

Oracle® Communications Billing and Revenue Management

Telco Integration



Release 15.2
G35856-01
January 2026



Copyright © 2017, 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

About This Content

Part I Installing Telco Services Managers

1 Installing Services Framework Manager

Installing Services Framework Manager	1
Uninstalling Services Framework Manager	1

2 Installing and Configuring GSM Manager and Provisioning Data Manager

About the GSM Manager Components	1
Mandatory Configuration Tasks	1
Installing GSM Manager and Provisioning Data Manager	2
Configuring and Testing GSM Manager and Provisioning Data Manager	2
Applying the Correct Partitioning Layout to Event Tables	3
Configuring the Provisioning Data Manager	3
Provisioning Data Manager Configuration File Entries	4
Connecting the Connection Manager to the Provisioning Data Manager	4
Creating Network Elements	5
Loading GSM Configuration Files	5
Loading Provisioning	5
Loading Service Order States	6
Loading GSM Provisioning Flag Definitions	6
Loading the Permit Mapping Files	6
Configuring Event Notification for GSM Manager	6
Uninstalling GSM Manager and Provisioning Data Manager	7

3 Installing GPRS Manager

Installing GPRS Manager	1
Configuring Event Notification for GPRS Manager	1

Part II Managing Telephone Number Inventory

4 About Managing Telephone Numbers

About Managing Numbers	1
About Using Number Administration Center	1
About Managing Blocks and Numbers	2
Getting Information about Number Usage	2
About Managing Telephone Numbers in Customer Center	2
About Number Device States	2
Taking Numbers Out of Quarantine	3
Displaying the Number Inventory	3
Creating a Block of Telephone Numbers	4
About Customizing Block Creation and Modification	5
About Modifying a Block	5
Changing the Size of the Number Block	6
Splitting a Block	7
Updating Telephone Numbers	8
About Number Customization Options	10

5 Installing and Configuring Number Manager and Number Administration Center

Mandatory Configuration Tasks	1
Installing Number Manager	2
Installing Number Administration Center	2
Configuring Event Notification for Number Manager	2
Customizing Number Quarantine	3
Changing the Default Quarantine Period	3
Changing the Quarantine Status Manually	4
Changing How Number Quarantine Works	5
Creating Network Elements	5
Creating Number Categories	6
Creating Vanity Types	6
Customizing How Service Types Are Used	7
Changing the Number Administration Center Number Display Format	7
How BRM Chooses a Mask	7
About Truncating Numbers by Using Masks	7
Changing the Number Display Format	8

Creating a Custom Number Format Class	9
Configuring Search Performance in Number Administration Center	10
Uninstalling Number Manager	10

Part III Managing SIM Card Inventory

6 About Managing SIM Card Inventory

About Managing SIM Cards	1
About Creating SIM Cards	1
About SIM Card Orders	3
About Managing Orders	4
About SIM Card Request and Response Files	5
About SIM Card Pre-Provisioning	6
Enabling SIM Card Pre-Provisioning	7
About Associating a SIM Card with a Customer's Service	8
About Assigning Numbers	9
About SIM Card Device States	10
Changing the Network Element of SIM Cards	11
About SIM Card Customization Options	11
Getting Information About SIM Card Usage	11

7 Installing and Configuring SIM Manager and SIM Administration Center

Mandatory Configuration Tasks	1
Installing SIM Manager	2
Installing SIM Administration Center	2
Configuring Event Notification for SIM Manager	2
Customizing SIM Card and Service Associations	3
Supporting Encrypted KI Values with Oracle	3
Creating Network Elements	4
Creating SIM Card Formats	5
Customizing SIM Card Device States	5
Adding Device State Names	5
Customizing Service Types	5
Customizing How to Pre-Provision SIM Cards	6
Specifying the Pre-Provisioning Service	6
Changing the Pre-Provisioning MSISDN	7
Loading Order Status Definitions	8
Customizing SIM Administration Center	8
Specifying the Maximum Quantity of SIM Cards in a Request File	8
Specifying the Maximum Quantity of SIM Cards in an Order	9

Specifying the Information You Want to Receive about SIM Cards	9
Specifying whether the Response File Includes a Check Digit	9
Specifying the Size of Searches	9
Creating SIM Cards for Testing	10
Uninstalling SIM Manager	11

Part IV Managing Voucher Inventory

8 About Managing Voucher Inventory

About Managing Voucher Cards	1
About Creating Vouchers	2
About Voucher Orders	3
About Voucher Bundles and Deals	4
About Dealers	4
About Recharge Card Types	4
About Voucher Details	5
About Managing Orders	5
About Modifying Orders	6
About Voucher Request and Response Files	6
Vendor Request Files and Voucher Orders	7
Vendor Response Files and Voucher Devices	7
About Request and Response File Templates	8
About Request and Response File Encryption	10
About Managing Vouchers	10
About Voucher Device States	10
Managing Expired Vouchers	11
About Voucher Customization Options	11
Getting Information About Voucher Usage	11
How Voucher Association Works	11
Customizing How Voucher Manager Manages Devices	12
Customizing Voucher Creation	12
Customizing Voucher Validation	12
Customizing Voucher Association	13
Customizing Voucher/Service Association	14
Customizing How Voucher Manager Manages Orders	14
Customizing Order Creation	14
Customizing Order Association	15
Customizing Order Attributes	15
Canceling Orders	16

9 Installing and Configuring Voucher Manager and Voucher Administration Center

Mandatory Configuration Tasks	1
Installing Voucher Manager	2
Installing Voucher Administration Center	2
Loading Recharge Card Details	2
Loading Dealer Details	3
Loading Voucher Device States	3
Loading Voucher Order States	4
Customizing and Loading Voucher Service Associations	4
Loading Voucher Details	5
Creating Vouchers for Testing	5
Uninstalling Voucher Manager	7

10 Voucher Manager Utilities

load_pin_dealers	1
load_pin_recharge_card_type	2
load_pin_voucher_config	3
pin_voucher_expiration	5

Part V Managing IP Address and APN Inventories

11 About IP Address Manager

About Managing IP Address and APN Device Inventories	1
About IP Address Manager	2
About IP and APN Device States	2
IP Address Device States	2
APN Device States	3
Associated Services Lists	3
IP Address Manager Configuration	3
Important /device Object Fields	4

12 Installing and Configuring IP Address Manager and IP Address Administration Center

Installing IP Address Manager	1
Installing IP Address Administration Center	1

Configuring IP Address Manager	1
Customizing IP Address and APN Device Life Cycles	2
Customizing IP Address Device States	2
Customizing APN Device States	2
Customizing IP Address and APN Service Association Lists	3
Customizing the IP Service Association List	3
Customizing the APN Service Association List	3
Uninstalling IP Address Manager	3

Part VI Telco Manager Utilities

13 Telco Service Manager Utilities

load_pin_network_elements	1
load_pin_num_config	2
load_pin_sim_config	3
pin_change_num_quarantine	4

About This Content

This guide describes how to install and configure the following Oracle Communication Billing and Revenue Management (BRM) optional managers:

- GPRS Manager
- GSM Manager
- IP Address Manager
- Number Manager
- Services Framework Manager
- SIM Manager
- Voucher Manager

Audience

This guide is for BRM system administrators and developers.

Part I

Installing Telco Services Managers

Part I describes how to install Oracle Communications Billing and Revenue Management (BRM) telco services managers. It contains the following chapters:

- [Installing Services Framework Manager](#)
- [Installing and Configuring GSM Manager and Provisioning Data Manager](#)
- [Installing GPRS Manager](#)

1

Installing Services Framework Manager

Learn how to install the Oracle Communications Billing and Revenue Management (BRM) Services Framework Manager software.

Topics in this document:

- [Installing Services Framework Manager](#)
- [Uninstalling Services Framework Manager](#)

Installing Services Framework Manager

 **Note**

If you already installed Services Framework Manager, you must uninstall its features before reinstalling them.

To install Services Framework Manager, see "Installing Individual BRM Components" in *BRM Installation Guide*.

Uninstalling Services Framework Manager

To uninstall Services Framework Manager, see "Uninstalling Optional Components" in *BRM Installation Guide*.

Installing and Configuring GSM Manager and Provisioning Data Manager

Learn how to install Oracle Communications Billing and Revenue Management (BRM) GSM Manager and the Provisioning Data Manager.

Topics in this document:

- [About the GSM Manager Components](#)
- [Mandatory Configuration Tasks](#)
- [Installing GSM Manager and Provisioning Data Manager](#)
- [Configuring and Testing GSM Manager and Provisioning Data Manager](#)
- [Configuring Event Notification for GSM Manager](#)
- [Uninstalling GSM Manager and Provisioning Data Manager](#)

About the GSM Manager Components

The GSM Manager installation includes two sets of components:

- GSM server components
- Provisioning Data Manager

You can choose to install either or both of these sets of components on the same machine or on different machines.

GSM Manager Customer Center Extension adds functionality to the core Customer Center.

 **Note**

The GSM Self-Care Manager pages are installed with Self-Care Manager.

Mandatory Configuration Tasks

After you install the GSM Manager software, you must:

- Be sure that the Connection Manager (CM) and Provisioning Data Manager are configured correctly, and if not, configure them. In most cases, configuration is done during installation, but you should verify that it was done correctly. See the following topics:
 - [Configuring the Provisioning Data Manager](#)
 - [Connecting the Connection Manager to the Provisioning Data Manager](#)
- Configure the event notification feature required for GSM provisioning. See "[Configuring Event Notification for GSM Manager](#)".

- Load the supplementary service and service-level ERA provisioning flag descriptions that Customer Center displays. See "[Loading GSM Provisioning Flag Definitions](#)".
- Load provisioning tags. See "Specifying the Details to Add to the Service Order" in *BRM Provisioning Services*.

Installing GSM Manager and Provisioning Data Manager

Note

If you already installed the product, you must uninstall its features before reinstalling them.

To install GSM Manager and Provisioning Data Manager, see "Installing Individual BRM Components" in *BRM Installation Guide*.

For information on starting the Provisioning Data Manager, see "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

Note

You should verify that the GSM Manager FMs were added to the CM configuration file. See "Using Configuration Files" in *BRM System Administrator's Guide*.

Configuring and Testing GSM Manager and Provisioning Data Manager

Note

Install GSM Manager and the Provisioning Data Manager software before starting these configuration tasks.

In most cases, the server configuration tasks are handled during installation. However, you might need to manually configure some components. You should check the configuration after installation to verify that it is correct.

To configure your GSM Manager system:

1. [Applying the Correct Partitioning Layout to Event Tables](#)
2. [Configuring the Provisioning Data Manager](#)
3. [Connecting the Connection Manager to the Provisioning Data Manager](#)
4. [Creating Network Elements](#)
5. [Loading GSM Configuration Files](#)

For information about configuration files, see "Using Configuration Files" in *BRM System Administrator's Guide*.

Applying the Correct Partitioning Layout to Event Tables

Note

If your event tables are partitioned, this is a mandatory configuration task. If your event tables are not partitioned, skip this task.

When you install GSM Manager, you create additional event tables. Run the **partition_utils** utility from the *BRM_home/apps/partition_utils* directory. See "Adding Partitions" and "partition_utils" in *BRM System Administrator's Guide*.

Configuring the Provisioning Data Manager

To configure the Provisioning Data Manager, you edit the Provisioning Data Manager configuration file (*BRM_home/sys/dm_prov_telco/pin.conf*).

1. Open the Provisioning Data Manager configuration file (*BRM_home/sys/dm_prov_telco/pin.conf*).
2. Edit the file according to the instructions in it.

Note

You must edit the **dm_provision prov_ptr** entry to connect to the provisioning system.

- For information about the entries specific to the Provisioning Data Manager, see "[Provisioning Data Manager Configuration File Entries](#)".
- For information about Data Manager configuration, see *BRM System Administrator's Guide*.

The following entries should appear in the Provisioning Data Manager **pin.conf** file and have the correct settings. See the **pin.conf** file for more details.

```
- dm dm_db_no 0.0.10.2 / 0
- dm dm_logfile BRM_home/dm_prov_telco/dm_prov_telco/dm_provision.pinlog
- dm dm_port 11990
- dm dm_sm_obj BRM_home/sys/dm_prov_telco/dm_prov_telco.so
- dm_provision prov_ptr ip Hostname_of_the_provisioning_system20000
- dm_provision connect_retry_interval 0
- dm_provision connect_retries 0
- dm dm_trans_timeout 0
```

3. Save the file.

Note

Before starting the Provisioning Data Manager, you must start the provisioning system.

