of the SIM card details file on the Network
Element's file repository.

Returns : Report object.
```

Output parameter

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_ADD_SIM-CARD-SECUR-BUNDLE_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_ADD_SIM-CARD-SECUR-BUNDLE_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ADD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_ADD_SUBSCRIBER

Adds a subscriber on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.addSubscriber.**

Table 13: A_GEMP-OTA_2-6_ADD_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
MSISDN	Mobile station ISDN.	5 - 15 digits		S	R
CREATOR_ID	The user account creator ID.			S	O

MML commands

If ICCID parameter is provided then call:

```
createSubscriberAccount(String MSISDN, String ICCID, String CREATOR_ID)
```

If CREATOR_ID is provided then call:

```
createSubscriberAccount(String MSISDN, String CREATOR_ID)
```

Else call:

```
createSubscriberAccount(String MSISDN,PlatformAccessPoint.getAccountID())
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_ADD_SUBSCRIBER_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_ADD_SUBSCRIBER_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ADD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_ADD_SUBSCRIBER-RB

Rollback functionality to delete a subscriber on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.addSubscriberRB.**

Table 14: A_GEMP-OTA_2-6_ADD_SUBSCRIBER-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
MSISDN	Mobile station ISDN number of 5 to 20 digits.			S	R
OLD_CREATOR_ID	The user account creator ID.			S	O

Table 14: A_GEMP-OTA_2-6_ADD_SUBSCRIBER-RB

Parameter Name	Description	Range	Default Value	Type	Class
OLD_PROFILE_NAME	Existing Profile Name.			S	R
OLD_FULL_NAME	Old name of the individual user corresponding to the account.			S	O
OLD_SUBSCRIBER_REFERENCE	Old free text to be used to make an external link with the operators subscriber database.			S	O
OLD_CARD_IDENTIFICATION	Old owned card identifier (to establish a link with the card manager database).			S	O
OLD_AUTHENTICATION_VALUE	Old data required in order to achieve authentication of the account.			S	O
OLD_EXPIRATION_DATE	Old user account expiration date. Valid date format - YYYY/MM/DD HR:MIN:SEC, it possible to provide only the date without the time value			S	O
OLD_AUTHENTICATION_METHOD	Old user account authentication method. Authentification methods can be by password, by mobile challenge, or by certificate. Valid values - AUTH_CERTIFICATE, AUTH_MOBILE_CHALLENGE, AUTH_PASSWORD			S	O

The OLD_AUTHENTICATION_VALUE parameter needs to be sent by the upstream in the workorder for the rollback operation.

MML commands

```
IF ICCID parameter is present then call:
```

```
createSubscriberAccount(String MSISDN, String ICCID, String CREATOR_ID).
   ELSE IF CREATOR_ID is present then call:
      createSubscriberAccount(String MSISDN, String CREATOR_ID).
ELSE call
createSubscriberAccount(String MSISDN, PlatformAccessPoint.getAccountID()).
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_ADD_SUBSCRIBER-RB_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_ADD_SUBSCRIBER-RB_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ADD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD

Add subscriber SIM Card on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.addSubscriberSimCard.**

Table 15: A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
MSISDN	Mobile station ISDN number of 5 to 20 digits.			S	O
ICCID	The ICCID of the card.			S	R
IMSI	International mobile subscriber identity of the card.			S	O

Table 15: A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
CARD_PROFILE	The name of the associated card profile.			S	R
STATE	The card state. Valid values are : STATE_ACTIVE, STATE_INACTIVE, STATE_MARK_DELETED			S	O

MML commands

```
IF all parameters are present then call:
createSIMCard(ICCID, CARD_PROFILE, STATE, MSISDN, IMSI).
   ELSE IF ICCID, CARD_PROFILE, STATE AND MSISDN are present then call:
      createSIMCard(ICCID, CARD_PROFILE, STATE, MSISDN, null).
   ELSE IF ICCID, CARD_PROFILE, STATE AND IMSI are present then call:
      createSIMCard(ICCID, CARD_PROFILE, STATE, null, IMSI).
   ELSE IF ICCID, CARD_PROFILE AND STATE are present then call:
      createSIMCard(ICCID, CARD_PROFILE, STATE, null, null).
   ELSE IF ICCID AND CARD_PROFILE are present then call:
      createSIMCard(ICCID, CARD_PROFILE, STATE, null, null).
ELSE call
Throw ProvCartridgeException with message stating missing mandatory
parameters - ICCID and CARD_PROFILE
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ADD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD-RB

Rollback functionality to delete a subscriber SIM Card on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.addSubscriberSimCardRB.**

Table 16: A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
OLD_MSISDN	Mobile station ISDN number of 5 to 20 digits.			S	O
OLD_ICCID	The ICCID of the card.			S	R
OLD_IMSI	International Mobile Subscriber Identity of the card.			S	O
OLD_CARD_PROFILE	The name of the associated card profile.			S	R
OLD_STATE	The card state. Valid values are : STATE_ACTIVE, STATE_INACTIVE, STATE_MARK_DELETED			S	O
OLD_LINKED_CARD	The linked Card identifier.			S	O

MML commands

```
IF all parameters are present then call:
createSIMCard(ICCID, CARD_PROFILE, STATE, MSISDN, IMSI).
   ELSE IF ICCID, CARD_PROFILE, STATE AND MSISDN are present then call:
      createSIMCard(ICCID, CARD_PROFILE, STATE, MSISDN, null).
   ELSE IF ICCID, CARD_PROFILE, STATE AND IMSI are present then call:
      createSIMCard(ICCID, CARD_PROFILE, STATE, null, IMSI).
   ELSE IF ICCID, CARD_PROFILE AND STATE are present then call:
      createSIMCard(ICCID, CARD_PROFILE, STATE, null, null).
   ELSE IF ICCID AND CARD_PROFILE are present then call:
      createSIMCard(ICCID, CARD_PROFILE, STATE, null, null).
```

```
ELSE call

Throw ProvCartridgeException with message stating missing mandatory
parameters - ICCID and CARD_PROFILE
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD-RB_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD-RB_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ADD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER

Deactivates a subscriber on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.deactivateSubscriber.**

Table 17: A_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
MSISDN	Mobile station ISDN.	5 - 15 digits		S	R

MML commands

deactivateSubscriberAccount(String MSISDN)

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_DEACTIVATE_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER-SIM-CARD

Deactivate subscriber SIM Card on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.deactivateSubscriberSimCard.**

Table 18: A_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
ICCID	The ICCID of the card.			S	O
MSISDN	Mobile station ISDN number of 5 to 20 digits			S	O
IMSI	International mobile subscriber identity of the card.			S	O

MML commands

```
IF ICCID is  present then call:
deactivateSIMCard(ICCID).
   ELSE IF MSISDN is present then call:
      deactivateSIMCard(MSISDN).
   ELSE IF IMSI is present then call:
```

```
      deactivateSIMCard(IMSI).
ELSE call
Throw ProvCartridgeException with message stating none of the optional
parameters are present on the order.
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER-SIM-CARD_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER-SIM-CARD_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_DEACTIVATE_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_DEL_SIM-CARDS

Deletes a set of SIM cards to the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.delSimCards.**

Table 19: A_GEMP-OTA_2-6_DEL_SIM-CARDS

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILENAME	The name of the SIM card details file on the Network Element's file repository, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R

MML commands

```
This command is used to delete a set of Sim Cards from a file.
```

```
This command used the deleteSIMCards method of the SIMCardLoader API.

Interface Method : deleteSIMCards(java.lang.String fileName)

Parameter : fileName - The name of the SIM card details file on the Network
Element's file repository.

Returns : an array of reports.
```

Output parameter

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_DEL_SIM-CARDS_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_DEL_SIM-CARDS_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_DEL_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_DEL_SIM-CARD-SECUR-BUNDLE-RB

Delete a specified Security Data Bundle setting for rollback. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.delSimCardSecurityDataRB.**

Table 20: A_GEMP-OTA_2-6_DEL_SIM-CARD-SECUR-BUNDLE-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R

Table 20: A_GEMP-OTA_2-6_DEL_SIM-CARD-SECUR-BUNDLE-RB

Parameter Name	Description	Range	Default Value	Type	Class
FILEPATH	The absolute path to the directory on the Network Element's file system where the file is located, e.g. /export/home/sunen290.	Length should not exceed 255 characters.		S	R
FILENAME	The name of the SIM card details file on the Network Element's file repository, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R

MML commands

```
This command is used to delete security settings for the specified card.

This command used the deleteSIMCardSecurityBundle method of the
SIMCardSecurityManager API.

Interface Method : deleteSIMCardSecurityBundle(ICCID iccId)

Parameter : iccId - card serial number
```

Output parameter

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_DEL_SIM-CARD-SECUR-BUNDLE-RB_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_DEL_SIM-CARD-SECUR-BUNDLE-RB_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_DEL_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_DEL_SIM-CARD-MSISDN

Unassigns MSISDN to a subscriber SIM Card on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.delSimCardMsisdn**.

Table 21: A_GEMP-OTA_2-6_DEL_SIM-CARD-MSISDN

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
ICCID	The ICCID of the card.			S	O
IMSI	International Mobile Subscriber Identity of the card.			S	O

MML commands

```
updateSIMCard(SIMCardImage)
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_DEL_SIM-CARD-MSISDN_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_DEL_SIM-CARD-MSISDN_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_DEL_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_DEL_SIM-CARD-SECURITY-DATA

Deletes a Security Data Bundle associated with a subscriber SIM Card on the OTA. It is implemented by the Java method

com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.delSimCardSecurityData.

Table 22: A_GEMP-OTA_2-6_DEL_SIM-CARD-SECURITY-DATA

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
ICCID	The ICCID of the card.			S	O
IMSI	International Mobile Subscriber Identity of the card.			S	O

MML commands

```
updateSIMCard(SIMCardImage)
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_DEL_SIM-CARD-SECURITY-DATA_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_DEL_SIM-CARD-SECURITY-DATA_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_DEL_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_DEL_SUBSCRIBER

Deletes a subscriber from the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.delSubscriber.**

Table 23: A_GEMP-OTA_2-6_DEL_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
MSISDN	Mobile station ISDN.	5 - 15 digits		S	R

MML commands

deleteSubscriberAccount(String MSISDN)

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_DEL_SUBSCRIBER_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_DEL_SUBSCRIBER_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_DEL_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_DEL_SUBSCRIBER-SIM-CARD

Delete subscriber SIM Card from the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.delSubscriberSimCard.**

Table 24: A_GEMP-OTA_2-6_DEL_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
ICCID	The ICCID of the card.			S	O
MSISDN	Mobile station ISDN number of 5 to 20 digits.			S	O
IMSI	International Mobile Subscriber Identity of the card.			S	O

MML commands

```
IF ICCID is  present then call:
deleteSIMCard(ICCID).
   ELSE IF MSISDN is present then call:
      deleteSIMCard(MSISDN).
   ELSE IF IMSI is present then call:
      deleteSIMCard(IMSI).
ELSE call
Throw ProvCartridgeException with message stating none of the optional
parameters are present on the order.
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_DEL_SUBSCRIBER-SIM-CARD_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_DEL_SUBSCRIBER-SIM-CARD_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_DEL_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_MOD_SIM-CARDS

Updates a set of SIM cards to the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.modSimCards.**

Table 25: A_GEMP-OTA_2-6_DEL_SIM-CARDS

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILENAME	The name of the SIM card details file on the Network Element's file repository, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters .		S	R

MML commands

```
This command is used to update a set of Sim Cards from a file.

This command used the updateSIMCards method of the SIMCardLoader API.

Interface Method : updateSIMCards(java.lang.String fileName)

Parameter : fileName - The name of the SIM card details file on the Network
Element's file repository.

Returns : an array of reports.
```

Output parameter

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_MOD_SIM-CARDS_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_MOD_SIM-CARDS_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_MOD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_MOD_SIM-CARD-SECUR-BUNDLE

Batch loading to update the existing SIM Card settings information. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.modSimCardSecurityBundle.**

Table 26: A_GEMP-OTA_2-6_MOD_SIM-CARD-SECUR-BUNDLE

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILENAME	The name of the SIM card details file on the Network Element's file repository, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R

MML commands

```
This command is used to batch load for update of existing card settings
information.

This command used the updateSIMCardSecurityBundle method of the
SIMCardSecurityLoader API.

Interface Method : updateSIMCardSecurityBundle(java.lang.String fileName)

Parameter : fileName - The name of the SIM card details file on the Network
Element's file repository.

Returns : Report object.s
```

Output parameter

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_MOD_SIM-CARD-SECUR-BUNDLE_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_MOD_SIM-CARD-SECUR-BUNDLE_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_MOD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_MOD_SUBSCRIBER

Modifies a subscriber on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.modSubscriber.**

Table 27: A_GEMP-OTA_2-6_MOD_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
MSISDN	Mobile station ISDN.	5 - 15 digits		S	R
PROFILE_NAME	Existing Profile Name.			S	R
FULL_NAME	Name of the individual user corresponding to the account.			S	O
SUBSCRIBER_REFERENCE	Free text to be used to make an external link with the operator's subscriber database.			S	O
CARD_IDENTIFICATION	Owned card identifier (to establish a link with the card manager database).			S	O

Table 27: A_GEMP-OTA_2-6_MOD_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
AUTHENTICATION_VALUE	Data required in order to achieve authentication of the account.			S	O
EXPIRATION_DATE	User account expiration date.	Valid date format - YYYY/MM/DD HR:MIN:SEC, it possible to provide only the date without the time value		S	O
AUTHENTICATION_METHOD	User account authentication method. Authentication methods can be by password, by mobile challenge, or by certificate.	Valid values - AUTH_CERTIFICATE, AUTH_MOBILE_CHALLENGE, AUTH_PASSWORD		S	O
NEW_ID	New subscriber ID.			S	O

MML commands

updateSubscriberAccount(SubscriberAccount account)

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_MOD_SUBSCRIBER_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_MOD_SUBSCRIBER_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_MOD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_MOD_SUBSCRIBER-RB

Rolls back the modification made to subscriber on the OTA. It is implemented by the Java method

com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.modSubscriberRB.

Table 28: A_GEMP-OTA_2-6_MOD_SUBSCRIBER-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
MSISDN	Mobile station ISDN.	5 - 15 digits		S	R
PROFILE_NAME	Existing Profile Name.			S	R
OLD_FULL_NAME	Old name of the individual user corresponding to the account.			S	O
OLD_SUBSCRIBER_REFERENCE	Old free text to be used to make an external link with the operator's subscriber database.			S	O
OLD_CARD_IDENTIFICATION	Old owned card identifier (to establish a link with the card manager database).			S	O

Table 28: A_GEMP-OTA_2-6_MOD_SUBSCRIBER-RB

Parameter Name	Description	Range	Default Value	Type	Class
OLD_AUTHENTICATION_VALUE	Old data required in order to achieve authentication of the account.			S	O
OLD_EXPIRATION_DATE	Old user account expiration date.	Valid date format - YYYY/MM/DD HR:MIN:SEC, it possible to provide only the date without the time value		S	O
OLD_AUTHENTICATION_METHOD	Old user account authentication method. Authentication methods can be by password, by mobile challenge, or by certificate.	Valid values - AUTH_CERTIFICATE, AUTH_MOBILE_CHALLENGE, AUTH_PASSWORD		S	O
NEW_ID	Old subscriber ID.			S	O

The OLD_AUTHENTICATION_VALUE parameter needs to be sent by the upstream in the workorder for the rollback operation.

MML commands

updateSubscriberAccount(SubscriberAccount account)

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_MOD_SUBSCRIBER-RB_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_MOD_SUBSCRIBER-RB_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_DEL_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD

Modify subscriber SIM Card on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.modSubscriberSimCard.**

Table 29: A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
ICCID	The ICCID of the card.			S	O
CARD_PROFILE	The name of the associated card profile.			S	O
MSISDN	Mobile Station ISDN Number of 5 to 20 digits of the card.			S	O
IMSI	International mobile subscriber identity of the card.			S	O
LINKED_CARD	The linked card identifier.			S	O

MML commands

```
updateSIMCard(SIMCardImage).
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_MOD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD-RB

Rollback Modify subscriber SIM Card on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.modSubscriberSimCardRB.**

Table 30: A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
OLD_ICCID	The ICCID of the card.			S	O
OLD_CARD_PROFILE	The name of the associated card profile.			S	O
OLD_MSISDN	Mobile Station ISDN Number of 5 to 20 digits of the card.			S	O
OLD_IMSI	International Mobile Subscriber Identity of the card.			S	O

Table 30: A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD-RB

Parameter Name	Description	Range	Default Value	Type	Class
OLD_LINKED_CARD	The linked Card identifier.			S	O

MML commands

```
updateSIMCard(SIMCardImage).
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD-RB_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD-RB_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_MOD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARDS

Returns the list of available files on the server. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.qryAllFilesSimCards.**

Table 31: A_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARDS

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

MML commands

```
This command is used to return the list of available files on the server.

This command used the getFileNames method of the SIMCardLoader API.

Interface Method : getFileNames()

Returns : Array of Strings(filenames).
```

Output parameter

RETURN AS CSDL PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARDS_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARDS_RETURN_INFO = <NE error description>

<RETURN_DATA_PREFIX>FILENAME[1]

<RETURN_DATA_PREFIX>FILENAME[2]

<RETURN_DATA_PREFIX>FILENAME[3]

<RETURN_DATA_PREFIX>FILENAME[n]

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

<RETURN_DATA_PREFIX>EROTA_QRY_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

<RETURN_DATA_PREFIX>ERRORINFO = <NE Error Message>

<RETURN_DATA_PREFIX>ERRORCODE = <user defined exit type>

<RETURN_DATA_PREFIX>ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARD-SECUR-BUNDLE

Returns the list of available files on the server. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.qryAllFilesSimCardSecurityBundle.**

Table 32: A_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARD-SECUR-BUNDLE

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

MML commands

```
This command is used to return the list of available files on the server.

This command used the getFileNames method of the SIMCardSecurityLoader API.

Interface Method : getFileNames()

Returns : Array of Strings(filenames).
```

Output parameter

RETURN AS CSDL PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARD-SECUR-BUNDLE_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARD-SECUR-BUNDLE_RETURN_INFO = <NE error description>

<RETURN_DATA_PREFIX>FILENAME[1]

<RETURN_DATA_PREFIX>FILENAME[2]

<RETURN_DATA_PREFIX>FILENAME[3]

<RETURN_DATA_PREFIX>FILENAME[n]

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

<RETURN_DATA_PREFIX>EROTA_QRY_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

<RETURN_DATA_PREFIX>ERRORINFO = <NE Error Message>

<RETURN_DATA_PREFIX>ERRORCODE = <user defined exit type>

<RETURN_DATA_PREFIX>ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_QRY_SUBSCRIBER

Queries a subscriber on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.qrySubscriber.**

Table 33: A_GEMP-OTA_2-6_QRY_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
MSISDN	Mobile station ISDN.	5 - 15 digits		S	O
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

MML commands

getSubscriberAccount(String MSISDN)

Output parameters

RETURN AS CSDL PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY_SUBSCRIBER_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY_SUBSCRIBER_RETURN_INFO = <NE error description>

RETURN AS CSDL AND INFO PARAMETER(s):

<RETURN_DATA_PREFIX>ID=<ID value>

<RETURN_DATA_PREFIX>FULL_NAME=<FULL_NAME value>

<RETURN_DATA_PREFIX>CREATION_DATE=<CREATION_DATE value>

<RETURN_DATA_PREFIX>EXPIRATION_DATE=<EXPIRATION_DATE value>

<RETURN_DATA_PREFIX>AUTHENTICATION_METHOD=<AUTHENTICATION_METHOD value>

<RETURN_DATA_PREFIX>PROFILE_NAME=<PROFILE_NAME value>

<RETURN_DATA_PREFIX>CREATOR_ID=<CREATOR_ID value>

<RETURN_DATA_PREFIX>ENABLE=<ENABLE value>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

<RETURN_DATA_PREFIX>EROTA_QRY_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

<RETURN_DATA_PREFIX>ERRORINFO = <NE Error Message>

<RETURN_DATA_PREFIX>ERRORCODE = <user defined exit type>

<RETURN_DATA_PREFIX>ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_QRY_SUBSCRIBER-RB

Rollback functionality for the Query subscriber on the OTA service. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.qrySubscriberRB.**

Table 34: A_GEMP-OTA_2-6_QRY_SUBSCRIBER-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
MSISDN	Mobile station ISDN number of 5 to 20 digits.			S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

MML commands

getSubscriberAccount(String MSISDN)

Output parameters

RETURN AS CSDL PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY_SUBSCRIBER-RB_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY_SUBSCRIBER-RB_RETURN_INFO = <NE error description>

RETURN AS CSDL AND INFO PARAMETER(s):

<RETURN_DATA_PREFIX>OLD_ID=<ID value>

<RETURN_DATA_PREFIX>OLD_FULL_NAME=<FULL_NAME value>

<RETURN_DATA_PREFIX>OLD_CREATION_DATE=<CREATION_DATE value>

<RETURN_DATA_PREFIX>OLD_EXPIRATION_DATE=<EXPIRATION_DATE value>

<RETURN_DATA_PREFIX>OLD_AUTHENTICATION_METHOD=<AUTHENTICATION_METHOD value>

<RETURN_DATA_PREFIX>OLD_PROFILE_NAME=<PROFILE_NAME value>

<RETURN_DATA_PREFIX>OLD_CREATOR_ID=<CREATOR_ID value>

<RETURN_DATA_PREFIX>OLD_ENABLE=<ENABLE value>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

<RETURN_DATA_PREFIX>EROTA_QRY_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

<RETURN_DATA_PREFIX>ERRORINFO = <NE Error Message>

<RETURN_DATA_PREFIX>ERRORCODE = <user defined exit type>

<RETURN_DATA_PREFIX>ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD

Query subscriber SIM Card on the OTA. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.qrySubscriberSimCard.**

Table 35: A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R

Table 35: A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
ICCID	The ICCID of the card.			S	O
MSISDN	Mobile station ISDN number of 5 to 20 digits			S	O
IMSI	International Mobile Subscriber Identity of the card.			S	O
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

MML commands

```
To get SIM Card Status:
IF ICCID is  present then call:
existSIMCard(ICCID).
   ELSE IF MSISDN is present then call:
      existSIMCard(MSISDN).
   ELSE IF IMSI is present then call:
      existSIMCard(IMSI).
ELSE call
Throw ProvCartridgeException with message stating none of the optional
parameters are present on the order.