Provisioning Data Manager Configuration File Entries

[Table 2-1](#) lists the Provisioning Data Manager configuration file entries.

Table 2-1 Provisioning Data Manager Configuration File Entries

Entry	Description
- dm dm_db_no	Specifies the Provisioning Data Manager database number. The number is 0.0.10.2 / 0 . Important: The first four digits of this number (0.0.10.2) must match the database number in the pointer to the Provisioning Data Manager in the CM pin.conf file. See " Connecting the Connection Manager to the Provisioning Data Manager ".
- dm dm_logfile	Specifies the path of the log file.
- dm dm_port	Specifies the port number on which the Provisioning Data Manager listens for connections. The default port number is 11990 . Important: This number must match the port number in the pointer to the Provisioning Data Manager in the CM pin.conf file. For more information, see " Connecting the Connection Manager to the Provisioning Data Manager ".
- dm dm_sm_obj	Specifies a pointer to the shared library that contains the code for the Provisioning Data Manager.
- dm_provision prov_ptr ip	Specifies the host name of the provisioning system. The default port number is 20000 .
- dm_provision connect_retry_interval	Specifies the retry interval in seconds.
- dm_provision connect_retries	Specifies the number of retries.
- dm dm_trans_timeout	Specifies the timeout interval in seconds.

Connecting the Connection Manager to the Provisioning Data Manager

The CM configuration file includes entries that point to the Provisioning Data Manager.

To check or add these entries:

1. Open the CM configuration file (**BRM_home/sys/cm/pin.conf**) and look for the following entries:


```
- cm dm_pointer 0.0.10.2 ip Hostname_of_the_Provisioning_Data_Manager 11990
- fm_prov_telco prov_db 0.0.10.2 / 0
```
2. If the entries do not exist, add them. For more information on the guidelines for database and port-number entries, see *BRM System Administrator's Guide*.
3. Restart the CM.

Creating Network Elements

 **Note**

This is a mandatory configuration task.

By default, there are only sample network elements. Before creating telephone numbers and SIM cards, you need to add the network elements that you will use with your GSM services. Your customized network elements are displayed in SIM Administration Center and Number Administration Center.

You assign network elements to SIM cards and telephone numbers:

- When you create a block of numbers, you specify the network element for all the numbers in the block.
- When you create an order of SIM cards, you specify the network element for all the SIM cards in the order.

For more information, see the following sections:

- [About Managing Telephone Numbers](#)
- [About Managing SIM Card Inventory](#)

Loading GSM Configuration Files

This section describes how to use utilities to load configuration files required for your system.

- [Loading Provisioning](#)
- [Loading Service Order States](#)
- [Loading GSM Provisioning Flag Definitions](#)
- [Loading the Permit Mapping Files](#)

 **Note**

Account-level ERAs and sample provisioning tags are loaded as part of installation.

Loading Provisioning

Load provisioning by running the **load_pin_telco_provisioning** utility:

```
load_pin_telco_provisioning -dv pin_telco_provisioning
load_pin_telco_provisioning -dv pin_telco_provisioning_gsm
```

For information about the utility's syntax and parameters, see "load_pin_telco_provisioning" in *BRM Provisioning Services*.

Loading Service Order States

Load telco and GSM service order states by running the **load_pin_telco_service_order_state** utility:

```
load_pin_telco_service_order_state -dv pin_telco_service_order_state
load_pin_telco_service_order_state -dv pin_telco_service_order_state_gsm
load_pin_telco_service_order_state -dv pin_telco_service_order_state_gsm_data
load_pin_telco_service_order_state -dv pin_telco_service_order_state_gsm_fax
load_pin_telco_service_order_state -dv pin_telco_service_order_state_gsm_sms
load_pin_telco_service_order_state -dv pin_telco_service_order_state_gsm_telephony
```

For information about the utility's syntax and parameters, see "load_pin_telco_service_order_state" in *BRM Provisioning Services*.

Loading GSM Provisioning Flag Definitions

 **Note**

This is a mandatory configuration task.

To add or change provisioning flags in the BRM database, you edit a copy of the **telco_features_and_profiles_states.en_US** sample file in the *BRM_home/sysmsgs/telcofeaturesandprofilesstates* directory. You then use the **load_localized_strings** utility to load the contents of the file into the *Istrings* objects.

Loading the Permit Mapping Files

Load the GSM number and SIM permit mapping files by running the **load_pin_device_permit_map** utility:

```
load_pin_device_permit_map pin_device_permit_map_num_telco_gsm
load_pin_device_permit_map pin_device_permit_map_sim_telco_gsm
```

For more information about the utility's syntax and parameters, see "load_pin_device_permit_map" in *BRM Developer's Guide*.

Configuring Event Notification for GSM Manager

 **Note**

This is a mandatory configuration task.

To enable provisioning, GSM delayed activation, and communication with ECE, BRM uses event notification.

Before you can use GSM Manager, you must configure the event notification feature as follows:

1. If your system has multiple configuration files for event notification, merge them. See "Merging Event Notification Lists" in *BRM Developer's Guide*.

2. Ensure that the merged file includes the entire event notification list in the *BRM_home\sys\data\config\pin_notify_telco* file.
3. (Optional) If necessary to accommodate your business needs, add, modify, or delete entries in your final event notification list. See "Editing the Event Notification List" in *BRM Developer's Guide*.
4. (Optional) If necessary to accommodate your business needs, create custom code for event notification to trigger. See "Triggering Custom Operations" in *BRM Developer's Guide*.
5. Load your final event notification list into the BRM database. See "Loading the Event Notification List" in *BRM Developer's Guide*.

For more information, see "Using Event Notification" in *BRM Developer's Guide*.

Uninstalling GSM Manager and Provisioning Data Manager

To uninstall GSM Manager and Provisioning Data Manager, see "Uninstalling Optional Components" in *BRM Installation Guide*.

Installing GPRS Manager

Learn how to install the Oracle Communications Billing and Revenue Management (BRM) GPRS Manager software. The GPRS Manager package includes both GPRS Manager and a subset of Services Framework Manager, which is installed as a dependent component during installation.

Topics in this document:

- [Installing GPRS Manager](#)
- [Configuring Event Notification for GPRS Manager](#)
- [Uninstalling GPRS Manager](#)

Installing GPRS Manager

Note

If you already installed the product, you must uninstall its features before reinstalling them.

To install GPRS Manager, see "Installing Individual BRM Components" in *BRM Installation Guide*.

Configuring Event Notification for GPRS Manager

Note

This is a mandatory configuration task.

Before you can use GPRS Manager, you must configure the event notification feature as follows:

1. If your system has multiple configuration files for event notification, merge them. See "Merging Event Notification Lists" in *BRM Developer's Guide*.
2. Ensure that the merged file includes the entire event notification list in the **BRM_home\sys\data\config\pin_notify_telco** file.
3. (Optional) If necessary to accommodate your business needs, add, modify, or delete entries in your final event notification list. See "Editing the Event Notification List" in *BRM Developer's Guide*.
4. (Optional) If necessary to accommodate your business needs, create custom code for event notification to trigger. See "Triggering Custom Operations" in *BRM Developer's Guide*.

5. Load your final event notification list into the BRM database. See "Loading the Event Notification List" in *BRM Developer's Guide*.

For more information, see "Using Event Notification" in *BRM Developer's Guide*.

Uninstalling GPRS Manager

To uninstall GPRS Manager, see "Uninstalling Optional Components" in *BRM Installation Guide*.

Part II

Managing Telephone Number Inventory

Part II describes how to manage telephone number inventory in an Oracle Communications Billing and Revenue Management (BRM) system. It contains the following chapters:

- [About Managing Telephone Numbers](#)
- [Installing and Configuring Number Manager and Number Administration Center](#)

About Managing Telephone Numbers

Learn how to manage your telephone number inventory in the Oracle Communications Billing and Revenue Management (BRM) system by using Number Manager and Number Administration Center.

Topics in this document:

- [About Managing Numbers](#)
- [About Using Number Administration Center](#)
- [About Managing Telephone Numbers in Customer Center](#)
- [About Number Device States](#)
- [Creating a Block of Telephone Numbers](#)
- [About Modifying a Block](#)
- [Updating Telephone Numbers](#)
- [About Number Customization Options](#)

Before setting up number management, you need to know the following information:

- Basic BRM concepts. See *BRM Concepts*.
- How to create and edit BRM configuration files. See "Using Configuration Files" in *BRM System Administrator's Guide*.

 **Note**

Number Manager and Number Administration Center are optional components.

About Managing Numbers

To manage telephone numbers, you use the following components:

- *Number Manager* includes the number management opcodes, utilities, configuration files, and storable classes.
- *Number Administration Center* is a GUI application used by operations personnel to manage telephone number inventory.

You use Customer Center to assign and manage numbers in customer accounts. For more information, see information about working with GSM accounts in Customer Center Help.

About Using Number Administration Center

To create an inventory of telephone numbers in the BRM database, you use Number Administration Center to create blocks of numbers. Blocks of numbers usually correspond to geographic regions or business entities. When you create a block of numbers, all the numbers

in the block share the block name and the telephone number attributes, such as the network element (also known as home location register, or HLR). See "[Creating a Block of Telephone Numbers](#)".

About Managing Blocks and Numbers

After creating the numbers as devices in the BRM database, you can do the following:

- Modify a block. You can change the block name and prefix (for example, the North American area code) of all the numbers in the block. See "[About Modifying a Block](#)".
- Change the size of an existing number block. See "[Changing the Size of the Number Block](#)".
- Split a block. You can split a block into smaller blocks, but you cannot join blocks. See "[Splitting a Block](#)".
- Change telephone number attributes for one or more numbers. See "[Updating Telephone Numbers](#)".
- Manage how numbers are quarantined after a customer cancels a service. See "[About Number Device States](#)".

Getting Information about Number Usage

You can get information about number inventory and status by running the Number Manager reports. See "Number Manager Reports" in *BRM Reports*.

In addition, you can display the history of a customer's number in Customer Center. See information about displaying telephone number history in Customer Center Help.

You can also display the number inventory in Number Administration Center. See "[Displaying the Number Inventory](#)".

About Managing Telephone Numbers in Customer Center

To create accounts and manage telephone numbers in existing accounts, use Customer Center. You can perform the following tasks:

- Assign telephone numbers to services.
- Change a customer's telephone number.
- Display the history of a customer's number.

For more information, see information about working with GSM accounts in Customer Center Help.

About Number Device States

Telephone numbers have the following device states: new, assigned, quarantined, and unassigned.

- When you create a number, the device state is *New*.
- When you assign a number to a customer, the device state is *Assigned*.
- When a customer cancels the service associated with the number, the device state is *Quarantined*.

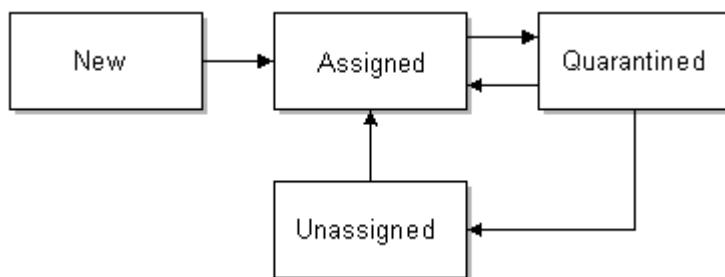
① Note

By default, you can assign a quarantined number to an account. You can modify the PCM_OP_NUM_POL_DEVICE_ASSOCIATE policy opcode to not allow assigning a quarantined number to an account. You can also write a custom opcode and use it in the device state transition from quarantined to assigned.

- By default, a number is unquarantined after 90 days. After a number has been unquarantined, the device state changes to *Unassigned*.

[Figure 4-1](#) shows how number device state changes:

Figure 4-1 Number Device State Changes



You cannot use Number Administrator to change the number device state. Instead, the number device state is changed automatically when the number is assigned to an account and when an account or service changes status (for example, when a customer cancels the service that the number is associated with).

① Note

You can create your own number device states. See "Defining the Device Life Cycle" in *BRM Developer's Guide*.

Taking Numbers Out of Quarantine

By default, a number is quarantined for 90 days. You can change the number of days by running the "[load_pin_num_config](#)" utility. See "[Changing the Default Quarantine Period](#)".

If you run out of new or unassigned numbers, you can use the "[pin_change_num_quarantine](#)" utility to unquarantine numbers. See "[Changing the Quarantine Status Manually](#)".

Displaying the Number Inventory

By default, Number Administration Center displays the number inventory when you start the application. To display it at any time, choose **View - Number Inventory**.