To get SIM Card details:
IF IMSI is present then call
   SIMCardViewer::getSIMCard(IMSI imsi);
ELSE IF MSISDN is present then call
   SIMCardViewer::getSIMCard(MSISDN msisdn);
```

Output parameters

RETURN AS CSDL PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD_RETURN_INFO = <NE error description>

RETURN AS CSDL AND INFO PARAMETER(s):

<RETURN_DATA_PREFIX>CARD_STATUS=<Card Status value>

<RETURN_DATA_PREFIX>IMSI=<IMSI value>

<RETURN_DATA_PREFIX>MSISDN=<MSISDN value>

<RETURN_DATA_PREFIX>ICCID=<ICCID value>

<RETURN_DATA_PREFIX>STATE=<STATE value>

<RETURN_DATA_PREFIX>CARD_PROFILE=<CARD_PROFILE value>

<RETURN_DATA_PREFIX>LINKED_CARD=<LINKED_CARD value>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

<RETURN_DATA_PREFIX>EROTA_QRY_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

<RETURN_DATA_PREFIX>ERRORINFO = <NE Error Message>

<RETURN_DATA_PREFIX>ERRORCODE = <user defined exit type>

<RETURN_DATA_PREFIX>ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD-RB

Rollback functionality for the Query subscriber SIM Card on the OTA service. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.qrySubscriberSimCardRB.**

Table 36: A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
ICCID	The ICCID of the card.			S	O
MSISDN	Mobile station ISDN number of 5 to 20 digits			S	O
IMSI	International Mobile Subscriber Identity of the card.			S	O
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

MML commands

```
To get SIM Card Status:
IF ICCID is  present then call:
existSIMCard(ICCID).
   ELSE IF MSISDN is present then call:
      existSIMCard(MSISDN).
   ELSE IF IMSI is present then call:
      existSIMCard(IMSI).
ELSE call
Throw ProvCartridgeException with message stating none of the optional
parameters are present on the order.
To get SIM Card details:
IF IMSI is present then call
   SIMCardViewer::getSIMCard(IMSI imsi);
ELSE IF MSISDN is present then call
   SIMCardViewer::getSIMCard(MSISDN msisdn)
```

Output parameters

RETURN AS CSDL PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD-RB_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD-RB_RETURN_INFO = <NE error description>

RETURN AS CSDL AND INFO PARAMETER(s):

<RETURN_DATA_PREFIX>CARD_STATUS=<Card Status value>

<RETURN_DATA_PREFIX>OLD_IMSI=<IMSI value>

<RETURN_DATA_PREFIX>OLD_MSISDN=<MSISDN value>

<RETURN_DATA_PREFIX>OLD_ICCID=<ICCID value>

<RETURN_DATA_PREFIX>OLD_STATE=<STATE value>

<RETURN_DATA_PREFIX>OLD_CARD_PROFILE=<CARD_PROFILE value>

<RETURN_DATA_PREFIX>OLD_LINKED_CARD=<LINKED_CARD value>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

<RETURN_DATA_PREFIX>EROTA_QRY_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

<RETURN_DATA_PREFIX>ERRORINFO = <NE Error Message>

<RETURN_DATA_PREFIX>ERRORCODE = <user defined exit type>

<RETURN_DATA_PREFIX>ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_QRY_SIM-CARD-MSISDN-RB

Rollback functionality for the Query subscriber SIM Card MSISDN on the OTA service. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.qrySimCardMSISDNRB**

Table 37: A_GEMP-OTA_2-6_QRY_SIM-CARD-MSISDN-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
ICCID	The ICCID of the card.			S	O
IMSI	International Mobile Subscriber Identity of the card.			S	O
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

MML commands

```
This command is used to get values of the already existing SIMCard to be used
by in the Rollback cases.
This command used the existSIMCard method of the SIMCardManager API and
getSIMCard method of the SIMCardViewer API.
Interface Method : existSIMCard(MSISDN msisdn) or existSIMCard(ICCID iccid)
or existSIMCard(IMSI imsi)
Parameter : iccid - ICCID of the card or imsi - IMSI of the card.
Interface Method : getSIMCard(ICCID iccid) or getSIMCard(IMSI imsi)
Parameter : iccid - ICCID of the card or imsi - IMSI of the card.
Returns : SIMCardImage object.
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_QRY_SIM-CARD-MSISDN-RB_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_QRY_SIM-CARD-MSISDN-RB_RETURN_INFO = <NE error description>

RETURN AS CSDL AND INFO PARAMETER(s):

OLD_IMSI=<IMSI value>

OLD_MSISDN=<MSISDN value>

OLD_ICCID=<ICCID value>

A_GEMP-OTA_2-6_QRY_SIM-CARDS-RB

Rollback functionality for the query SIM cards service. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.qrySimCardsRB.**

Table 38: A_GEMP-OTA_2-6_QRY_SIM-CARDS-RB

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILEPATH	The absolute path to the directory on the Network Element's file system where the file is located, e.g. /export/home/sunen290.	Length should not exceed 255 characters.		S	R
FILENAME	The name of the SIM card details file on the Network Element's file repository, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

MML commands

```
This command is used to get values of the already existing SIMCard.

This command used the existSIMCard method of the SIMCardManager API and
getSIMCard method of the SIMCardViewer API.

Interface Method : existSIMCard(ICCID iccid) or existSIMCard(IMSI imsi)

Parameter : iccid - ICCID of the card or imsi - IMSI of the card.

Interface Method : getSIMCard(ICCID iccid) or getSIMCard(IMSI imsi)

Parameter : iccid - ICCID of the card or imsi - IMSI of the card.

Returns : SIMCardImage object.
```

Output parameter

RETURN AS CSDL PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY_SIM-CARDS-RB_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): <RETURN_DATA_PREFIX>A_GEMP-OTA_2-6_QRY_SIM-CARDS-RB_RETURN_INFO = <NE error description>

RETURN AS CSDL AND INFO PARAMETER(s):

<RETURN_DATA_PREFIX>CARD_STATUS=<Card Status value>

<RETURN_DATA_PREFIX>OLD_IMSI=<IMSI value>

<RETURN_DATA_PREFIX>OLD_MSISDN=<MSISDN value>

<RETURN_DATA_PREFIX>OLD_ICCID=<ICCID value>

<RETURN_DATA_PREFIX>OLD_STATE=<STATE value>

<RETURN_DATA_PREFIX>OLD_CARD_PROFILE=<CARD_PROFILE value>

<RETURN_DATA_PREFIX>OLD_LINKED_CARD=<LINKED_CARD value>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

<RETURN_DATA_PREFIX>EROTA_QRY_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

<RETURN_DATA_PREFIX>ERRORINFO = <NE Error Message>

<RETURN_DATA_PREFIX>ERRORCODE = <user defined exit type>

<RETURN_DATA_PREFIX>ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS

Upload file containing a set of Sim Card settings information. It is implemented by the Java method **com.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.uploadFileSimCards**

Table 39: A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R

Table 39: A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS

Parameter Name	Description	Range	Default Value	Type	Class
FILEPATH	The absolute path to the directory on the ASAP host file system where the file is located, e.g. /export/home/sunen290	Length should not exceed 255 characters.		S	R
FILENAME	The name of the SIM card details file on the ASAP host's file system, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R

MML commands

```
This command is used to upload a file specified by its path from the client
side to the server.

This command used the upload method of the SIMCardLoader API.

Interface Method : upload(java.lang.String src, java.lang.String fileName)

Parameter :
src - The absolute path to the directory on the ASAP host file system where
the file is located.
fileName - The name of the SIM card details file on the ASAP host's file
system.
```

Output parameters

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ADD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE

Uploading files containing information on a set of SIM Card settings. It is implemented by the Java method **ccom.mslv.activation.cartridge.gemp.ota.x2_6.prov.OTAProvisioning.uploadFileSimCardSecurityBundle.**

Table 40: A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILEPATH	The absolute path to the directory on the ASAP host file system where the file is located, e.g. /export/home/sunen290	Length should not exceed 255 characters.		S	R
FILENAME	The name of the SIM card details file on the ASAP host's file system, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R

MML commands

```
This command is used to upload a file specified by its path from the client
side to the server.

This command used the upload method of the SIMCardSecurityLoader API.

Interface Method : upload(java.lang.String src, java.lang.String fileName)

Parameter :
src - The absolute path to the directory on the ASAP host file system where
the file is located.
fileName - The name of the SIM card details file on the ASAP host's file
system.
```

Output parameter

RETURN AS CSDL PARAMETER(s): A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE_UDET=<user defined exit type>

RETURN AS INFO PARAMETER(s): A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE_RETURN_INFO = <NE error description>

In Error case the following parameters will be returned along with the above parameters:

RETURN AS CSDL PARAMETER(s):

EROTA_ADD_ERR_CODE = <user defined exit type>

RETURN AS INFO PARAMETER(s):

ERRORINFO = <NE Error Message>

ERRORCODE = <user defined exit type>

ERRORDESCRIPTION = <NE Error Description>

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>
</userDefinedExitType>
<userDefinedExitType>
    <searchPattern>251-275&&[^261-265]</searchPattern>[6]
    <userType>U_DELAYED_FAIL</userType>
    <baseType>DELAYED_FAIL</baseType>
    <description>The ASDL has failed during provisioning.</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.
6. 251 to 275 but not 261 to 265 will translate to a base type of DELAYED_FAILURE.

```
</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.
- \b\d{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 41: User defined ASDL exit types

Search pattern	User_type	Base_type	Description
((?s).)*ProvCartridgeException((?s).)*	GEMOTA_PROVCARTEXCEP	FAIL	The NE command was denied due to Provisioning cartridge Exception.
((?s).)*IOException((?s).)*	GEMOTA_IOEXCEPTION	RETRY_DIS	The NE command was denied due to IO exception.
((?s).)*ORBException((?s).)*	GEMOTA_ORBEXCEP	RETRY_DIS	The NE command was denied due to ORB exception.

Table 41: User defined ASDL exit types

Search pattern	User_type	Base_type	Description
((?s).)*GenericException((?s).)*	GEMOTA_GRAL_EXCEP	FAIL	The NE call was denied due to general exception.
((?s).)*DeniedAccessException((?s).)*	GEMOTA_DENIED_ACCESS	FAIL	Connection to platform lost. Access denied.
((?s).)*ExpiredAccountException((?s).)*	GEMOTA_EXP_ACCOUNT	FAIL	Connection to platform lost. Expired account.
((?s).)*TimeOutException((?s).)*	GEMOTA_TIMEOUT	SOFT_FAIL	Connection to platform lost. Timeout.
((?s).)*AlreadyExistsException((?s).)*	GEMOTA_ALRDY_EXISTS	FAIL	Subscriber account already exists.
((?s).)*NoMatchingObjectException((?s).)*	GEMOTA_NO_MATCH	FAIL	Account has not been previously declared.
((?s).)*IllegalArgumentException((?s).)*	GEMOTA_ILLEGAL_ARG	FAIL	Argument does not exist.
((?s).)*InvalidStateException((?s).)*	GEMOTA_INVALID_STATE	FAIL	Card manager is not active.
((?s).)*Missing mandatory parameter - MSISDN((?s).)*	GEMOTA_MISSPARAM_EX	FAIL	Provisioning cartridge Exception due to missing mandatory MSISDN parameter.
SIM Card exists in the CardManager	GEMOTA_SIMCARD_EX	SUCCEED	SIM Card exists in the CardManager.
SIM Card does not exist in the CardManager	GEMOTA_SIMCARD_NOEX	SOFT_FAIL	SIM Card does not exist in the CardManager.
SUBSCRIBER_CREATED	GEMOTA_SUB_CREATED	SUCCEED	Successfully created Subscriber Account.
SUBSCRIBER_DELETED	GEMOTA_SUB_DELETED	SUCCEED	Successfully deleted Subscriber Account.
SUBSCRIBER_ACTIVATED	GEMOTA_SUB_ACTIVATED	SUCCEED	Successfully activated Subscriber Account.
SUBSCRIBER_DEACTIVATED	GEMOTA_SUB_DEACTIVE	SUCCEED	Successfully de-activated Subscriber Account.

Table 41: User defined ASDL exit types

Search pattern	User_type	Base_type	Description
SUBSCRIBER_UPDATED	GEMOTA_SUB_UPDATED	SUCCEED	Successfully updated Subscriber Account.
SUBSCRIBER_QUERIED	GEMOTA_SUB_EXISTS	SUCCEED	Successfully got details of Subscriber Account.
SUBSCRIBER_NOT_FOUND	GEMOTA_SUB_NO_EX	SOFT_FAIL	Subscriber Account does not exists.
CARD_CREATED	GEMOTA_SIM_CREATED	SUCCEED	Successfully created SIM Card.
CARD_DELETED	GEMOTA_SIM_DELETED	SUCCEED	Successfully deleted SIM Card.
CARD_ACTIVATED	GEMOTA_SIM_ACTIVATED	SUCCEED	Successfully activated SIM Card.
CARD_DEACTIVATED	GEMOTA_SIM_DEACTIVE	SUCCEED	Successfully de-activated SIM Card.
CARD_UPDATED	GEMOTA_SIM_UPDATED	SUCCEED	Successfully updated SIM Card.
SECURITY_DELETED	GEMOTA_SECDATE_DEL	SUCCEED	Successfully deleted the SIM Card security bundle.
ALL_FILES_QUERIED	GEMOTA_FILE_EXISTS	SUCCEED	Successfully retrieved the files on the server.
FILE_NOT_FOUND	GEMOTA_NO_FILE_EXIST	FAIL	File does not exist on the sever.
SIM_CARDS_CREATED	GEMOTA_SIMCARDS_ADD	SUCCEED	A set of sim cards created successfully.
SIM_CARDS_DELETED	GEMOTA_SIMCARDS_DEL	SUCCEED	A set of sim cards deleted successfully.
FILE_UPLOADED	GEMOTA_FILE_UPLOADED	SUCCEED	Successfully uploaded the files on the server.
SIM_CARDS_UPDATED	GEMOTA_SIMCARDS_MOD	SUCCEED	A set of sim cards updated successfully.
SIM_CARDS_SECUR_CREATED	GEMOTA_SIMSECU_ADD	SUCCEED	Successfully batch loaded new Sim Card settings information.

Table 41: User defined ASDL exit types

Search pattern	User_type	Base_type	Description
SIM_CARDS_SECUR_UPDATED	GEMOTA_SIMSECU_MOD	SUCCEED	Successfully batch loaded update of existing Sim Card settings information.
SIM_CARDS_SECUR_DELETED	GEMOTA_SIMSECU_DEL	SUCCEED	Successfully deleted a specified Sim Card settings information.
((?s).)*CardNotFoundException((?s).)*	GEMOTA_CARD_NO_EXIST	FAIL	CardNotFoundException.
((?s).)*ServiceAccessDeniedException((?s).)*	GEMOTA_ACCESS_DENIED	FAIL	Service Access denied.
((?s).)*ObjectAlreadyDefinedException((?s).)*	GEMOTA_OBJ_ALRDY_EXS	FAIL	Object may be already defined in the repository.
((?s).)*FileNotFoundException((?s).)*	GEMOTA_FILE_NOT_FOUN	FAIL	No such file or directory.
((?s).)*Unable to add Smart Cards((?s).)*	GEMOTA_UNABLE_AD_SIM	FAIL	Unable to add Smart Cards
((?s).)*ObjectNotFoundException((?s).)*	GEMOTA_OBJ_NOT_FOUND	FAIL	Card Profile does not exist in repository.
((?s).)*BadDataConsistencyException((?s).)*	GEMOTA_BAD_DATA_CONS	FAIL	BadDataConsistencyException
((?s).)*BadParameterFormatException((?s).)*	GEMOTA_BAD_PARAM	FAIL	BadParameterFormatException
((?s).)*BadRequestFormatException((?s).)*	GEMOTA_BAD_REQFORMAT	FAIL	BadRequestFormatException.
((?s).)*CardAssociatedToProfileException((?s).)*	GEMOTA_CARD_ASSO_PRO	FAIL	CardAssociatedToProfileException
((?s).)*ChannelNotFoundException((?s).)*	GEMOTA_NO_CHANNEL	FAIL	ChannelNotFoundException
((?s).)*ComponentNotFoundException((?s).)*	GEMOTA_COMP_NOT_FOUN	FAIL	ComponentNotFoundException.
((?s).)*ComProblemException((?s).)*	GEMOTA_COMPROB_EXCEP	FAIL	ComProblemException.

Table 41: User defined ASDL exit types

Search pattern	User_type	Base_type	Description
((?s).)*FormattingLibraryException((?s).)*	GEMOTA_LIBRARY_EXCEP	FAIL	FormattingLibraryException.
((?s).)*GatewayDownException((?s).)*	GEMOTA_GATEWAY_DOWN	FAIL	GatewayDownException.
((?s).)*GeneralSystemException((?s).)*	GEMOTA_GEN_SYS_EXCEP	FAIL	GeneralSystemException.
((?s).)*InactiveCardException((?s).)*	GEMOTA_INACTIVE_CARD	FAIL	InactiveCardException.
((?s).)*InactiveComponentException((?s).)*	GEMOTA_INACTIVE_COMP	FAIL	InactiveComponentException.
((?s).)*InactiveProfileException((?s).)*	GEMOTA_INACTIVE_PROF	FAIL	InactiveProfileException.
((?s).)*InactiveServiceException((?s).)*	GEMOTA_INACT_SERVICE	FAIL	InactiveServiceException.
((?s).)*IncompatibleDataTypeException((?s).)*	GEMOTA_INCOM_DATATYP	FAIL	IncompatibleDataTypeException.
((?s).)*MissingInformationException((?s).)*	GEMOTA_MISSING_INFO	FAIL	MissingInformationException
((?s).)*MissingParameterException((?s).)*	GEMOTA_MISSNG_PARAM	FAIL	MissingParameterException.
((?s).)*NoChannelForProtocolException((?s).)*	GEMOTA_NO_CHAN_PROTO	FAIL	NoChannelForProtocolException.
((?s).)*NoElementInListException((?s).)*	GEMOTA_NO_ELEME_LIST	FAIL	NoElementInListException.
((?s).)*NoFilterFoundException((?s).)*	GEMOTA_NO_FILTR_FOUN	FAIL	NoFilterFoundException.
((?s).)*ProductNotConnectedException((?s).)*	GEMOTA_PROD_NOT_CONN	FAIL	ProductNotConnectedException.
((?s).)*RequestDeletedByUserException((?s).)*	GEMOTA_REQDEL_BYUSER	FAIL	RequestDeletedByUserException.

Table 41: User defined ASDL exit types

Search pattern	User_type	Base_type	Description
((?s).)*RequestExecutionFailedException((?s).)*	GEMOTA_REQ_EXEC_FAIL	FAIL	RequestExecutionFailedException.
((?s).)*RequestExecutionPartialSucessException((?s).)*	GEMOTA_REQ_PART_SUCC	FAIL	RequestExecutionPartialSucessException.
((?s).)*RequestExpiredException((?s).)*	GEMOTA_REQ_EXPIRED	FAIL	RequestExpiredException.
((?s).)*RequestNotFoundException((?s).)*	GEMOTA_REQ_NOT_FOUND	FAIL	RequestNotFoundException.
((?s).)*SendingButConnectionLostException((?s).)*	GEMOTA_SEND_CONN_LOS	FAIL	SendingButConnectionLostException.
((?s).)*SendingRouterException((?s).)*	GEMOTA_SEND_ROUTR_EX	FAIL	SendingRouterException.
((?s).)*ServiceDeclarationException((?s).)*	GEMOTA_SERV_DECLR_EX	FAIL	ServiceDeclarationException.
((?s).)*ServiceDefinitionException((?s).)*	GEMOTA_SERVIC_DEF_EX	FAIL	ServiceDefinitionException.
((?s).)*ServiceExecutionException((?s).)*	GEMOTA_SERVC_EXECUTE	FAIL	ServiceExecutionException.
((?s).)*ServiceNotFoundException((?s).)*	GEMOTA_SERVC_NOTFOUN	FAIL	ServiceNotFoundException.
((?s).)*ServiceStateException((?s).)*	GEMOTA_SERV_STATE_EX	FAIL	ServiceStateException.
((?s).)*ShutdownException((?s).)*	GEMOTA_SHUTDOWN_EX	FAIL	ShutdownException.
((?s).)*UnhandledException((?s).)*	GEMOTA_UNHANDLED_EX	FAIL	UnhandledException.
((?s).)*UnknownProtocolException((?s).)*	GEMOTA_UNKNOW_PROTCL	FAIL	UnknownProtocolException.

Table 41: User defined ASDL exit types

Search pattern	User_type	Base_type	Description
((?s).)*UnknownServiceDeclException((?s).)*	GEMOTA_UNKN_SER_DECL	FAIL	UnknownServiceDeclException.
((?s).)*UnknownServiceImplException((?s).)*	GEMOTA_UNKN_SER_IMPL	FAIL	UnknownServiceImplException.
((?s).)*UserAccessException((?s).)*	GEMOTA_USER_ACCES_EX	FAIL	UserAccessException.
((?s).)*FormattingException((?s).)*	GEMOTA_FORMAT_EXCEP	FAIL	FormattingException.
((?s).)*InactiveFormattingTypeException((?s).)*	GEMOTA_INACTV_FORMAT	FAIL	InactiveFormattingTypeException
((?s).)*SecurityDataException((?s).)*	GEMOTA_SECU_DATA_EX	FAIL	SecurityDataException.
((?s).)*TransportKeyException((?s).)*	GEMOTA_TRANSPORT_KEY	FAIL	TransportKeyException.
((?s).)*UnhandledFormattingTypeException((?s).)*	GEMOTA_UNHANDL_FORMA	FAIL	UnhandledFormattingTypeException.
SIM_CARDS_EXCEPTION	GEMOTA_SIMCARD_EXCEP	FAIL	Exceptions occurred for SIM Cards.
((?s).)*ParsingException((?s).)*	GEMOTA_PARSE_EXCEP	FAIL	ParsingException.
GEMOTA_NO_ICCID_FOUN	GEMOTA_NO_ICCID_FOUN	FAIL	The associated ICCID for the simcard is not found.
GEMOTA_NO_FILE_FOUND	GEMOTA_NO_FILE_FOUND	FAIL	No such file or directory.
((?s).)*DatabaseException((?s).)*	GEMOTA_DATABASEEXCEP	FAIL	DatabaseException.
SIM_CARDS_SECU_BUNDLE_QUERIED	GEMOTA_SECU_DOM_QRY	SUCCEED	Successfully queried the SIM card security bundle details for ICCID.

Table 41: User defined ASDL exit types

Search pattern	User_type	Base_type	Description
((?s).)*BadXMLDefinitionException((?s).)*	GEMOTA_BAD_XML_DEFN	FAIL	BadXMLDefinitionException.

UserExitType.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceModel xmlns="http://www.metasolv.com/ServiceActivation/2003/
ServiceModel">
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*ProvCartridgeException((?s).)*</
searchPattern>
          <userType>GEMOTA_PROVCARTEXCEP</userType>
          <baseType>FAIL</baseType>
         <description>The NE command was denied due to Provisioning cartridge
Exception.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*IOException((?s).)*</searchPattern>
          <userType>GEMOTA_IOEXCEPTION</userType>
          <baseType>RETRY_DIS</baseType>
          <description>The NE command was denied due to IO exception.</
description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*ORBException((?s).)*</searchPattern>
          <userType>GEMOTA_ORBEXCEP</userType>
          <baseType>RETRY_DIS</baseType>
          <description>The NE command was denied due to ORB exception.</
description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
```

```
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
           </neDescriptor>
           <searchPattern>((?s).)*GenericException((?s).)*</searchPattern>
           <userType>GEMOTA_GRAL_EXCEP</userType>
           <baseType>FAIL</baseType>
           <description>The NE call was denied due to general exception.</
description>
     </userDefinedExitType>
     <userDefinedExitType>
           <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
           </neDescriptor>
           <searchPattern>((?s).)*DeniedAccessException((?s).)*</
searchPattern>
           <userType>GEMOTA_DENIED_ACCESS</userType>
           <baseType>FAIL</baseType>
           <description>Connection to platform lost. Access denied.</
description>
     </userDefinedExitType>
     <userDefinedExitType>
           <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
           </neDescriptor>
           <searchPattern>((?s).)*ExpiredAccountException((?s).)*</
searchPattern>
           <userType>GEMOTA_EXP_ACCOUNT</userType>