You can also display the number status of the numbers in a single block. See "[About Modifying a Block](#)".

Creating a Block of Telephone Numbers

Note

Before creating numbers, you must define network elements and any other number properties required by your system (for example, number categories). See "[Mandatory Configuration Tasks](#)".

To create a block of numbers in the BRM database, start Number Administration Center and choose **File - New Block**.

[Figure 4-2](#) shows the block creation screen in Number Administration Center:

Figure 4-2 Block Creation in Number Administration Center

The screenshot shows a 'New Block' dialog box with the following fields:

Start number:	<input type="text"/>
End number:	<input type="text"/>
Block name:	<input type="text"/>
Service type:	<input type="text"/>
Number category:	None
Network element:	Portal
Brand:	Brand Host

To create a block of numbers, you enter the following information:

- A start number and an end number that together form the range of numbers in the block. You can use any common format when entering numbers. For example:
 - 1-800-555-1212
 - 1(800) 555-1212
 - 1.800.555.1212
 - 18005551212
 - 444-111-2345
 - 444.111.2345
 - 444.1112345
 - 444 111 2345
- A block name. This can be any name you want, but you can only use the same name for one block. The maximum length is 255 characters. You can change the block name at any time.

In addition to the block name and the range of numbers in the block, you assign the following number attributes to all the numbers in the block:

- The *network element*. You can change the network element after you create the block. To specify the network element, you choose from a list. By default, there are no network elements; you must add your own. You can also localize network elements. See "[Creating Network Elements](#)".
- The *number category*. This is a configurable field that you can use for managing numbers. It is typically used for identifying corporate entities, but it can be used for defining any type of number classification. For example, you can create number categories for business or residential numbers. You can also localize number categories. You can change the number category after you create the block. To customize the list of number categories, see "[Creating Number Categories](#)". By default, the number categories are *None* and *Reserved*.
- The *vanity type*. This attribute can be used for customized pricing and rating. Number Manager includes only sample vanity types; if you use vanity types, you must create your own. You can localize vanity types. See "[Creating Vanity Types](#)". When you create a block of numbers, you cannot choose the vanity type for the entire block. By default, the vanity type applied to a new block is the first vanity type defined in the `num_vanities.locale` file. After creating the numbers, you can apply vanity types to numbers.
- The *service type*. You can use the service type to customize business policies. By default, the service type is not applied to any business logic. See "[Customizing How Service Types Are Used](#)".

About Customizing Block Creation and Modification

By default, BRM prevents the creation of duplicate numbers. To customize how numbers are created (for example, to prevent numbers or ranges of numbers from being created), edit the `PCM_OP_NUM_POL_DEVICE_CREATE` source code.

You can change the prefix (for example, the North American area code) for a block of numbers. To allow changing of other parts of the number (for example, country codes), modify the `PCM_OP_NUM_POL_DEVICE_SET_ATTR` source code.

See *BRM Opcode Guide* for more information.

About Modifying a Block

To modify a block:

1. In Number Administration Center, choose **File - Open a Block**.
2. Enter the search criteria. You can search for block name and telephone number. You can use wildcards when searching for numbers:
 - Use the asterisk (*) to match zero or more characters.
 - Use the question mark (?) to match a single character.
3. Select the block and click **Open**.
4. Change the block name or prefix. By default, you can change only the North American area code digits.
5. Choose **File - Commit to Database**.

Changing the Size of the Number Block

You can extend the block size by specifying a new higher value for the end number. You can shrink the range of an existing block of numbers by doing either of the following:

- Specifying a new lower value for the end number.
- Specifying a new higher value for the start number.

To change the size of the number block:

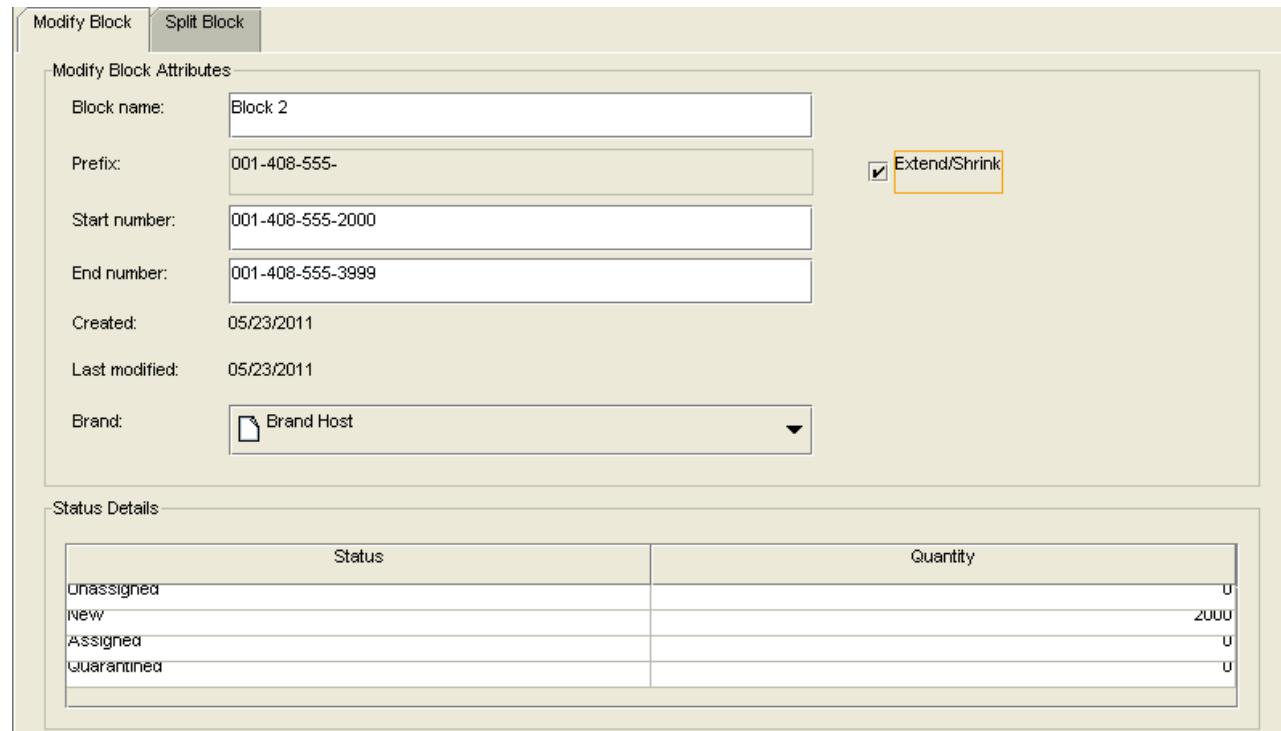
1. In Number Administration Center, choose **File - Open a Block**.
2. Enter the search criteria.

You can search for block name and telephone number. You can use wildcards when searching for numbers:

- Use the asterisk (*) to match zero or more characters.
- Use the question mark (?) to match a single character.

3. Select the block and click **Open**.
4. In the **Modify Block** tab, select the **Extend/Shrink** check box as shown in [Figure 4-3](#).

Figure 4-3 Extend/Shrink Check Box



The screenshot shows the 'Modify Block' page with the 'Extend/Shrink' checkbox selected. The 'Extend/Shrink' checkbox is located in the 'Modify Block Attributes' section, which also includes fields for 'Block name', 'Prefix', 'Start number', 'End number', 'Created', 'Last modified', and 'Brand'. Below this section is a 'Status Details' table with four rows: 'Unassigned', 'New', 'Assigned', and 'Quarantined'. The 'New' row has a quantity of 2000.

Status	Quantity
Unassigned	0
New	2000
Assigned	0
Quarantined	0

5. Do one of the following:
 - To extend the number block, in the **End number** field, enter a value that is higher than the previous end number.

Note

The new end number that you specify must be in a continuous range and not overlap with any other existing blocks.

- To shrink the number block, in the **End Number** field, enter a value that is less than the previous end number. Alternatively, in the **Start Number** field, enter a value that is higher than the previous start number.

6. Choose **File - Commit to Database**.

Splitting a Block

You can split a block into two or more blocks. After you split a block, you can modify the blocks (for example, change the North American area code) of the numbers in the blocks.

1. In Number Administration Center, choose **File - Open the block**.
2. Click the **Split Block** tab.

By default, two rows are displayed in a table as shown in [Figure 4-4](#):

Figure 4-4 Splitting a Block

Block Name	Quantity	Start Number	End Number
Block 2	999	0014085552000	0014085552998
Block 2	1	0014085552999	0014085552999

You can use two methods for splitting a block:

- Enter a quantity in either of the rows. It must be smaller than the current quantity of the block.
- Enter an end number. It must be a number within the current block range. You cannot enter a start number.

In either case, the quantity and start and end numbers are updated to match the new block size.

3. Enter names for the new blocks.
4. Click **Add Block**.
5. Click **Commit**.

Updating Telephone Numbers

You can use Number Administration Center to search for telephone numbers and modify them. You can update all number attributes except the status and the number itself. For information about number attributes, see "[Creating a Block of Telephone Numbers](#)".

Note

- You can update numbers by updating a block of numbers. For example, you can change the prefix (by default, the North American area code) for a block of numbers.
- Changing a telephone number or changing any of its attributes does not create a new **/device/number** object in the database. Instead, the existing number is updated.
- It is possible to delete numbers by using opcodes, but it is not recommended. You could reduce data integrity, or accidentally delete a number that is in use.

To update a number:

1. In Number Administration Center, choose **File - Search Numbers**.

The Search window opens.

2. Enter the search criteria.

You can search on any number attribute. You can use wildcards when searching for numbers and number categories:

- Use the asterisk (*) to match zero or more characters.
- Use the question mark (?) to match a single character.
- Number Administration Center ignores all other punctuation. For example, searching for ***8-5*** is the same as searching for ***85***.

[Figure 4-5](#) shows the Search Numbers dialog after a search:

Figure 4-5 Search Numbers Dialog Box

The screenshot shows the 'Search Numbers' dialog box with the following search parameters:

- Number: *1408*
- Block name: (empty)
- Numbers per page: 50
- Service type: (empty)
- Brand: Brand A
- Number category: <Not specified>
- Network element: <Not specified>
- Status: New
- Vanity type: <Not specified>
- Include sub-brands

The results table displays 1000 results (as indicated in the status bar) across 100 pages. The first few rows of the table are:

Number	Block Name	Category	Network Element	Service Type	Status	Vanity Type	Brand
001-408-555-1000	California	None	sample_network		New	Gold	Brand A
001-408-555-1001	California	None	sample_network		New	None	Brand A
001-408-555-1002	California	None	sample_network		New	None	Brand A
001-408-555-1003	California	None	sample_network		New	None	Brand A
001-408-555-1004	California	None	sample_network		New	None	Brand A
001-408-555-1005	California	None	sample_network		New	None	Brand A
001-408-555-1006	California	None	sample_network		New	None	Brand A
001-408-555-1007	California	None	sample_network		New	None	Brand A

Buttons at the bottom include: ? (Help), Previous Page, Next Page, Open, and Open All. The status bar at the bottom left says 'Ready - Results: 1009'.

3. Open the numbers that you want to modify.

- To open all numbers found in the search, click **Open All**.
- To open the page of numbers that is currently displayed, click **Open**. Click **Next Page** and **Previous Page** to display different numbers.
- Select one or more numbers and click **Open**.
- Choose **File - Search Numbers** to return to the search results.

To select numbers:

- To select one number, click it.
- To select all numbers, click **Select All**.
- To select a range of numbers, press **SHIFT** and click the first and last numbers in the range.
- To select multiple numbers not in a range, press **CTRL** and click the numbers.

Note

If the quantity of numbers found is the same or less than the quantity of numbers per page, the **Open** and **Open All** buttons open the same set of numbers. For example, if you specify 100 numbers per page, and only 50 numbers are found, both buttons open 50 numbers.

[Figure 4-6](#) shows the Modify Number Parameters dialog box:

Figure 4-6 Modify Number Parameters Dialog Box

The screenshot shows the 'Modify Number Attributes' dialog box. At the top, there are four dropdown menus: 'Number category', 'Network element', 'Service type', and 'Vanity type'. Below these is a table with the following data:

Number	Block Name	Category	Network Element	Service Type	Status	Vanity Type	Brand
001-408-555-1000	California	None	sample_network		New	Gold	Brand A
001-408-555-1001	California	None	sample_network		New	None	Brand A
001-408-555-1002	California	None	sample_network		New	None	Brand A
001-408-555-1003	California	None	sample_network		New	None	Brand A
001-408-555-1004	California	None	sample_network		New	None	Brand A
001-408-555-1005	California	None	sample_network		New	None	Brand A
001-408-555-1006	California	None	sample_network		New	None	Brand A
001-408-555-1007	California	None	sample_network		New	None	Brand A

At the bottom right of the dialog box is a 'Select All' button.