           <baseType>FAIL</baseType>
           <description>Connection to platform lost. Expired account.</
description>
     </userDefinedExitType>
     <userDefinedExitType>
           <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
           </neDescriptor>
           <searchPattern>((?s).)*TimeOutException((?s).)*</searchPattern>
           <userType>GEMOTA_TIMEOUT</userType>
           <baseType>SOFT_FAIL</baseType>
           <description>Connection to platform lost. Timeout.</description>
     </userDefinedExitType>
     <userDefinedExitType>
           <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
           </neDescriptor>
```

```
          <searchPattern>((?s).)*AlreadyExistsException((?s).)*</
searchPattern>
          <userType>GEMOTA_ALRDY_EXISTS</userType>
          <baseType>FAIL</baseType>
          <description>Subscriber account already exists.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*NoMatchingObjectException((?s).)*</
searchPattern>
          <userType>GEMOTA_NO_MATCH</userType>
          <baseType>FAIL</baseType>
         <description>Account has not been previously declared.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*IllegalArgumentException((?s).)*</
searchPattern>
          <userType>GEMOTA_ILLEGAL_ARG</userType>
          <baseType>FAIL</baseType>
          <description>Argument does not exist.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*InvalidStateException((?s).)*</
searchPattern>
          <userType>GEMOTA_INVALID_STATE</userType>
          <baseType>FAIL</baseType>
          <description>Card manager is not active.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*Missing mandatory parameter -
MSISDN((?s).)*</searchPattern>
          <userType>GEMOTA_MISSPARAM_EX</userType>
          <baseType>FAIL</baseType>
```

```
          <description> Provisioning cartridge Exception due to missing
mandatory MSISDN parameter.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>SIM Card exists in the CardManager</searchPattern>
          <userType>GEMOTA_SIMCARD_EX</userType>
          <baseType>SUCCEED</baseType>
          <description>SIM Card exists in the CardManager.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>SIM Card does not exist in the CardManager</
searchPattern>
          <userType>GEMOTA_SIMCARD_NOEX</userType>
          <baseType>SOFT_FAIL</baseType>
          <description>SIM Card does not exist in the CardManager.</
description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>SUBSCRIBER_CREATED</searchPattern>
          <userType>GEMOTA_SUB_CREATED</userType>
          <baseType>SUCCEED</baseType>
          <description>Successfully created Subscriber Account.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>SUBSCRIBER_DELETED</searchPattern>
          <userType>GEMOTA_SUB_DELETED</userType>
          <baseType>SUCCEED</baseType>
          <description>Successfully deleted Subscriber Account.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
```

```
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>SUBSCRIBER_ACTIVATED</searchPattern>
            <userType>GEMOTA_SUB_ACTIVATED</userType>
            <baseType>SUCCEED</baseType>
            <description>Successfully activated Subscriber Account.</
description>
      </userDefinedExitType>
      <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>SUBSCRIBER_DEACTIVATED</searchPattern>
            <userType>GEMOTA_SUB_DEACTIVE</userType>
            <baseType>SUCCEED</baseType>
            <description>Successfully de-activated Subscriber Account.</
description>
      </userDefinedExitType>
      <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>SUBSCRIBER_UPDATED</searchPattern>
            <userType>GEMOTA_SUB_UPDATED</userType>
            <baseType>SUCCEED</baseType>
            <description>Successfully updated Subscriber Account.</description>
      </userDefinedExitType>
      <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>SUBSCRIBER_QUERIED</searchPattern>
            <userType>GEMOTA_SUB_EXISTS</userType>
            <baseType>SUCCEED</baseType>
            <description>Successfully got details of Subscriber Account.</
description>
      </userDefinedExitType>
      <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>SUBSCRIBER_NOT_FOUND</searchPattern>
            <userType>GEMOTA_SUB_NO_EX</userType>
            <baseType>SOFT_FAIL</baseType>
            <description>Subscriber Account does not exists.</description>
```

```
</userDefinedExitType>
<userDefinedExitType>
     <neDescriptor>
          <softwareLoad>2-6</softwareLoad>
          <technology>OTA</technology>
          <neVendor>GEMP</neVendor>
     </neDescriptor>
     <searchPattern>CARD_CREATED</searchPattern>
     <userType>GEMOTA_SIM_CREATED</userType>
     <baseType>SUCCEED</baseType>
     <description>Successfully created SIM Card.</description>
</userDefinedExitType>
<userDefinedExitType>
     <neDescriptor>
          <softwareLoad>2-6</softwareLoad>
          <technology>OTA</technology>
          <neVendor>GEMP</neVendor>
     </neDescriptor>
     <searchPattern>CARD_DELETED</searchPattern>
     <userType>GEMOTA_SIM_DELETED</userType>
     <baseType>SUCCEED</baseType>
     <description>Successfully deleted SIM Card.</description>
</userDefinedExitType>
<userDefinedExitType>
     <neDescriptor>
          <softwareLoad>2-6</softwareLoad>
          <technology>OTA</technology>
          <neVendor>GEMP</neVendor>
     </neDescriptor>
     <searchPattern>CARD_ACTIVATED</searchPattern>
     <userType>GEMOTA_SIM_ACTIVATED</userType>
     <baseType>SUCCEED</baseType>
     <description>Successfully activated SIM Card.</description>
</userDefinedExitType>
<userDefinedExitType>
     <neDescriptor>
          <softwareLoad>2-6</softwareLoad>
          <technology>OTA</technology>
          <neVendor>GEMP</neVendor>
     </neDescriptor>
     <searchPattern>CARD_DEACTIVATED</searchPattern>
     <userType>GEMOTA_SIM_DEACTIVE</userType>
     <baseType>SUCCEED</baseType>
     <description>Successfully de-activated SIM Card.</description>
</userDefinedExitType>
<userDefinedExitType>
     <neDescriptor>
          <softwareLoad>2-6</softwareLoad>
          <technology>OTA</technology>
          <neVendor>GEMP</neVendor>
     </neDescriptor>
     <searchPattern>CARD_UPDATED</searchPattern>
     <userType>GEMOTA_SIM_UPDATED</userType>
```

```
          <baseType>SUCCEED</baseType>
          <description>Successfully updated SIM Card.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>SECURITY_DELETED</searchPattern>
          <userType>GEMOTA_SECDATE_DEL</userType>
          <baseType>SUCCEED</baseType>
          <description>Successfully deleted the SIM Card security bundle.</
description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>ALL_FILES_QUERIED</searchPattern>
          <userType>GEMOTA_FILE_EXISTS</userType>
          <baseType>SUCCEED</baseType>
          <description>Successfully retrieved the files on the server.</
description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>FILE_NOT_FOUND</searchPattern>
          <userType>GEMOTA_NO_FILE_EXIST</userType>
          <baseType>FAIL</baseType>
          <description>File does not exist on the sever.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>SIM_CARDS_CREATED</searchPattern>
          <userType>GEMOTA_SIMCARDS_ADD</userType>
          <baseType>SUCCEED</baseType>
          <description>A set of sim cards created successfully.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
```

```
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>SIM_CARDS_DELETED</searchPattern>
            <userType>GEMOTA_SIMCARDS_DEL</userType>
            <baseType>SUCCEED</baseType>
            <description>A set of simcards deleted successfully.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>FILE_UPLOADED</searchPattern>
            <userType>GEMOTA_FILE_UPLOADED</userType>
            <baseType>SUCCEED</baseType>
            <description>Successfully uploaded the files on the server.</
description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>SIM_CARDS_UPDATED</searchPattern>
            <userType>GEMOTA_SIMCARDS_MOD</userType>
            <baseType>SUCCEED</baseType>
            <description>A set of sim cards updated successfully.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>SIM_CARDS_SECUR_CREATED</searchPattern>
            <userType>GEMOTA_SIMSECU_ADD</userType>
            <baseType>SUCCEED</baseType>
            <description>Successfully batch loaded new Sim Card settings
information.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>SIM_CARDS_SECUR_UPDATED</searchPattern>
            <userType>GEMOTA_SIMSECU_MOD</userType>
            <baseType>SUCCEED</baseType>
            <description>Successfully batch loaded update of existing Sim Card
settings information.</description>
```

```
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>SIM_CARDS_SECUR_DELETED</searchPattern>
          <userType>GEMOTA_SIMSECU_DEL</userType>
          <baseType>SUCCEED</baseType>
          <description>Successfully deleted a specified Sim Card settings
information.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*CardNotFoundException((?s).)*</
searchPattern>
          <userType>GEMOTA_CARD_NO_EXIST</userType>
          <baseType>FAIL</baseType>
          <description>CardNotFoundException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*ServiceAccessDeniedException((?s).)*</
searchPattern>
          <userType>GEMOTA_ACCESS_DENIED</userType>
          <baseType>FAIL</baseType>
          <description>Service Access denied.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*ObjectAlreadyDefinedException((?s).)*</
searchPattern>
          <userType>GEMOTA_OBJ_ALRDY_EXS</userType>
          <baseType>FAIL</baseType>
          <description>Object may be already defined in the repository.</
description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
```

```
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
           </neDescriptor>
           <searchPattern>((?s).)*FileNotFoundException((?s).)*</
searchPattern>
           <userType>GEMOTA_FILE_NOT_FOUN</userType>
           <baseType>FAIL</baseType>
           <description>No such file or directory.</description>
      </userDefinedExitType>
      <userDefinedExitType>
           <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
           </neDescriptor>
           <searchPattern>((?s).)*Unable to add Smart Cards((?s).)*</
searchPattern>
           <userType>GEMOTA_UNABLE_AD_SIM</userType>
           <baseType>FAIL</baseType>
           <description>Unable to add Smart Cards</description>
      </userDefinedExitType>
      <userDefinedExitType>
           <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
           </neDescriptor>
           <searchPattern>((?s).)*ObjectNotFoundException((?s).)*</
searchPattern>
           <userType>GEMOTA_OBJ_NOT_FOUND</userType>
           <baseType>FAIL</baseType>
           <description>Card Profile does not exist in repository.</
description>
      </userDefinedExitType>
      <userDefinedExitType>
           <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
           </neDescriptor>
           <searchPattern>((?s).)*BadDataConsistencyException((?s).)*</
searchPattern>
           <userType>GEMOTA_BAD_DATA_CONS</userType>
           <baseType>FAIL</baseType>
           <description>BadDataConsistencyException</description>
      </userDefinedExitType>
      <userDefinedExitType>
           <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
           </neDescriptor>
```

```
          <searchPattern>((?s).)*BadParameterFormatException((?s).)*</
searchPattern>
          <userType>GEMOTA_BAD_PARAM</userType>
          <baseType>FAIL</baseType>
          <description>BadParameterFormatException</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*BadRequestFormatException((?s).)*</
searchPattern>
          <userType>GEMOTA_BAD_REQFORMAT</userType>
          <baseType>FAIL</baseType>
          <description>BadRequestFormatException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*CardAssociatedToProfileException((?s).)*</
searchPattern>
          <userType>GEMOTA_CARD_ASSO_PRO</userType>
          <baseType>FAIL</baseType>
          <description>CardAssociatedToProfileException</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*ChannelNotFoundException((?s).)*</
searchPattern>
          <userType>GEMOTA_NO_CHANNEL</userType>
          <baseType>FAIL</baseType>
          <description>ChannelNotFoundException</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*ComponentNotFoundException((?s).)*</
searchPattern>
          <userType>GEMOTA_COMP_NOT_FOUN</userType>
          <baseType>FAIL</baseType>
          <description>ComponentNotFoundException.</description>
```

```
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*ComProblemException((?s).)*</searchPattern>
            <userType>GEMOTA_COMPROB_EXCEP</userType>
            <baseType>FAIL</baseType>
            <description>ComProblemException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*FormattingLibraryException((?s).)*</
searchPattern>
            <userType>GEMOTA_LIBRARY_EXCEP</userType>
            <baseType>FAIL</baseType>
            <description>FormattingLibraryException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*GatewayDownException((?s).)*</searchPattern>
            <userType>GEMOTA_GATEWAY_DOWN</userType>
            <baseType>FAIL</baseType>
            <description>GatewayDownException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*GeneralSystemException((?s).)*</
searchPattern>
            <userType>GEMOTA_GEN_SYS_EXCEP</userType>
            <baseType>FAIL</baseType>
            <description>GeneralSystemException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
```

```
          <searchPattern>((?s).)*InactiveCardException((?s).)*</
searchPattern>
          <userType>GEMOTA_INACTIVE_CARD</userType>
          <baseType>FAIL</baseType>
          <description>InactiveCardException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*InactiveComponentException((?s).)*</
searchPattern>
          <userType>GEMOTA_INACTIVE_COMP</userType>
          <baseType>FAIL</baseType>
          <description>InactiveComponentException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*InactiveProfileException((?s).)*</
searchPattern>
          <userType>GEMOTA_INACTIVE_PROF</userType>
          <baseType>FAIL</baseType>
          <description>InactiveProfileException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*InactiveServiceException((?s).)*</
searchPattern>
          <userType>GEMOTA_INACT_SERVICE</userType>
          <baseType>FAIL</baseType>
          <description>InactiveServiceException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*IncompatibleDataTypeException((?s).)*</
searchPattern>
          <userType>GEMOTA_INCOM_DATATYP</userType>
          <baseType>FAIL</baseType>
          <description>IncompatibleDataTypeException.</description>
```

```
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*MissingInformationException((?s).)*</
searchPattern>
          <userType>GEMOTA_MISSING_INFO</userType>
          <baseType>FAIL</baseType>
          <description>MissingInformationException</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*MissingParameterException((?s).)*</
searchPattern>
          <userType>GEMOTA_MISSNG_PARAM</userType>
          <baseType>FAIL</baseType>
          <description>MissingParameterException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*NoChannelForProtocolException((?s).)*</
searchPattern>
          <userType>GEMOTA_NO_CHAN_PROTO</userType>
          <baseType>FAIL</baseType>
          <description>NoChannelForProtocolException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*NoElementInListException((?s).)*</
searchPattern>
          <userType>GEMOTA_NO_ELEME_LIST</userType>
          <baseType>FAIL</baseType>
          <description>NoElementInListException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
```

```
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*NoFilterFoundException((?s).)*</
searchPattern>
            <userType>GEMOTA_NO_FILTR_FOUN</userType>
            <baseType>FAIL</baseType>
            <description>NoFilterFoundException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*ProductNotConnectedException((?s).)*</
searchPattern>
            <userType>GEMOTA_PROD_NOT_CONN</userType>
            <baseType>FAIL</baseType>
            <description>ProductNotConnectedException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*RequestDeletedByUserException((?s).)*</
searchPattern>
            <userType>GEMOTA_REQDEL_BYUSER</userType>
            <baseType>FAIL</baseType>
            <description>RequestDeletedByUserException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*RequestExecutionFailedException((?s).)*</
searchPattern>
            <userType>GEMOTA_REQ_EXEC_FAIL</userType>
            <baseType>FAIL</baseType>
            <description>RequestExecutionFailedException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>

<searchPattern>((?s).)*RequestExecutionPartialSucessException((?s).)*</
searchPattern>
```

```
          <userType>GEMOTA_REQ_PART_SUCC</userType>
          <baseType>FAIL</baseType>
          <description>RequestExecutionPartialSucessException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*RequestExpiredException((?s).)*</
searchPattern>
          <userType>GEMOTA_REQ_EXPIRED</userType>
          <baseType>FAIL</baseType>
          <description>RequestExpiredException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*RequestNotFoundException((?s).)*</
searchPattern>
          <userType>GEMOTA_REQ_NOT_FOUND</userType>
          <baseType>FAIL</baseType>
          <description>RequestNotFoundException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*SendingButConnectionLostException((?s).)*</
searchPattern>
          <userType>GEMOTA_SEND_CONN_LOS</userType>
          <baseType>FAIL</baseType>
          <description>SendingButConnectionLostException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*SendingRouterException((?s).)*</
searchPattern>
          <userType>GEMOTA_SEND_ROUTR_EX</userType>
          <baseType>FAIL</baseType>
          <description>SendingRouterException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
```

```
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*ServiceDeclarationException((?s).)*</
searchPattern>
          <userType>GEMOTA_SERV_DECLR_EX</userType>
          <baseType>FAIL</baseType>
          <description>ServiceDeclarationException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*ServiceDefinitionException((?s).)*</
searchPattern>
          <userType>GEMOTA_SERVIC_DEF_EX</userType>
          <baseType>FAIL</baseType>
          <description>ServiceDefinitionException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*ServiceExecutionException((?s).)*</
searchPattern>
          <userType>GEMOTA_SERVC_EXECUTE</userType>
          <baseType>FAIL</baseType>
          <description>ServiceExecutionException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*ServiceNotFoundException((?s).)*</
searchPattern>
          <userType>GEMOTA_SERVC_NOTFOUN</userType>
          <baseType>FAIL</baseType>
          <description>ServiceNotFoundException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
```

```
          <searchPattern>((?s).)*ServiceStateException((?s).)*</
searchPattern>
          <userType>GEMOTA_SERV_STATE_EX</userType>
          <baseType>FAIL</baseType>
          <description>ServiceStateException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*ShutdownException((?s).)*</searchPattern>
          <userType>GEMOTA_SHUTDOWN_EX</userType>
          <baseType>FAIL</baseType>
          <description>ShutdownException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*UnhandledException((?s).)*</searchPattern>
          <userType>GEMOTA_UNHANDLED_EX</userType>
          <baseType>FAIL</baseType>
          <description>UnhandledException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*UnknownProtocolException((?s).)*</
searchPattern>
          <userType>GEMOTA_UNKNOW_PROTCL</userType>
          <baseType>FAIL</baseType>
          <description>UnknownProtocolException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*UnknownServiceDeclException((?s).)*</
searchPattern>
          <userType>GEMOTA_UNKN_SER_DECL</userType>
          <baseType>FAIL</baseType>
          <description>UnknownServiceDeclException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
```

```
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*UnknownServiceImplException((?s).)*</
searchPattern>
          <userType>GEMOTA_UNKN_SER_IMPL</userType>
          <baseType>FAIL</baseType>
          <description>UnknownServiceImplException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*UserAccessException((?s).)*</searchPattern>
          <userType>GEMOTA_USER_ACCES_EX</userType>
          <baseType>FAIL</baseType>
          <description>UserAccessException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*FormattingException((?s).)*</searchPattern>
          <userType>GEMOTA_FORMAT_EXCEP</userType>
          <baseType>FAIL</baseType>
          <description>FormattingException.</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*InactiveFormattingTypeException((?s).)*</
searchPattern>
          <userType>GEMOTA_INACTV_FORMAT</userType>
          <baseType>FAIL</baseType>
          <description>InactiveFormattingTypeException</description>
     </userDefinedExitType>
     <userDefinedExitType>
          <neDescriptor>
               <softwareLoad>2-6</softwareLoad>
               <technology>OTA</technology>
               <neVendor>GEMP</neVendor>
          </neDescriptor>
          <searchPattern>((?s).)*SecurityDataException((?s).)*</
searchPattern>
```

```
            <userType>GEMOTA_SECU_DATA_EX</userType>
            <baseType>FAIL</baseType>
            <description>SecurityDataException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*TransportKeyException((?s).)*</
searchPattern>
            <userType>GEMOTA_TRANSPORT_KEY</userType>
            <baseType>FAIL</baseType>
            <description>TransportKeyException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*UnhandledFormattingTypeException((?s).)*</
searchPattern>
            <userType>GEMOTA_UNHANDL_FORMA</userType>
            <baseType>FAIL</baseType>
            <description>UnhandledFormattingTypeException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>SIM_CARDS_EXCEPTION</searchPattern>
            <userType>GEMOTA_SIMCARD_EXCEP</userType>
            <baseType>FAIL</baseType>
            <description>Exceptions occurred for SIM Cards.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*ParsingException((?s).)*</searchPattern>
            <userType>GEMOTA_PARSE_EXCEP</userType>
            <baseType>FAIL</baseType>
            <description>ParsingException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
```

```
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>GEMOTA_NO_ICCID_FOUN</searchPattern>
            <userType>GEMOTA_NO_ICCID_FOUN</userType>
            <baseType>FAIL</baseType>
            <description>The associated ICCID for the simcard is not found.</
description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>GEMOTA_NO_FILE_FOUND</searchPattern>
            <userType>GEMOTA_NO_FILE_FOUND</userType>
            <baseType>FAIL</baseType>
            <description>No such file or directory.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*DatabaseException((?s).)*</searchPattern>
            <userType>GEMOTA_DATABASEEXCEP</userType>
            <baseType>FAIL</baseType>
            <description>DatabaseException.</description>
        </userDefinedExitType>
        <userDefinedExitType>
            <neDescriptor>
                <softwareLoad>2-6</softwareLoad>
                <technology>OTA</technology>
                <neVendor>GEMP</neVendor>
            </neDescriptor>
            <searchPattern>((?s).)*BadXMLDefinitionException((?s).)*</
searchPattern>
            <userType>GEMOTA_BAD_XML_DEFN</userType>
            <baseType>FAIL</baseType>
            <description>BadXMLDefinitionException.</description>
        </userDefinedExitType>
</serviceModel>
```