4. Select the numbers that you want to modify.
5. Enter the new number attributes.
6. Click **Commit**.

About Number Customization Options

See *BRM Opcode Guide* for information about customize number management. You can customize how you manage number inventory as follows:

- You can customize network elements.

 **Note**

By default, there are only sample network elements; you must add your own. See "[Creating Network Elements](#)".

- You can customize number device and service associations.
- You can customize number device states.
- You can customize how numbers are quarantined.
- You can create number categories.
- You can create vanity types.
- You can customize how service types are used.
- You can customize how Number Administration Center works.

Installing and Configuring Number Manager and Number Administration Center

Learn how to install and configure Oracle Communications Billing and Revenue Management (BRM) Number Manager and Number Administration Center.

Topics in this document:

- [Mandatory Configuration Tasks](#)
- [Installing Number Manager](#)
- [Installing Number Administration Center](#)
- [Configuring Event Notification for Number Manager](#)
- [Customizing Number Quarantine](#)
- [Creating Network Elements](#)
- [Creating Number Categories](#)
- [Creating Vanity Types](#)
- [Customizing How Service Types Are Used](#)
- [Changing the Number Administration Center Number Display Format](#)
- [Configuring Search Performance in Number Administration Center](#)
- [Uninstalling Number Manager](#)

For information about managing number inventory, see "[About Managing Telephone Numbers](#)".

Mandatory Configuration Tasks

In addition to installing the Number Manager software, you must perform the following configuration tasks before you can use Number Manager and Number Administration Center:

 **Note**

You can perform these tasks in any order, but you must install Number Manager first.

- Define and load network elements. (If SIM Manager is already installed, this might already be done.) See "[Creating Network Elements](#)".
- Set the default number of quarantine period; for example, 90 days. See "[Changing the Default Quarantine Period](#)".
- Define number device states. This requires two procedures: defining and loading the state transition model, and loading the localized state descriptions.
- Define number categories. See "[Creating Number Categories](#)".

- Define vanity number types. Even if you do not use vanity numbers, you must load the `num_vanities.locale` file. See "[Creating Vanity Types](#)".

Installing Number Manager

Note

If you already installed the product, you must uninstall its features before reinstalling them.

To install Number Manager, see "Installing Individual BRM Components" in *BRM Installation Guide*.

Note

You should verify that the Number Manager FMs were added to the Connection Manager (CM) configuration file.

Installing Number Administration Center

To install Number Administration Center, see "Installing Individual BRM Clients" in *BRM Installation Guide*.

Configuring Event Notification for Number Manager

Note

- This is a mandatory configuration task.
- If you already configured event notification for SIM Manager, skip this procedure.

To use Number Manager, you must configure the event notification feature as follows:

1. If your system has multiple configuration files for event notification, merge them. See "Merging Event Notification Lists" in *BRM Developer's Guide*.
2. Ensure that the merged file includes the entire event notification list in the `BRM_home/sys/data/config/pin_notify` file.
3. (Optional) If necessary to accommodate your business needs, add, modify, or delete entries in your final event notification list. See "Editing the Event Notification List" in *BRM Developer's Guide*.
4. (Optional) If necessary to accommodate your business needs, create custom code for event notification to trigger. See "Triggering Custom Operations" in *BRM Developer's Guide*.
5. Load your final event notification list into the BRM database. See "Loading the Event Notification List" in *BRM Developer's Guide*.

For more information, see "Using Event Notification" in *BRM Developer's Guide*.

Customizing Number Quarantine

You can customize how numbers are quarantined in the following ways:

- You can change the default quarantine period. See "[Changing the Default Quarantine Period](#)".
- You can manually unquarantine numbers. "[Changing the Quarantine Status Manually](#)".
- You can customize policy source code to change how number quarantine works. "[Changing How Number Quarantine Works](#)".

For more information, see "[About Managing Telephone Numbers](#)".

Changing the Default Quarantine Period

Note

This is a mandatory configuration task.

By default, the quarantine period is 90 days. You can change the default quarantine period.

To customize the default quarantine period, you run the **load_pin_num_config** utility to load the contents of the **pin_num_config** file into a **/config/num** object in the BRM database.

Note

The utility needs a configuration file in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in *BRM System Administrator's Guide*.

1. Edit the **pin_num_config** file in *BRM_home/sys/data/config*. The default entry is:

```
quarantine_period 90
```

To change the quarantine period, change the number. For example:

```
quarantine_period 45
```

2. Save the file.

3. Use the following command to load the **pin_num_config** file:

```
load_pin_num_config pin_num_config
```

For more information, see "[load_pin_num_config](#)".

To verify that the new value was loaded, you can display the **/config/num** object by using the Object Browser, or use the **robj** command with the **testsnap** utility. (See "Reading an Object and Writing Its Contents to a File" in *BRM Developer's Guide*.) The **/config/num** object shows the quarantine period in seconds in the **PIN_FLD_VALUE** field. In this example, 7776000 seconds = 90 days.

```
0 PIN_FLD_VALUE     STR [0] "7776000"
```

Changing the Quarantine Status Manually

If you run out of new or unassigned numbers, use the **pin_change_num_quarantine** utility to make numbers available for unquarantine. When the numbers are available for unquarantine, you run the **pin_deferred_act** utility to change the status to unquarantined. Therefore, you first run the **pin_change_num_quarantine** utility, then you run the **pin_deferred_act** utility.

Note

Both utilities need a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

When you use the **pin_change_num_quarantine** utility, you have the following options:

- You can use the **-l** parameter to display how many numbers will be unquarantined without committing anything to the database.
- When you make numbers available for unquarantine, remember that there might already be numbers that are available for unquarantine, and they will be unquarantined when you run the **pin_deferred_act** utility. You can use the **-a** parameter to take into account all numbers, even those that are ready for unquarantine.

For example, you might want to unquarantine 50 numbers, but you have 30 numbers ready for unquarantine already.

- If you use the **-a** parameter, the utility affects only 20 numbers because 30 are ready to be unquarantined already. Your total quantity is 50.
- If you do not use the **-a** parameter, the utility unquarantines 50 numbers in addition to the 30 that are already available for unquarantine. Your total quantity is 80.

For more information, see "[Examples](#)".

1. Use the following command to run the **pin_change_num_quarantine** utility:

```
pin_change_num_quarantine quantityOfNumbers
```

For example, to unquarantine 100 numbers, enter:

```
pin_change_num_quarantine 100
```

For more information, see "[pin_change_num_quarantine](#)".

Note

If there are fewer numbers available, the utility displays a warning.

2. Use the following command to run the **pin_deferred_act** utility:

```
pin_deferred_act
```

There are no input parameters for this utility. For more information, see *BRM Configuring and Running Billing*.

Changing How Number Quarantine Works

By default, you can assign a quarantined number to an account. You can modify the PCM_OP_NUM_POL_DEVICE_ASSOCIATE policy opcode to not allow assigning a quarantined number to an account.

In addition, you can create your own opcode to handle number quarantine. By default, the PCM_OP_NUM_QUARANTINE opcode controls how numbers are quarantined. To customize how number quarantine works, you can edit and load the **pin_device_state_num** file to call a custom opcode instead of calling PCM_OP_NUM_QUARANTINE. See "Defining the Device Life Cycle" in *BRM Developer's Guide*.

Creating Network Elements

Note

This is a mandatory configuration task.

Note

If you already created network elements for SIM cards, you do not need to create them again. The same network elements are used for numbers and SIM cards.

By default, there are only sample network elements. Before creating numbers, you need to add the network elements that you will use with your GSM services.

To customize network elements, you edit the **pin_network_elements** file, then run the [**"load_pin_network_elements"**](#) utility to load the contents of the file into a **/config/network_element** object in the BRM database.

Note

The utility needs a configuration file in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in *BRM System Administrator's Guide*.

1. Edit the **pin_network_elements** file in *BRM_home/sys/data/config*. The **pin_network_elements** file includes examples and instructions.

The file includes a list of network elements. For example:

```
sample_network_element_1
sample_network_element_2
```

Note

The **load_pin_network_elements** utility overwrites existing network elements. If you are updating network elements, you cannot load new network elements only. You must load complete sets of network elements each time you run the **load_pin_network_elements** utility.

2. Save the **pin_network_elements** file.
3. Use the following command to load the **pin_network_elements** file:

```
load_pin_network_elements pin_network_elements
```

To verify that the network elements were loaded, you can display the **/config/network_element** object by using the Object Browser, or use the **robj** command with the **testnap** utility. (See "Reading an Object and Writing Its Contents to a File" in *BRM Developer's Guide*.) This example shows a network element in the **/config/network_element** object:

```
PIN_FLD_NETWORK_ELEMENT  STR  [ 0 ]  "Sample_Network_Element"
```

Creating Number Categories

 **Note**

This is a mandatory configuration task.

You can customize the list of number categories displayed in Number Administration Center. For information about number categories, see "[Creating a Block of Telephone Numbers](#)".

The default number category is **None**. The only other default option is **Reserved**. To customize number categories, you edit the **num_categories.en_US** sample file in the *BRM_home/sysmsgs/numcategories* directory. You then use the **load_localized_strings** utility to load the contents of the file into the **/strings** objects. See "load_localized_strings" in *BRM Developer's Guide*.

Creating Vanity Types

 **Note**

This is a mandatory configuration task.

You can customize the list of vanity types displayed in Number Administration Center. For information about vanity types, see "[Creating a Block of Telephone Numbers](#)".

To create vanity types, you edit the **num_vanities.en_US** sample file in the *BRM_home/sysmsgs/numvanities* directory. You then use the **load_localized_strings** utility to load the contents of the file into the **/strings** objects. See "load_localized_strings" in *BRM Developer's Guide*.

 **Note**

You can customize the **PCM_OP_NUM_POL_DEVICE_CREATE** policy opcode to apply vanity types automatically. For example, you could automatically apply a vanity type to numbers ending in 2000.

Customizing How Service Types Are Used

By default, values entered in the **Service type** box do not have any business logic applied to them. You can customize the Number Manager business policies to use the value entered in the **Service type** box. For example, you might create service types for voice-only numbers or data-only numbers.

You can customize the following Number Manager policy opcodes to add business logic to service types:

- PCM_OP_NUM_POL_DEVICE_ASSOCIATE
- PCM_OP_NUM_POL_DEVICE_CREATE
- PCM_OP_NUM_POL_DEVICE_SET_ATTR

Changing the Number Administration Center Number Display Format

To change the Number Administration Center number display format, edit the Number Format entry in the **NumAdmin.properties** file. The Number Format entry specifies the default Java class that defines the Number Administration Center number display. To use a different number format, create a Java class and specify it in the Number Format entry.

The default mask is:

xxx-xxx-xxx-xxxx

How BRM Chooses a Mask

When you specify several masks, BRM chooses the mask that best matches the input number. For example, if the input number is:

044.112.233.3444

and the list of masks is:

(091)xx-xx-xxx-xxx

(044)xx.xx.xxxx.xxxx

(212.xxxx.xxxx)

the mask used is:

(044)xx.xx.xxxx.xxxx

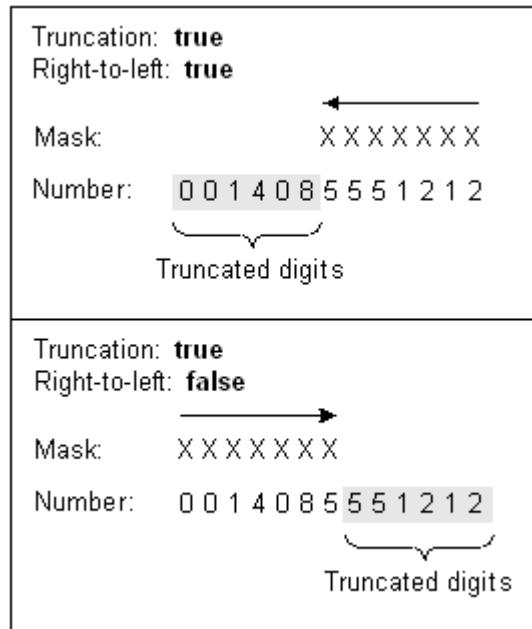
About Truncating Numbers by Using Masks

When setting the number format, you can truncate numbers. For example, if a number is entered as 001(408)5551212, you can truncate it to display as 5551212. Numbers are truncated by default.