4

Service Definition

The Gemplus OTA 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 42: 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 42: ASDL parameter information

Item	Description
Type	Indicates one of the following parameter types: ◆ 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. For more information on parameter types, refer to the *ASAP Developer Reference*.
Class	Indicates one of the following parameter classifications: ◆ 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_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER
- C_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER-SIM-CARD
- C_GEMP-OTA_2-6_ADD_SIM-CARDS
- C_GEMP-OTA_2-6_ADD_SIM-CARD-SECUR-BUNDLE
- C_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN
- C_GEMP-OTA_2-6_ADD_SUBSCRIBER
- C_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD
- C_GEMP-OTA_2-6_CREATE_SUBSCRIBER
- C_GEMP-OTA_2-6_CREATE_SUBSCRIBER-SIM-CARD
- C_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER
- C_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER-SIM-CARD
- C_GEMP-OTA_2-6_DEL_SIM-CARDS
- C_GEMP-OTA_2-6_DEL_SIM-CARD-MSISDN
- C_GEMP-OTA_2-6_DEL_SIM-CARD-SECURITY-DATA
- C_GEMP-OTA_2-6_DEL_SUBSCRIBER
- C_GEMP-OTA_2-6_DEL_SUBSCRIBER-SIM-CARD
- C_GEMP-OTA_2-6_MOD_SIM-CARDS
- C_GEMP-OTA_2-6_MOD_SIM-CARD-SECUR-BUNDLE
- C_GEMP-OTA_2-6_MOD_SUBSCRIBER
- C_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD
- C_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARDS
- C_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARD-SECUR-BUNDLE
- C_GEMP-OTA_2-6_QRY_SUBSCRIBER
- C_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD
- C_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS
- C_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE

C_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER

Activates a subscriber on the OTA.

Table 43: C_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
MSISDN	Mobile station ISDN.	5 - 15 digits		S	R
NE_ID_GEMP-OTA	The remote network element name.			S	R

Mapping to ASDLs

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

Table 44: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER	A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER

C_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER-SIM-CARD

Activates the subscriber SIM Card on the OTA.

Table 45: C_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
ICCID	The ICCID of the card.			S	O
IMSI	International mobile subscriber identity of the card.			S	O
MSISDN	Mobile station ISDN number of 5 to 20 digits.			S	O
NE_ID_GEMP-OTA	The remote network element name.			S	R

Mapping to ASDLs

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

Table 46: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER-SIM-CARD	A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER-SIM-CARD

C_GEMP-OTA_2-6_ADD_SIM-CARDS

Adds a set of SIM Cards.

Table 47: C_GEMP-OTA_2-6_ADD_SIM-CARDS

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILEPATH	The absolute path to the directory on the ASAP host file system where the file is located, e.g. /export/home/sunen290	Length should not exceed 255 characters.		S	R
FILENAME	The name of the SIM card details file on the ASAP host's file system, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R

Mapping to ASDLs

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

Table 48: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_ADD_SIM-CARDS	A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS
	A_GEMP-OTA_2-6_ADD_SIM-CARDS

C_GEMP-OTA_2-6_ADD_SIM-CARD-SECUR-BUNDLE

Batch loading of new Sim Card settings information.

Table 49: C_GEMP-OTA_2-6_ADD_SIM-CARD-SECUR-BUNDLE

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILEPATH	The absolute path to the directory on the ASAP host file system where the file is located, e.g. /export/home/sunen290	Length should not exceed 255 characters.		S	R
FILENAME	The name of the SIM card details file on the ASAP host's file system, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R

Mapping to ASDLs

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

Table 50: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_ADD_SIM-CARD-SECUR-BUNDLE	A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE
	A_GEMP-OTA_2-6_ADD_SIM-CARD-SECUR-BUNDLE

C_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN

Assigns MSISDN to a subscriber SIM Card on the OTA.

Table 51: C_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN

Parameter Name	Description	Range	Default Value	Type	Class
ICCID	The ICCID of the card.			S	O
IMSI	International Mobile Subscriber Identity of the card.			S	O
MSISDN	Mobile Station ISDN Number of 5 to 20 digits of the card.			S	R
NE_ID_GEMP-OTA	The remote network element name.			S	R

Mapping to ASDLs

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

Table 52: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN	A_GEMP-OTA_2-6_ADD_SIM-CARD-MSISDN

C_GEMP-OTA_2-6_ADD_SUBSCRIBER

Adds a subscriber on the OTA.

Table 53: C_GEMP-OTA_2-6_ADD_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
CREATOR_ID	The user account creator ID.			S	O
MSISDN	Mobile station ISDN.	5 - 15 digits		S	R
NE_ID_GEMP-OTA	The remote network element name.			S	R

Mapping to ASDLs

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

Table 54: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_ADD_SUBSCRIBER	A_GEMP-OTA_2-6_ADD_SUBSCRIBER

C_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD

Adds a subscriber SIM card on the OTA.

Table 55: C_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
CARD_PROFILE	The name of the associated card profile.			S	R
ICCID	The ICCID of the card.			S	R
IMSI	International mobile subscriber identity of the card.			S	O
MSISDN	Mobile station ISDN number of 5 to 20 digits.			S	O

Table 55: C_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
NE_ID_GEMP-OTA	The remote network element name.			S	R
STATE	The card state. Valid values are : STATE_ACTIVE, STATE_INACTIVE, STATE_MARK_DELETED.			S	O

Mapping to ASDLs

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

Table 56: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD	A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD

C_GEMP-OTA_2-6_CREATE_SUBSCRIBER

Creates an active subscriber on the OTA. This adds, updates and activates a subscriber on the OTA.

Table 57: C_GEMP-OTA_2-6_CREATE_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
AUTHENTICATION_METHOD	User account authentication method. Authentication methods can be by password, by mobile challenge, or by certificate.			S	O
AUTHENTICATION_VALUE	Data required in order to achieve authentication of the account.			S	O
CARD_IDENTIFICATION	Owned card identifier (to establish a link with the card manager database).			S	O

Table 57: C_GEMP-OTA_2-6_CREATE_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
CREATOR_ID	The user account creator ID.			S	O
EXPIRATION_DATE	User account expiration date.			S	O
FULL_NAME	Name of the individual user corresponding to the account.			S	O
ICCID	Account's card identification.			S	O
MSISDN	Mobile station ISDN.	5 - 15 digits		S	R
NEW_ID	New subscriber ID.			S	O
NE_ID_GEMP-OTA	The remote network element name.			S	R
PROFILE_NAME	Existing Profile Name.			S	R
SUBSCRIBER_REFERENCE	Free text to be used to make an external link with the operator’s subscriber database.			S	O

Mapping to ASDLs

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

Table 58: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_CREATE_SUBSCRIBER	A_GEMP-OTA_2-6_ADD_SUBSCRIBER
	A_GEMP-OTA_2-6_MOD_SUBSCRIBER
	A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER

C_GEMP-OTA_2-6_CREATE_SUBSCRIBER-SIM-CARD

Creates an active subscriber SIM card on the OTA. This adds, and activates a subscribers SIM card on the OTA.

Table 59: C_GEMP-OTA_2-6_CREATE_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
CARD_PROFILE	The name of the associated card profile.			S	R
ICCID	The ICCID of the card.			S	R
IMSI	International mobile subscriber identity of the card.			S	O
LINKED_CARD	The linked card identifier.			S	O
MSISDN	Mobile station ISDN number of 5 to 20 digits.			S	O
NE_ID_GEMP-OTA	The remote network element name.			S	R
STATE	The card state. Valid values are : STATE_ACTIVE, STATE_INACTIVE, STATE_MARK_DELETED.			S	O

Mapping to ASDLs

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

Table 60: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_CREATE_SUBSCRIBER-SIM-CARD	A_GEMP-OTA_2-6_ADD_SUBSCRIBER-SIM-CARD
	A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD
	A_GEMP-OTA_2-6_ACTIVATE_SUBSCRIBER-SIM-CARD

C_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER

Deactivates a subscriber on the OTA.

Table 61: C_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
MSISDN	Mobile station ISDN.	5 - 15 digits		S	R
NE_ID_GEMP-OTA	The remote network element name.			S	R

Mapping to ASDLs

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

Table 62: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER	A_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER

C_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER-SIM-CARD

Deactivates a subscriber SIM Card on the OTA.

Table 63: C_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
ICCID	The ICCID of the card.			S	O
IMSI	International mobile subscriber identity of the card.			S	O
MSISDN	Mobile station ISDN number of 5 to 20 digits.			S	O
NE_ID_GEMP-OTA	The remote network element name.			S	R

Mapping to ASDLs

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

Table 64: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER-SIM-CARD	A_GEMP-OTA_2-6_DEACTIVATE_SUBSCRIBER-SIM-CARD

C_GEMP-OTA_2-6_DEL_SIM-CARDS

Deletes a set of SIM Cards.

Table 65: C_GEMP-OTA_2-6_DEL_SIM-CARDS

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILEPATH	The absolute path to the directory on the ASAP host file system where the file is located, e.g. /export/home/sunen290	Length should not exceed 255 characters.		S	R
FILENAME	The name of the SIM card details file on the ASAP host's file system, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R

Mapping to ASDLs

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

Table 66: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_DEL_SIM-CARDS	A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS
	A_GEMP-OTA_2-6_DEL_SIM-CARDS

C_GEMP-OTA_2-6_DEL_SIM-CARD-MSISDN

Unassign MSISDN to a subscriber SIM Card on the OTA.

Table 67: C_GEMP-OTA_2-6_DEL_SIM-CARD-MSISDN

Parameter Name	Description	Range	Default Value	Type	Class
ICCID	The ICCID of the card.			S	O
IMSI	International Mobile Subscriber Identity of the card.			S	O
NE_ID_GEMP-OTA	The remote network element name.			S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

Mapping to ASDLs

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

Table 68: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_DEL_SIM-CARD-MSISDN	A_GEMP-OTA_2-6_QRY_SIM-CARD-MSISDN-RB
	A_GEMP-OTA_2-6_DEL_SIM-CARD-MSISDN

C_GEMP-OTA_2-6_DEL_SIM-CARD-SECURITY-DATA

Deletes the Security Data Bundle associated with a subscriber SIM Card on the OTA.

Table 69: C_GEMP-OTA_2-6_DEL_SIM-CARD-SECURITY-DATA

Parameter Name	Description	Range	Default Value	Type	Class
ICCID	The ICCID of the card.			S	O
IMSI	International Mobile Subscriber Identity of the card.			S	O
NE_ID_GEMP-OTA	The remote network element name.			S	R

Mapping to ASDLs

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

Table 70: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_DEL_SIM-CARD-SECURITY-DATA	A_GEMP-OTA_2-6_DEL_SIM-CARD-SECURITY-DATA

C_GEMP-OTA_2-6_DEL_SUBSCRIBER

Deletes a subscriber from the OTA.

Table 71: C_GEMP-OTA_2-6_DEL_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
MSISDN	Mobile station ISDN.	5 - 15 digits		S	R
NE_ID_GEMP-OTA	The remote network element name.			S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

Mapping to ASDLs

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

Table 72: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_DEL_SUBSCRIBER	A_GEMP-OTA_2-6_QRY_SUBSCRIBER-RB
	A_GEMP-OTA_2-6_DEL_SUBSCRIBER

C_GEMP-OTA_2-6_DEL_SUBSCRIBER-SIM-CARD

Deletes a subscriber SIM Card from the OTA.

Table 73: C_GEMP-OTA_2-6_DEL_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
ICCID	The ICCID of the card.			S	O
IMSI	International mobile subscriber identity of the card.			S	O
MSISDN	Mobile station ISDN number of 5 to 20 digits.			S	O
NE_ID_GEMP-OTA	The remote network element name.			S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

Mapping to ASDLs

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

Table 74: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_DEL_SUBSCRIBER-SIM-CARD	A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD-RB
	A_GEMP-OTA_2-6_DEL_SUBSCRIBER-SIM-CARD

C_GEMP-OTA_2-6_MOD_SIM-CARDS

Updates a set of SIM Cards.

Table 75: C_GEMP-OTA_2-6_MOD_SIM-CARDS

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILEPATH	The absolute path to the directory on the ASAP host file system where the file is located, e.g. /export/home/sunen290	Length should not exceed 255 characters.		S	R
FILENAME	The name of the SIM card details file on the ASAP host's file system, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

Mapping to ASDLs

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

Table 76: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_MOD_SIM-CARDS	A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS
	A_GEMP-OTA_2-6_QRY_SIM-CARDS-RB
	A_GEMP-OTA_2-6_MOD_SIM-CARDS

C_GEMP-OTA_2-6_MOD_SIM-CARD-SECUR-BUNDLE

Batch loading for updating of existing Sim Card settings information.

Table 77: C_GEMP-OTA_2-6_MOD_SIM-CARD-SECUR-BUNDLE

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILEPATH	The absolute path to the directory on the ASAP host file system where the file is located, e.g. /export/home/sunen290	Length should not exceed 255 characters.		S	R
FILENAME	The name of the SIM card details file on the ASAP host's file system, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R

Mapping to ASDLs

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

Table 78: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_MOD_SIM-CARD-SECUR-BUNDLE	A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE
	A_GEMP-OTA_2-6_MOD_SIM-CARD-SECUR-BUNDLE

C_GEMP-OTA_2-6_MOD_SUBSCRIBER

Modify subscriber on the OTA.

Table 79: C_GEMP-OTA_2-6_MOD_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
AUTHENTICATION_METHOD	User account authentication method. Authentication methods can be by password, by mobile challenge, or by certificate.	Valid values - AUTH_CERTIFICATE, AUTH_MOBILE_CHALLENGE, AUTH_PASSWORD		S	O
AUTHENTICATION_VALUE	Data required in order to achieve authentication of the account.			S	O
CARD_IDENTIFICATION	Owned card identifier (to establish a link with the card manager database).			S	O

Table 79: C_GEMP-OTA_2-6_MOD_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
EXPIRATION_DATE	User account expiration date.	Valid date format - YYYY/ MM/DD HR:MIN: SEC, it possible to provide only the date without the time value		S	O
FULL_NAME	Name of the individual user corresponding to the account.			S	O
MSISDN	Mobile station ISDN.	5 - 15 digits		S	R
NEW_ID	New subscriber ID.			S	O
NE_ID_GEMP-OTA	The remote network element name.			S	R
PROFILE_NAME	Existing Profile Name.			S	R
SUBSCRIBER_REFERENCE	Free text to be used to make an external link with the operator's subscriber database.			S	O
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

Mapping to ASDLs

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

Table 80: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_MOD_SUBSCRIBER	A_GEMP-OTA_2-6_QRY_SUBSCRIBER-RB
	A_GEMP-OTA_2-6_MOD_SUBSCRIBER

C_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD

Modifies a subscriber SIM Card on the OTA.

Table 81: C_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
CARD_PROFILE	The name of the associated card profile.			S	O
ICCID	The ICCID of the card.			S	O
IMSI	International mobile subscriber identity of the card.			S	O
LINKED_CARD	The linked card identifier.			S	O
MSISDN	Mobile Station ISDN Number of 5 to 20 digits of the card.			S	O
NE_ID_GEMP-OTA	The remote network element name.			S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

Mapping to ASDLs

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

Table 82: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD	A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD-RB
	A_GEMP-OTA_2-6_MOD_SUBSCRIBER-SIM-CARD

C_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARDS

Returns the list of available files on the server.

Table 83: C_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARDS

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

Mapping to ASDLs

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

Table 84: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARDS	A_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARDS

C_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARD-SECUR-BUNDLE

Returns the list of available files on the server.

Table 85: C_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARD-SECUR-BUNDLE

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

Mapping to ASDLs

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

Table 86: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARD-SECUR-BUNDLE	A_GEMP-OTA_2-6_QRY-ALL-FILES_SIM-CARD-SECUR-BUNDLE

C_GEMP-OTA_2-6_QRY_SUBSCRIBER

Queries a subscriber on the OTA.

Table 87: C_GEMP-OTA_2-6_QRY_SUBSCRIBER

Parameter Name	Description	Range	Default Value	Type	Class
MSISDN	Mobile station ISDN.	5 - 15 digits		S	O
NE_ID_GEMP-OTA	The remote network element name.			S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

Mapping to ASDLs

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

Table 88: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_QRY_SUBSCRIBER	A_GEMP-OTA_2-6_QRY_SUBSCRIBER

C_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD

Queries a subscriber SIM card on the OTA. This service returns the status if the subscribers SIM Card exists or not.

Table 89: C_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD

Parameter Name	Description	Range	Default Value	Type	Class
ICCID	The ICCID of the card.			S	O
IMSI	International mobile subscriber identity of the card.			S	O
MSISDN	Mobile station ISDN number of 5 to 20 digits			S	O
NE_ID_GEMP-OTA	The remote network element name.			S	R
RETURN_DATA_PREFIX	Parameter to identify ASDLs for multiple nodes.			S	O

Mapping to ASDLs

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

Table 90: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD	A_GEMP-OTA_2-6_QRY_SUBSCRIBER-SIM-CARD

C_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS

Upload file containing a set of Sim Card settings information.

Table 91: C_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R
FILEPATH	The absolute path to the directory on the ASAP host file system where the file is located, e.g. /export/home/sunen290.	Length should not exceed 255 characters.		S	R
FILENAME	The name of the SIM card details file on the ASAP host's file system, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R

Mapping to ASDLs

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

Table 92: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS	A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARDS

C_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE

Upload file containing a set of Sim Card settings information.

Table 93: C_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE

Parameter Name	Description	Range	Default Value	Type	Class
MCLI	The remote network element name.			S	R

Table 93: C_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE

Parameter Name	Description	Range	Default Value	Type	Class
FILEPATH	The absolute path to the directory on the ASAP host file system where the file is located, e.g. /export/home/sunen290.	Length should not exceed 255 characters.		S	R
FILENAME	The name of the SIM card details file on the ASAP host's file system, e.g. CC_SIM_6302.xml.	Length should not exceed 255 characters.		S	R

Mapping to ASDLs

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

Table 94: CSDL to ASDL Mapping

CSDL	ASDL
C_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE	A_GEMP-OTA_2-6_UPLOAD-FILE_SIM-CARD-SECUR-BUNDLE

5

Configuring ASAP to Support Additional NE Instances

You can configure ASAP to support the GEMP-OTA_2-6-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 Gemplus OTA cartridge.

```
<?xml version="1.0" encoding="UTF-8"?>
<activationConfig xmlns="http://www.metasolv.com/ServiceActivation/
  2003/ActivationConfig" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance" xsi:schemaLocation="http://www.metasolv.com/ServiceActi-
  vation/2003/ActivationConfig
C:\Trabajos\ASAP\schema\xsd\ActivationConfig.xsd">
   <connectionPool name="GPO_POOL">
      <device name="GEMP-OTA_2-6-HOST_Corba_dev_1">
         <environment>MY_ASAP_SYS</environment>
         <lineType>GENERIC_MESSAGE_BASED_CONNECTION</lineType>
      </device>
   </connectionPool>
   <nepServer name="NEP_GOTA" xsi:type="NEPServerType">
      <description>NEP Server for Gemplus OTA</description>
      <system>$ASAP_ENV</system>
      <territory>$ASAP_ENV</territory>
      <diagnosticFilename>NEP_GOTA.diag</diagnosticFilename>
      <diagnosticLevel>LOW_LEVEL</diagnosticLevel>
      <autoStart>true</autoStart>
      <controlServer>$CTRL</controlServer>
      <interfaceHostname>
         <hostname>MY_LOCALHOST</hostname>
      </interfaceHostname>
      <interfacePort>MY_INT_PORT</interfacePort>
      <secondaryPool/>
      <jinterpreterPort>MY_JINT_PORT</jinterpreterPort>
      <enableJInterpreter>true</enableJInterpreter>
   </nepServer>
   <element name="GEMP-OTA_2-6-HOST">
      <technology>GEMP-OTA</technology>
      <softwareLoad>2-6</softwareLoad>
      <nepServerName>NEP_GOTA</nepServerName>
      <primaryPool>GPO_POOL</primaryPool>
      <maximumConnections>1</maximumConnections>
      <dropTimeout>2</dropTimeout>
      <spawnThreshold>10</spawnThreshold>
```

```
<killThreshold>8</killThreshold>
<routingElement name="GEMP-OTA_2-6-HOST">
   <atomicService/>
</routingElement>
<communicationParameter>
   <label>HOST_IPADDR</label>
   <value>
      <value>10.1.50.45</value>
   </value>
   <description>The name (or IP Address) of the host the Naming
   Service is running on.</description>
   <deviceName>COMMON_DEVICE_CFG</deviceName>
   <lineType>GENERIC_MESSAGE_BASED_CONNECTION</lineType>
</communicationParameter>
<communicationParameter>
   <label>PORT</label>
   <value>
      <value>23</value>
   </value>
   <description>The port the Naming Service is listening on.
   </description>
   <deviceName>COMMON_DEVICE_CFG</deviceName>
   <lineType>GENERIC_MESSAGE_BASED_CONNECTION</lineType>
</communicationParameter>
<communicationParameter>
   <label>HOST_USERID</label>
   <value>
      <value>userid</value>
   </value>
   <description>User name for login.</description>
   <deviceName>COMMON_DEVICE_CFG</deviceName>
   <lineType>GENERIC_MESSAGE_BASED_CONNECTION</lineType>
</communicationParameter>
<communicationParameter>
   <label>PRODUCT</label>
   <value>
      <value>RCA1</value>
   </value>
   <description>Name of the Product.</description>
   <deviceName>COMMON_DEVICE_CFG</deviceName>
   <lineType>GENERIC_MESSAGE_BASED_CONNECTION</lineType>
</communicationParameter>
<communicationParameter>
   <label>HOST_PASSWORD</label>
   <value>
      <value>passwd</value>
   </value>
   <description>Password for login.</description>
   <deviceName>COMMON_DEVICE_CFG</deviceName>
   <lineType>GENERIC_MESSAGE_BASED_CONNECTION</lineType>
</communicationParameter>
<communicationParameter>
   <label>RESPONSELOG</label>
   <value>
      <value>TRUE</value>
   </value>
   <description>Flag to turn off or on Response log.
```

```
            </description>
            <deviceName>COMMON_DEVICE_CFG</deviceName>
            <lineType>GENERIC_MESSAGE_BASED_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.

To extract source files

1. If necessary, create a repository directory under /Gemplus_ota_2_6, 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 14, for directions on how to load a new XML file.