When you truncate numbers, you can specify where to truncate: the left side of the number or the right side. By default, numbers are read from right to left, so the left side of the number is truncated.

[Figure 5-1](#) shows the difference between truncating right to left and left to right. (A value of **true** truncates right to left.)

Figure 5-1 Truncating Numbers Differences



[Table 5-1](#) shows examples of how numbers are formatted based on the mask and truncation. A truncate value of **true** means right-to-left.

Table 5-1 Number Truncation Examples

Input	Mask	Truncated	Truncate direction	Result
1(408)5179011	xxx.xxx.xxxx	true	true (<<<)	408.517.9011
1(408)5179011	xxx.xxx.xxxx	false	true (<<<)	1408.517.9011
234-5678	x-xxx-xxx-xxxx	false	true (<<<)	234-5678
234-5678	x-xxx-xxx-xxxx	false	false (>>>)	2-345-678
001(408)517901100	001-xxx-xxx-xxxx	true	false (>>>)	001-408-517-9011
001(408)517901100	001-xxx-xxx-xxxx	false	false (>>>)	001-408-517-901100
1(408)517-901100	x.xxx.xxx.xxxx	true	true (<<<)	0.851.790.1100
1(408)517-901100	x.xxx.xxx.xxxx	false	true (<<<)	140.851.790.1100

Changing the Number Display Format

To change the number display format:

1. Open the **NumAdmin.properties** file in the **C:\Program Files\Common Files\Portal Software** directory.
2. Edit the **mask** entries as needed. The default mask looks like this:

```
device.num.formatter.mask.default=xxx-xxx-xxx-xxxx
```

3. (Optional) To add a new mask, add the mask name to the **device.num.formatter.masks** entry.

This example shows a new mask for Denmark:

```
device.num.formatter.masks=default france uk denmark
```

This example shows the mask definition:

```
device.num.formatter.mask.denmark=45-xxxx-xxxxx
```

4. (Optional) If a number is larger than the mask, you can specify whether to truncate the number and how to truncate it:

- To truncate numbers that are longer than the mask, edit the **device.num.formatter.truncate** entry. Enter **true** to truncate numbers, enter **false** to not truncate numbers.

By default, numbers are not truncated:

```
device.num.formatter.truncate=false
```

- To specify how to truncate the number, edit the **device.num.formatter.direction** entry. By default, numbers are read from right to left, and truncation occurs at the left side of the number.

For example, if the number is 0014085551212 and the mask is xxx-xxxx, the truncated number is 5551212.

Enter **true** to truncate right-to-left, enter **false** to truncate left-to-right:

```
device.num.formatter.direction=true
```

5. Save and close the file.
6. Restart Number Administration Center.

Creating a Custom Number Format Class

You can create and use a custom number format Java class. You might want to do this if defining custom masks is not sufficient.

1. Create the class.
2. Open the **NumAdmin.properties** file in the **C:\Program Files\Common Files\Portal Software** directory.
3. Edit the **device.num.formatter.class** entry:

```
device.num.formatter.class=custom_Class
```

where *custom_Class* is your class.

4. Edit the **device.num.formatter.method** entry to use the method defined in your class. See "[Changing the Number Display Format](#)".

```
device.num.formatter.method=format
```

5. Edit the **device.num.formatter.masks** entry and add new masks, as described in "[Changing the Number Administration Center Number Display Format](#)".

6. Save and close the file.
7. Restart Number Administration Center.

Configuring Search Performance in Number Administration Center

You can edit the **NumAdmin.properties** file to change how Number Administration Center searches for blocks and numbers.

1. Open the **NumAdmin.properties** file in the **C:\Program Files\Common Files\Portal Software** directory.
2. To set the default quantity of numbers displayed in a page of search results, edit the **pageSize** entry. For example:

```
pageSize=50
```

You can change the page size at any time when using Number Administration Center.

3. To change the search performance, edit the **searchSize** entry. For example:

```
searchSize=100
```

If you use a low-bandwidth system with a lot of numbers, reducing the step size can speed up searches.

4. Save and close the file.
5. Restart Number Administration Center.

Uninstalling Number Manager

To uninstall Number Manager, see "Uninstalling Optional Components" in *BRM Installation Guide*.

Part III

Managing SIM Card Inventory

Part III describes how to manage SIM card inventory in an Oracle Communications Billing and Revenue Management (BRM) system. It contains the following chapters:

- [About Managing SIM Card Inventory](#)
- [Installing and Configuring SIM Manager and SIM Administration Center](#)

About Managing SIM Card Inventory

Learn how to manage SIM card inventory in your Oracle Communications Billing and Revenue Management (BRM) system by using SIM Manager and SIM Administration Center.

Topics in this document:

- [About Managing SIM Cards](#)
- [About Creating SIM Cards](#)
- [About Associating a SIM Card with a Customer's Service](#)
- [About SIM Card Device States](#)
- [Changing the Network Element of SIM Cards](#)
- [About SIM Card Customization Options](#)
- [Getting Information About SIM Card Usage](#)

 **Note**

SIM Manager and SIM Administration Center are optional components, not part of base BRM.

Before using SIM Manager and SIM Administration Center, you need to know the following information:

- Basic BRM concepts. See *BRM Concepts*.
- BRM system architecture. See "BRM System Architecture" in *BRM Concepts*.
- How to create and edit BRM configuration files. See "Using Configuration Files to Connect and Configure Components" in *BRM System Administrator's Guide*.

About Managing SIM Cards

To manage SIM cards, you use the following components:

- SIM Manager includes the opcodes and storables required for creating and managing SIM card devices in the BRM database.
- SIM Administration Center is a GUI application that you use to create and process SIM card orders, and to change SIM card network element.

You use Customer Center to assign and manage SIM cards in customer accounts. For more information, see information about working with GSM accounts in Customer Center Help.

About Creating SIM Cards

Use SIM Administration Center to order SIM cards from a vendor. For complete instructions on using SIM Administration Center, see the SIM Administration Center Help.

To order SIM cards:

1. Create an order in SIM Administration Center.

The order is stored in the BRM database and can be updated or canceled.

2. Create one or more *vendor request files*. This file specifies the quantity of SIM cards, the starting SIM card and IMSI numbers, and the SIM card attributes, such as the carrier name and SIM card format. This text file follows the standard GEMPLUS file format for SIM card orders.

Vendor request files cannot specify more than 5000 SIM cards. If you create an order for more than 5,000 cards, the order is automatically divided into multiple files. For example, if you create an order for 50,000 SIM cards, SIM Administration Center creates 10 vendor request files of 5,000 SIM cards each. (See "[About SIM Card Request and Response Files](#)".)

You can create a maximum order of 499,995,000 SIM cards, which would create 99,999 vendor request files.

You can change the default values for maximum quantity of SIM cards in an order and maximum number of SIM cards in a request file. See "[Customizing SIM Administration Center](#)".

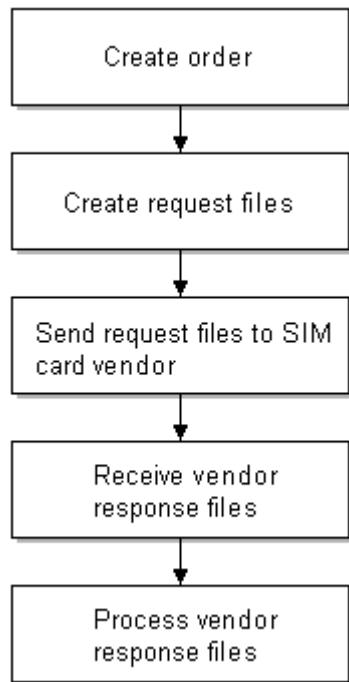
3. Send the vendor request files to the vendor. You can use any file transfer method, such as FTP or email.
4. Receive the *vendor response files* and process them in SIM Administration Center. This creates a **/device/sim** object for each SIM card.

 **Note**

Multiple vendor *request files* are created automatically, based on the size of the order. However, you need to process each vendor *response file* individually.

You can pre-provision SIM cards when you process the vendor response files. When you pre-provision a SIM card, the SIM card is assigned a SIM card number and an IMSI number on the network. For more information, see "[About SIM Card Pre-Provisioning](#)".

[Figure 6-1](#) shows the process for creating SIM cards:

Figure 6-1 SIM Card Creation Process

About SIM Card Orders

① Note

Contact your SIM card vendor for information about the SIM card attributes. For example, you need to define your graphic elements (the logo and graphics that are printed on the card).

When you create an order, you can specify the following order attributes:

- Customer name and address information.

The name and address information for the vendor who creates the SIM cards.

① Note

There are limitations on how many characters you can use for each piece of customer or vendor information. However, these limitations allow for more characters than you will probably use. For more information, see the **order** object definition.

- The quantity of SIM cards in the order. By default, the maximum is 499,995,000 SIM cards.
- Starting SIM card number and IMSI.
- SIM card attributes:
 - SIM card format, for example, ISO or Plug-in
 - Carrier name

- Transport key
- Electrical profile
- Graphical reference, for example, a company logo
- SIM card number prefix
- IMSI prefix
- Network element

[Figure 6-2](#) shows the **Order Details** tab used for specifying SIM card attributes in SIM Administration Center:

Figure 6-2 Order Details Tab

Quantity:	10000	Start SIM:	10000
SIM prefix:	50000	Start IMSI:	60000
IMSI prefix:	20000	Graphical profile:	01.00
SIM card format:	Plug-in	Network element:	Portal
Transport key:	255	Carrier name: Carrier 11	
Electrical profile:	01.00	Brand: Brand A	

About Managing Orders

When you create and manage SIM card orders, BRM assigns a status to the order. SIM Administration Center displays the order status.

1. When you create an order, the status is *New*. You can save a new order, or cancel it. When you cancel an order, the order status is set to *Canceled*. You can also modify an order if it is in the *New* status. You cannot delete an order.
2. After you create a request file, the order status is set to *Requested*. When an order has the *Requested* status, you cannot cancel or modify it.
3. When you begin processing vendor response files, the order status is set to *Partially Received*.
4. After you process all the vendor response files in an order, the order status is set to *Received*.

When the order has a Received or Partially Received status, you cannot change the order attributes. However, after you process the vendor response file, you can change the SIM card and network element.

You can search for orders of any status. [Figure 6-3](#) shows the attributes you use when searching for an order:

Figure 6-3 Order Search Attributes

The dialog box is titled "Order Parameters". It contains the following fields:

- Order Number: Text input field
- SIM number: Text input field
- IMSI number: Text input field
- Carrier name: Text input field
- Date created: Text input field (with placeholder "(mm/dd/yyyy)")
- Date processed: Text input field (with placeholder "(mm/dd/yyyy)")
- Status: Drop-down menu showing "<Not specified>"
- Brand: Drop-down menu showing "Brand Host"
- Include sub-brands: Check box

At the bottom are four buttons: Reset, Stop Search, Search (highlighted in blue), and Close.

① Note

SIM card orders are stored in the database as **/order/sim** objects.

About SIM Card Request and Response Files

You manage orders with your SIM card vendor by exchanging request and response files. The request files include information about the order, and are used by the vendor to create the SIM cards. The response files contain information from the vendor about the SIM cards.

SIM Administration Center automatically generates request files using the following naming convention:

Customer_abbreviationXXXXX.inp

- The customer abbreviation is the first three letters from the name in the **Customer name** field when creating the order.
- The customer abbreviation is followed by five digits, incremented for each file in the order. The first file in an order uses 00001.

For example, if the customer's name is "Oracle," the first file in an order is named **Ora00001.inp**.

① Note

All orders that you create for the same customer use the same file names. Therefore, when creating request files, you should always put them in a new or empty folder for each order, so files from previous orders are not overwritten.

The vendor response files use the following naming convention:

Customer_abbreviationXXXX.out

- The customer abbreviation is the first three letters from the name in the **Customer name** field when creating the order.
- The customer abbreviation is followed by five digits, incremented for each order. The first file in an order uses 00001.

For example, if the customer name is "Oracle," the first file in an order is named **Ora00001.out**.

About SIM Card Pre-Provisioning

When you process vendor response files, you can also pre-provision SIM cards. This allows a customer to perform tasks such as calling a service center with their new phone.

① Note

- If you pre-provision SIM cards, you must define the pre-provisioning service. See "[Specifying the Pre-Provisioning Service](#)".
- If you do not pre-provision SIM cards, you must change the device state transition. See "[Customizing How to Pre-Provision SIM Cards](#)".

When a SIM card is pre-provisioned, the following actions occur:

1. BRM sends information about each SIM card to the Provisioning Data Manager (**dm_prov_telco**). This information includes the IMSI, SIM card number, an encrypted KI value, the network element, and a dummy MSISDN number.

① Note

The dummy MSISDN number is a placeholder. You can change the MSISDN number if you customize pre-provisioning. See "[Customizing How to Pre-Provision SIM Cards](#)".

2. The Wireless Provisioning DM sends the provisioning request to the provisioning system, which communicates with the network to provision the SIM card.
3. BRM updates the SIM card device state to one of the following:
 - If provisioning succeeds, the SIM card device state is set to *Released*. A CSR can assign this SIM card to a customer. Because the customer does not need to know about the pre-provisioning service, the SIM card device in the BRM database is disassociated from the pre-provisioning service object.
 - If provisioning fails, the SIM card device state is set to *Failed Provisioning*.

Enabling SIM Card Pre-Provisioning

To enable SIM card pre-provisioning:

1. Configure PCM_OP_SIM_DEVICE_PROVISION (opcode number 2403) in state transition **0** (Raw) -> **1** (New) by editing the **pin_device_state_sim** file and loading it with the **load_pin_device_state** utility.

This example shows how to enable pre-provisioning:

```
New -> Assigned or Provisioning
1: 1: 1: 1: 2403: 0
2: 0: 0
5: 0: 0
3: 0: 0
```

2. Configure the following information in the *BRM_home/sys/data/config/pin_notify* file (if not already configured) for state transition and load it with **load_pin_notify**.

```
2706      0      /event/notification/device/state
```

3. In the *BRM_home/sys/data/config/pin_telco_provisioning* configuration file, specify the pre-provisioning information for each device type using the following syntax. A separate line is required for each field, even for the same device type.

Device provisioning info:

```
DeviceType, DeviceState, ProvAction, Field1, "String1",
DeviceType, DeviceState, ProvAction, Field2, "String2"
```

where:

- *DeviceType* specifies the type of device to be pre-provisioned.
- *DeviceState* specifies the device state that triggers the generation of a service order. If you do not specify the **DeviceState** value, service orders are generated for all device state transitions.
- *ProvAction* specifies the provisioning action for the device objects in the service order. In this case, you always use the value **P**, for provisioning.
- *FieldX* specifies the name of a field that contains a device attribute to include in the service order. Field names are replaced by actual values when you run **pin_telco_provisioning**.
- *StringX* specifies a string that is used in the service order to identify the attribute specified by the corresponding Field value. You can query for the string in the service order.

For example, the following lines add the authentication key, IMSI, and device ID values to the service order when a SIM card is pre-provisioned.

```
/device/sim, 3, P,PIN_FLD_DEVICE_SIM.PIN_FLD_KI "KI",
/device/sim, 3,P,PIN_FLD_DEVICE_SIM.PIN_FLD_IMSI "IMSI",
/device/sim, 3,P,PIN_FLD_DEVICE_SIM.PIN_FLD_DEVICE_ID "SIM"
```

4. Load the file into the provisioning configuration object by using the **load_pin_telco_provisioning** utility.

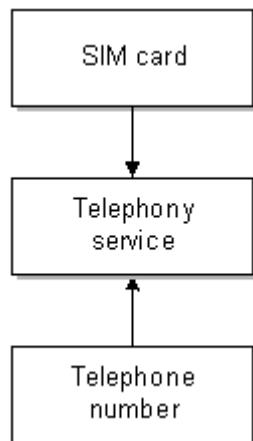
About Associating a SIM Card with a Customer's Service

When an account is created with a GSM service, or a SIM card is added to an existing account, the SIM card is provisioned, and associated with a service. The SIM card device state is set to *Assigned*.

Use Customer Center to associate SIM cards and telephone numbers with telco services and to change the SIM card and number for a customer's service.

SIM cards and telephone numbers are not associated with each other. Instead, you associate SIM cards and telephone numbers with the appropriate service, as shown in [Figure 6-4](#):

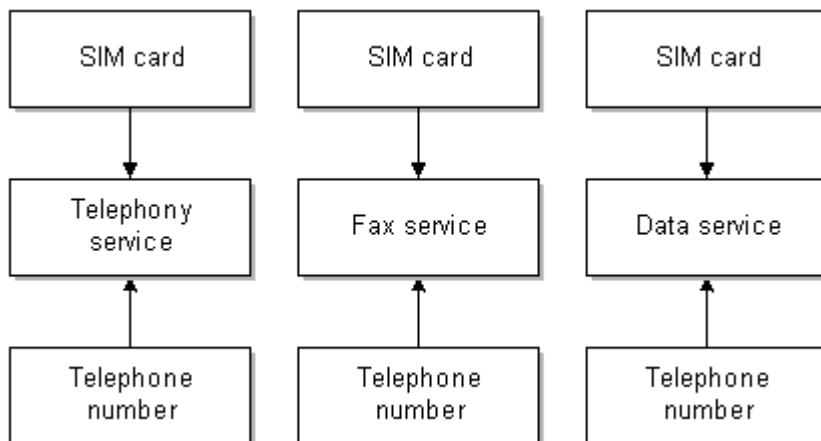
Figure 6-4 SIM Card to Service Association



You can associate SIM cards and numbers in a variety of ways. For example, a customer can own multiple services, SIM cards, and numbers.

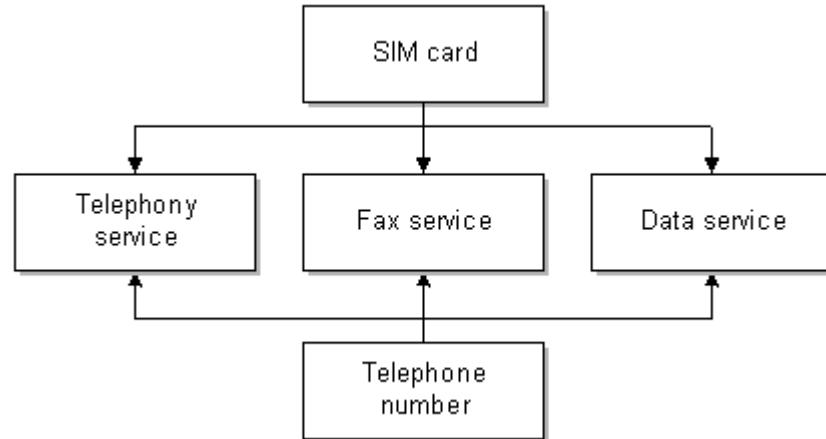
[Figure 6-5](#) shows associations for three services, each with its own SIM card and number:

Figure 6-5 Three SIM Cards to Services Associations



You can also associate SIM cards and numbers with multiple services as shown in [Figure 6-6](#):

Figure 6-6 SIM Card to Multiple Services Associations



You can customize the SIM card and number policy opcodes, `PCM_OP_SIM_POL_DEVICE_ASSOCIATE` and `PCM_OP_NUM_POL_DEVICE_ASSOCIATE`, to customize how services, SIM cards, and numbers can be associated. By default, the following business policies are used:

- Each account can have one or more SIM cards and one or more telephone numbers.
- You can use one SIM card and number for all services, or you can use multiple SIM cards and numbers; for example, you can use a separate SIM card and number for each service.
- A SIM card can be associated with multiple services only if the services are of different types, such as fax and telephony, and each service must belong to the same network element. For example, two `/service/telco/gsm/telephony` services can be associated with the same SIM card to support dual lines on one SIM card.
- A number can be associated with multiple services, with the following restrictions:
 - All services must belong to the same account.
 - All services must belong to the same network element.
 - Each service must be of a different type. For example, you cannot associate one number with two telco services. To do so, you need to customize the policy opcode for the device type, `PCM_OP_SIM_POL_DEVICE_ASSOCIATE` or `PCM_OP_NUM_POL_DEVICE_ASSOCIATE`.
- With the exception of the telephony service, a SIM card can share a service association with only one telephone number. For example, if you associate a SIM card with a fax service and a data service, the phone number must be the same for the fax service and the data service.

If you associate a SIM card with two telephony services, you must associate a different number for each of the telephony services.

- A SIM card and number associated with the same service must have the same network element.

About Assigning Numbers

When assigning numbers:

- You cannot change the network element, number category, vanity flag, or status.
- If an account has more than one telephone number, you can specify one of them as the primary number. If ECE processes an event that includes the IMSI, and if the IMSI is the same for multiple telephone numbers, ECE uses the primary number to look up the service.
- You can assign an IMEI number, but it is not required.
- You can assign vanity numbers to charge a different amount for specific numbers.

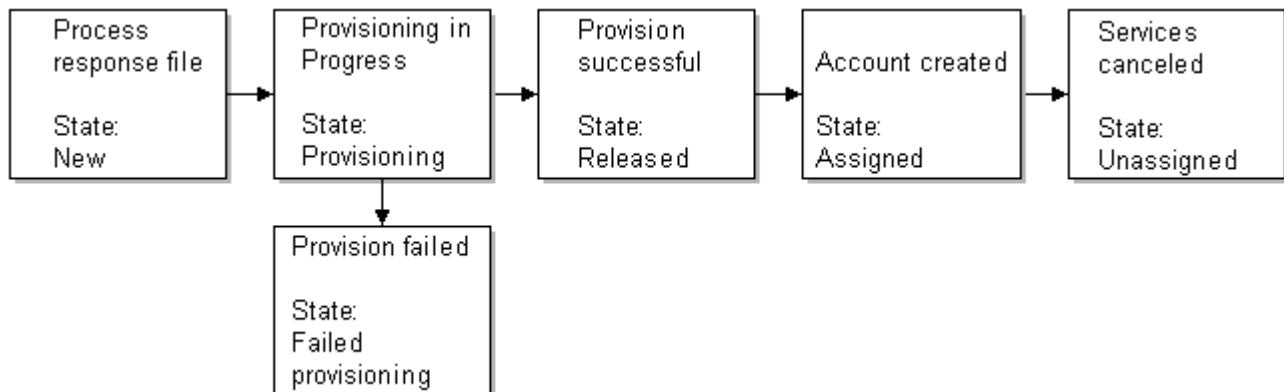
About SIM Card Device States

When you manage SIM cards, the SIM card devices have the following device states:

- When you process a vendor response file, you create SIM card devices in the database. The device state of the SIM cards is *New*.
- While a SIM card is being provisioned, the device state of the SIM cards in the file is set to *Provisioning*. After provisioning, a card is in one of two states:
 - If provisioning succeeds, the device state is set to *Released*.
 - If provisioning fails, the device state is set to *Failed Provisioning*. The device object is created in the database, but it cannot be assigned. (You can change the device state transitions to enable failed SIM cards to be assigned. See "[Customizing SIM Card Device States](#)".)
- When an account is created and assigned a SIM card, the device state is set to *Assigned*. If the account is inactive, the SIM card status remains *Assigned*.
- When all services associated with the SIM card are canceled, the SIM card is disassociated from all services and the device state is set to *Unassigned*. (If the SIM card is associated with multiple services, the SIM card remains *Assigned* as long as one service is active.) By default, you cannot assign a SIM card when it is in the *Unassigned* state. (You can change the device state transitions to allow unassigned SIM cards to be assigned. See "[Customizing SIM Card Device States](#)".)

[Figure 6-7](#) shows the progression of SIM card device states:

Figure 6-7 SIM Card Device State Progression



Changing the Network Element of SIM Cards

You use SIM Administrator to change the network element.

To change the SIM card network element, you search for the SIM card. You can base your search on the SIM card attributes.

From the search results, select the SIM cards that you want to modify. You can only change the network elements for SIM cards in the Released state.

For complete instructions on using SIM Administration Center, see the SIM Administration Center Help.

About SIM Card Customization Options

You can customize how you manage SIM Card inventory as follows:

- You can customize network elements.

Note

By default, there are only sample network elements; you must add your own. See "[Creating Network Elements](#)".

- You can customize SIM card device and service associations. See "[Customizing SIM Card and Service Associations](#)".
- You can customize SIM card device states. See "[Customizing SIM Card Device States](#)".
- You can create custom SIM card formats. See "[Creating SIM Card Formats](#)".
- You can choose to not pre-provision SIM cards, or to pre-provisioning them with a different service, for example, a prepaid service. See "[Customizing How to Pre-Provision SIM Cards](#)".
- You can customize how SIM Administration Center works. See the following:
 - [Specifying the Maximum Quantity of SIM Cards in a Request File](#)
 - [Specifying the Maximum Quantity of SIM Cards in an Order](#)
 - [Specifying the Information You Want to Receive about SIM Cards](#)
 - [Specifying whether the Response File Includes a Check Digit](#)
 - [Specifying the Size of Searches](#)

Getting Information About SIM Card Usage

You can get information about SIM card inventory and status by running the SIM Manager reports. See "SIM Card Inventory Management Report" in *BRM Reports*.

In addition, you can display the history of a customer's SIM card in Customer Center. See information about displaying SIM card history in Customer Center Help.

Installing and Configuring SIM Manager and SIM Administration Center

Learn how to install and customize Oracle Communications Billing and Revenue Management (BRM) SIM Manager and SIM Administration Center.

Topics in this document:

- [Mandatory Configuration Tasks](#)
- [Installing SIM Manager](#)
- [Installing SIM Administration Center](#)
- [Configuring Event Notification for SIM Manager](#)
- [Customizing SIM Card and Service Associations](#)
- [Supporting Encrypted KI Values with Oracle](#)
- [Creating Network Elements](#)
- [Creating SIM Card Formats](#)
- [Customizing SIM Card Device States](#)
- [Customizing Service Types](#)
- [Customizing How to Pre-Provision SIM Cards](#)
- [Loading Order Status Definitions](#)
- [Customizing SIM Administration Center](#)
- [Creating SIM Cards for Testing](#)
- [Uninstalling SIM Manager](#)

For information about SIM Manager and SIM Administration Center, see "[About Managing SIM Card Inventory](#)".

Mandatory Configuration Tasks

In addition to installing the SIM Manager software, you must perform the following configuration tasks before you can use SIM Manager and SIM Administration Center:

 **Note**

You can perform these tasks in any order, but you must install SIM Manager first.

- Define and load network elements. (If Number Manager is already installed, this might already be done.) See "[Creating Network Elements](#)".
- Define SIM device states. This requires two procedures: defining and loading the state transition model, and loading the localized state descriptions. See "[Customizing SIM Card Device States](#)".

- Define SIM device and service associations. See "[Customizing SIM Card and Service Associations](#)".
- Define SIM card formats. See "[Creating SIM Card Formats](#)".
- Preprovision SIM cards, you must specify the pre-provisioning service. If you do not pre-provision SIM cards, you must change the SIM card device states. See "[Customizing How to Pre-Provision SIM Cards](#)".
- Specify whether to pre-provision SIM cards, and if so, the pre-provisioning service. See "[Customizing How to Pre-Provision SIM Cards](#)".
- Load the `order_sim_status.locale` file. You use this file to localize order status descriptions in SIM Administration Center. Even if you do not run a localized version of SIM Administration Center, you need to load this file. See "[Loading Order Status Definitions](#)".
- If you use an Oracle database, allow for encrypted fields. (By default, the KI value is encrypted.) See "[Supporting Encrypted KI Values with Oracle](#)".

Installing SIM Manager

Note

If you already installed the product, you must uninstall its features before reinstalling them.

To implement SIM card provisioning, you need to install GSM Manager, including the wireless provisioning opcodes. See "[Installing and Configuring GSM Manager and Provisioning Data Manager](#)".

To install SIM Manager, see "Installing Individual BRM Components" in *BRM Installation Guide*.

Installing SIM Administration Center

To install SIM Administration Center, see "Installing Individual BRM Clients" in *BRM Installation Guide*.

Configuring Event Notification for SIM Manager

Note

This is a mandatory configuration task.

Note

If you already configured event notification for Number Manager, skip this procedure.

To enable SIM card provisioning, BRM uses event notification. Before you can use SIM Manager, you must configure the event notification feature as follows:

1. If your system has multiple configuration files for event notification, merge them. See "Merging Event Notification Lists" in *BRM Developer's Guide*.

2. Ensure that the merged file includes the entire event notification list in the *BRM_home/sys/data/config/pin_notify* file.
3. (Optional) If necessary to accommodate your business needs, add, modify, or delete entries in your final event notification list. See "Editing the Event Notification List" in *BRM Developer's Guide*.
4. (Optional) If necessary to accommodate your business needs, create custom code for event notification to trigger. See "Triggering Custom Operations" in *BRM Developer's Guide*.
5. Load your final event notification list into the BRM database. See "Loading the Event Notification List" in *BRM Developer's Guide*.

For more information, see "Using Event Notification" in *BRM Developer's Guide*.

Customizing SIM Card and Service Associations

Note

This is a mandatory configuration task.

By default, SIM card devices can be associated with the following services:

- */service/telco/gsm/telephony*
- */service/telco/gsm/fax*
- */service/telco/gsm/data*
- */service/telco/gsm/sms*

To change the services that a SIM card device can be associated with, edit the **pin_device_permit_map** file and load it by using the **load_pin_device_permit_map** utility. See "Defining Device-to-Service Associations" in *BRM Developer's Guide*.

Supporting Encrypted KI Values with Oracle

Note

This is a mandatory configuration task.

By default, the field in the **Idevice/sim** object that stores the KI value (PIN_FLD_KI) is encrypted. If you use an Oracle database, you must edit the Oracle DM configuration file to allow for encrypted fields. Otherwise, you cannot create SIM cards.

Note

Alternatively, you can unencrypt the PIN_FLD_KI field, but this is not recommended for security reasons. See *BRM Developer's Guide*.

1. Open the Oracle DM configuration file (*BRM_home/sys/dm_oracle/pin.conf*).
2. Uncomment the appropriate line for your operating system:

```
#- crypt md5| libpin_crypt.extension "Abracadabra dabracaaabrA"
```

where extension is **so** for Linux.

3. Save the file.
4. Restart the Oracle DM.

Creating Network Elements

Note

This is a mandatory configuration task.

Note

If you already created network elements for telephone numbers, you do not need to create them again. The same network elements are used for numbers and SIM cards.

By default, there are only sample network elements. Before creating SIM cards, you need to add the network elements that you will use with your GSM services.

To customize network elements, you edit the **pin_network_elements** file, then run the "[load_pin_network_elements](#)" utility to load the contents of the file into a **/config/network_element** object in the BRM database.

Note

The utility needs a configuration file in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in *BRM System Administrator's Guide*.

1. Edit the **pin_network_elements** file in *BRM_home/sys/data/config*. The **pin_network_elements** file includes examples and instructions.

The file includes a list of network elements. For example:

```
sample_network_element_1  
sample_network_element_2
```

Note

The **load_pin_network_elements** utility overwrites existing network elements. If you are updating network elements, you cannot load new network elements only. You must load complete sets of network elements each time you run the **load_pin_network_elements** utility.

2. Save the **pin_network_elements** file.
3. Use the following command to load the **pin_network_elements** file:

```
load_pin_network_elements pin_network_elements
```

To verify that the network elements were loaded, you can display the **/config/network_element** object by using the Object Browser, or use the **roj** command with the **testnap** utility. (See "Reading an Object and Writing Its Contents to a File" in *BRM Developer's Guide*.) This example shows a network element in the **/config/network_element** object:

```
PIN_FLD_NETWORK_ELEMENT    STR  [0]  "Sample_Network_Element"
```

Creating SIM Card Formats

Note

This is a mandatory configuration task.

You can customize the list of SIM card formats displayed in SIM Administration Center. The default SIM card formats are **ISO** and **Plug-in**.

To customize SIM card formats, you edit the **sim_card_types.en_US** sample file in the **BRM_home/sysmsgs/simcardtypes** directory. You then use the **load_localized_strings** utility to load the contents of the file into the **/strings** objects. See "load_localized_strings" in *BRM Developer's Guide*.

Customizing SIM Card Device States

Note

This is a mandatory configuration task.

You can customize SIM card device states to support custom business logic. For example, you can change how SIM cards are pre-provisioned by adding custom device states. See "[Customizing How to Pre-Provision SIM Cards](#)".

To customize device states, you edit the **pin_device_state_sim** file in **BRM_home/sys/data/config**, and load it by running the **load_pin_device_state** utility. See "Defining the Device Life Cycle" in *BRM Developer's Guide*.

Adding Device State Names

When you customize device states, you need to add the new device state name to the list of device states that is displayed in SIM Administration Center. To do so, you edit the **sim_device_states.en_US** sample file in the **BRM_home/sysmsgs/simdevicestates** directory. You then use the **load_localized_strings** utility to load the contents of the file into the **/strings** objects. See "load_localized_strings" in *BRM Developer's Guide*.

Customizing Service Types

Note

This is a mandatory configuration task.

You can change the services that are allowed to be associated with SIM card devices. For example, you can associate customized services or restrict the list of services.

To change service types, edit the **pin_device_permit_map_sim** file and load it by running the **load_pin_device_permit_map** utility. See "Defining Device-to-Service Associations" in *BRM Developer's Guide*.

Customizing How to Pre-Provision SIM Cards

Note

This is a mandatory configuration task.

You can customize pre-provisioning as follows:

- You can bypass pre-provisioning. To do so, change the device state transition by editing the **pin_device_state_sim** file and loading it with the **load_pin_device_state** utility.

This example shows how to configure SIM card device states without provisioning:

```
# The storable class to be loaded
/config/device_state/sim
# The type of device for this config file
/device/sim
# Raw -> Released
0: 0: 0: 1: 0: 0
          2: 0: 0
# Release -> Assigned
2: 2: 2: 1: 0: 0
          5: 0: 0
# Assigned -> Unassigned
5: 2: 5: 1: 0: 0
          6: 0: 0
# Unassigned -> Unassigned
6: 3: 6: 1: 0: 0
```

See "[Customizing SIM Card Device States](#)".

- You can change the default pre-provisioning service to use your custom pre-provisioning service. See "[Specifying the Pre-Provisioning Service](#)". You can also change the dummy MSISDN used for pre-provisioning. See "[Changing the Pre-Provisioning MSISDN](#)".

Specifying the Pre-Provisioning Service

By default, there is no pre-provisioning service. If you pre-provision SIM cards, you must create a pre-provisioning service object so you can associate pre-provisioned SIM cards with a service.

When you pre-provision SIM cards, all SIM cards are associated with a service by using a pre-provisioning service object. When you assign a SIM card to an account, the SIM card is associated with the customer's service object. (See "[About Associating a SIM Card with a Customer's Service](#)".)

To change the pre-provisioning service, you edit the **pin_sim_config** file and run the **load_pin_sim_config** utility to load the contents of the file into the **/config/pre_provisioning_sim** object in the BRM database.

ⓘ Note

The **load_pin_sim_config** utility needs a configuration file in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in *BRM System Administrator's Guide*.

1. To use a different service, you need the POID of a service object created by using your custom service storable subclass:
 - a. (Optional) If you use a custom service, create the service. See "Adding Support for a New Service" in *BRM Developer's Guide*.
 - b. Create a package that uses the service (for example, **/service/telco/gsm/telephony**), and create an account using that service. You can use the **root** account.
 - c. Use the **testnap** application or the Object Browser to find the POID of the service object.
- If you use the **testnap** application, use the **robject** command. See "Reading an Object and Writing Its Contents to a File" in *BRM Developer's Guide*.
2. Edit the **pin_sim_config** file in *BRM_home/sys/data/config*:
 - Add the POID of the default pre-provisioning service.
 - Remove the comment character (#) from the beginning of the line.

The default entry is:

```
# Service_Poid:0.0.0.1 /service/telco/gsm/telephony 0 0
```

Example of a customized entry:

```
Service_Poid:0.0.0.1 /service/telco/gsm/telephony 1 0
```

3. Save the **pin_sim_config** file.
4. Use the following command to load the **pin_sim_config** file:

```
load_pin_sim_config pin_sim_config
```

For more information, see "[load_pin_sim_config](#)".

To verify that the pre-provisioning service POID was loaded, you can display the **/config/pre_provisioning_sim** object by using the Object Browser, or use the **robject** command with the **testnap** utility. (See "Reading an Object and Writing Its Contents to a File" in *BRM Developer's Guide*.) The **/config/pre_provisioning_sim** object shows the pre-provisioning service in the **PIN_FLD_SERVICE_OBJ** field:

```
1 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/telephony 1 0
```

Changing the Pre-Provisioning MSISDN

To change the MSISDN used for pre-provisioning, change the following value in the **fm_prov_wireless_pol_create_svc_order.c** policy source code file and recompile it:

```
#define msisdn_val "4085551212"
```

For more information, see "Adding and Modifying Policy Facilities Modules" in *BRM Developer's Guide*.

Loading Order Status Definitions

Note

This is a mandatory configuration task.

The status of SIM card orders is displayed in SIM Administration Center. You cannot change the functionality of the order status: for example, you cannot add a custom status; however, you can localize the status definitions that are displayed.

Note

Even if you do not use localized order status definitions, you need to load the `order_sim_status.locale` file.

To localize the file, you edit a copy of the `order_sim_status.en_US` sample file in the `BRM_home/sysmsgs/ordersimstatus` directory and save the edited version with the correct locale file extension. To load the file, you use the `load_localized_strings` utility to load the contents of the file into the `/strings` objects. See "load_localized_strings" in *BRM Developer's Guide*.

Customizing SIM Administration Center

You can edit the `SIMAdmin.properties` file to customize how SIM Administration Center works. See the following topics:

- [Specifying the Maximum Quantity of SIM Cards in a Request File](#)
- [Specifying the Maximum Quantity of SIM Cards in an Order](#)
- [Specifying the Information You Want to Receive about SIM Cards](#)
- [Specifying whether the Response File Includes a Check Digit](#)

Specifying the Maximum Quantity of SIM Cards in a Request File

You can customize the quantity of SIM cards per vendor request file. By default, the quantity is 5,000. Contact your SIM card vendor to find out if they have a maximum quantity of SIM cards per file.

1. Open the `SIMAdmin.properties` file in the following folder:
C:\Program Files\Common Files\Portal Software
2. Edit the `RequestFileCardsPerBatch` entry:

Note

The value must be a multiple of 50.

`RequestFileCardsPerBatch=5000`

3. Save and close the file.
4. Restart SIM Administration Center.

Specifying the Maximum Quantity of SIM Cards in an Order

By default, the maximum quantity of SIM cards in a single order is 499,995,000.

1. Open the **SIMAdmin.properties** file in the following folder:

C:\Program Files\Common Files\Portal Software

2. Edit the **MaxOrderQuantity** entry:

MaxOrderQuantity=499995000

3. Save and close the file.
4. Restart SIM Administration Center.

Specifying the Information You Want to Receive about SIM Cards

The information returned about SIM cards from the vendor is specified in the **Var_Out** field in the vendor request file. By default, this field is formatted in the request files as follows:

Var_Out:PIN1/PIN2/PUK1/PUK2/ADM1/Ki

To change how the entry is formatted in the request files:

1. Open the **SIMAdmin.properties** file in the following folder:

C:\Program Files\Common Files\Portal Software

2. Edit the **Var_Out** entry:

Var_Out=PIN1/PIN2/PUK1/PUK2/ADM1/Ki

3. Save and close the file.
4. Restart SIM Administration Center.

Specifying whether the Response File Includes a Check Digit

In some cases, the SIM card vendor requires a check digit in the response file. By default, SIM Administration Center assumes that the vendor response file does not include a check digit.

To include a check digit in the response file:

1. Open the **SIMAdmin.properties** file in the following folder:

C:\Program Files\Common Files\Portal Software

2. Edit the **ResponseContainsCheckDigit** entry:

ResponseContainsCheckDigit=false

3. Save and close the file.
4. Restart SIM Administration Center.

Specifying the Size of Searches

You can change the value of the step search that SIM Administration Center uses. For slower systems, a smaller step search size can return results faster.

1. Open the **SIMAdmin.properties** file in the following folder:
C:\Program Files\Common Files\Portal Software
2. Edit the **searchStepSize** entry:
`searchStepSize=100`
3. Save and close the file.
4. Restart SIM Administration Center.

Specifying SIM and IMSI number maximum length

You can determine whether SIM Administration Center prepends zeroes to SIM and IMSI numbers to make them the maximum standard length (19 characters for SIM and 15 for IMSI). This property applies to numbers created in request files and to numbers specified in searches in SIM Administration Center.

By default, zeroes are not prepended to the numbers. Setting the **maximumStandardLength** property to **true** causes SIM Administration Center to prepend zeroes to SIM and IMSI numbers.

 **Note**

In previous versions, the default behavior was to prepend zeroes to the numbers. Therefore, if you need to search on SIM card inventory created in earlier versions, you should set the **maximumStandardLength** property to **true**.

1. Open the **SIMAdmin.properties** file in the following folder:
C:\Program Files\Common Files\Portal Software
2. Edit the **maximumStandardLength** entry:
`maximumStandardLength=true`
3. Save and close the file.
4. Restart SIM Administration Center.

Creating SIM Cards for Testing

To test your wireless integration, you need SIM card devices in your database so you can create accounts. You can create a set of SIM cards to test with.

 **Note**

This procedure creates an order with five SIM cards. The standard file format uses multiples of 50 SIM cards.

To create a set of SIM cards to test with:

1. In SIM Administration Center, create an order using the following entries:
 - Customer Info tab:
 - Customer name: Sample
 - Address: 100 Main St

City: Anytown

State/Province: CA

Zip/Postal: 99999

Country: USA

- Order Details tab:

Quantity: 5

SIM prefix: 1000

Start SIM: 3000

IMSI prefix: 1000

Start IMSI: 4000

SIM card format: Plug-in

Transport key: 255

2. Generate the request file.
3. Open the request file in a text editor. The file name is **Sam00001.inp**.
4. Save the file as **Sam00001.out**. This file will be the vendor response file.
5. Add the following lines after the last line in the file:

```
10004000 10003000 7349 6215 37177972 11267846 83307381
1A73DFCD69F6E11DF3678DC81C9E8057
10004001 10003001 6604 3374 84688421 63029874 91983041
B6FB4A9029FFB2139A2828CF541A566B
10004002 10003002 3708 9818 35042443 24656819 46904714
BB5EA7413C9C237DE94ECC6A55ECA7A6
10004003 10003003 4856 6049 92238513 12035890 92454872
B2888B7F6471501D4AFA1E6D67853581
10004004 10003004 3315 5878 94466890 55544514 19668055
7EC49FD7144221295F2E1090AD92D623
```

6. Save the file.
7. Load the response file.

Uninstalling SIM Manager

To uninstall SIM Manager, see "Uninstalling Optional Components" in *BRM Installation Guide*.

Part IV

Managing Voucher Inventory

Part IV describes how to manage voucher inventory in an Oracle Communications Billing and Revenue Management (BRM) system. It contains the following chapters:

- [About Managing Voucher Inventory](#)
- [Installing and Configuring Voucher Manager and Voucher Administration Center](#)
- [Voucher Manager Utilities](#)

 **Important**

Voucher Manager is deprecated and supported only for backward compatibility. Use Oracle Communications Convergent Charging Controller to manage vouchers.

About Managing Voucher Inventory

Learn how to manage voucher inventory in your Oracle Communications Billing and Revenue Management (BRM) system by using Voucher Manager and Voucher Administration Center.

Topics in this document:

- [About Managing Voucher Cards](#)
- [About Creating Vouchers](#)
- [About Managing Vouchers](#)
- [Managing Expired Vouchers](#)
- [About Voucher Customization Options](#)
- [Getting Information About Voucher Usage](#)
- [How Voucher Association Works](#)
- [Customizing How Voucher Manager Manages Devices](#)
- [Customizing How Voucher Manager Manages Orders](#)

 **Important**

Voucher Manager is deprecated and supported only for backward compatibility. Use Oracle Communications Convergent Charging Controller to manage vouchers.

Before using Voucher Manager and Voucher Administration Center, you need to know the following information:

- Basic BRM concepts. See *BRM Concepts*.
- BRM system architecture. See "BRM System Architecture" in *BRM Concepts*.
- How to create and edit BRM configuration files. See "Using Configuration Files to Connect and Configure Components" in *BRM System Administrator's Guide*.

About Managing Voucher Cards

You use Voucher Manager and Voucher Administration Center to manage the life cycle of voucher cards.

- Voucher Manager includes the opcodes and storables required for creating and managing voucher card devices in the BRM database.
- Voucher Administration Center is the client application that you use to create and process voucher orders.

You use Customer Center to assign voucher cards to customers and manage voucher cards in customer accounts. See "[Getting Information About Voucher Usage](#)".

About Creating Vouchers

Use Voucher Administration Center to order voucher cards from a vendor. For complete instructions on using Voucher Administration Center, see the Voucher Administration Center Help.

Note

- Before you create an order, you must configure voucher data. See "[Mandatory Configuration Tasks](#)". You must also create voucher bundles in PDC, or voucher deals in Pricing Center. See "[About Voucher Bundles and Deals](#)" or Pricing Center Help.
- When creating orders, do not log on to Voucher Administration Center as **root**. If you do, Voucher Administration Center will stop responding when you try to select any deal, other than Product Purchase Fee Event, to retrieve deal information for the order. Instead, log on as a CSR user.

To create vouchers:

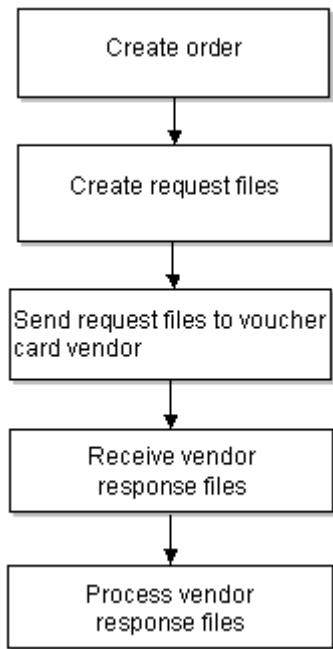
1. Create an order in Voucher Administration Center.

The order is stored in the BRM database and can be updated or canceled. See "[About Voucher Orders](#)".

2. Create one or more *vendor request files*. See "[Vendor Request Files and Voucher Orders](#)".
3. Send the vendor request files to the vendor.
You can use any file transfer method, such as FTP or email.
4. Receive the *vendor response files* and process them in Voucher Administration Center. See "[Vendor Response Files and Voucher Devices](#)".

This creates an **/order/voucher** object for each voucher card.

[Figure 8-1](#) shows the process for creating voucher cards:

Figure 8-1 Voucher Card Creation Process

About Voucher Orders

When you create an order, you can specify the following order attributes:

- Customer name and address information. This is usually your company.
- Vendor name and address of the vendor who creates the vouchers.

 **Note**

There are limitations on how many characters you can use for each piece of customer or vendor information. These limitations allow more characters than you will probably use. For more information, see the **Order** object definition.

- Voucher card attributes:
 - Quantity of voucher cards
 - Package and batch part numbers and quantity
 - Voucher card number
 - Deal associated with the voucher
 - Voucher card expiration date
 - Dealer name

[Figure 8-2](#) shows the **Order Details** tab you use to specify voucher order attributes in Voucher Administration Center:

Figure 8-2 Order Details Tab

The screenshot shows the 'Order Details' tab in a software application. The interface includes the following fields and sections:

- Recharge card:** A dropdown menu showing 'V02 Voucher2'.
- Expiry:** A radio button for 'Days' (selected) and a text input field showing '10'.
- Date:** A radio button for 'Date' and a text input field showing '12/18/2003' with a calendar icon.
- Order number:** A dropdown menu showing '1.35667'.
- Batch part number:** A dropdown menu showing 'BTH001' with '(50 packages)' in parentheses.
- Package part number:** A dropdown menu showing 'PT02' with '(50 vouchers)' in parentheses.
- Start serial number:** A dropdown menu showing 'BTH001' and a text input field showing 'PT02' with a value '734536'.
- Dealer:** A dropdown menu showing 'D001 Airtel'.
- Order size:** A radio button for 'Total vouchers' (selected) and a text input field showing '2500'.
- Number of batches:** A radio button for 'Number of batches' and a text input field showing '1'.
- Deal Selected:** A dropdown menu.
- Deal Details:** A table with columns: Resource, Quantity, Amount, Valid From, and Valid To. The table is currently empty.

About Voucher Bundles and Deals

Customers purchase vouchers by purchasing a voucher bundle or voucher deal. You create a voucher bundle in PDC or a voucher deal in Pricing Center by following the normal procedure. The bundle or deal must have these attributes:

- There must be only one charge offer in the bundle.
- There must be only one balance impact in the charge offer.
- The charge offer must rate only a one-time purchase fee event.
- The charge offer must be an item charge offer.

About Dealers

Dealers are the voucher card distributors from whom you buy the voucher cards. A dealer can also be the service provider. You must load the dealer's details (dealer name and dealer code) before you start using Voucher Manager and Voucher Administration Center.

The dealer codes are used by BRM system and dealer names are displayed in Voucher Administration Center. You can also search for dealer names, and display them in reports.

Use the **load_pin_dealers** utility to load the dealer information into the BRM database. See "[Loading Dealer Details](#)".

About Recharge Card Types

You must create recharge card types and load them into the database. For example, a card type might be \$20 Promotional Card. To create a card type, you must enter the recharge card details:

- Dealer name.

- Dealer code.
- Recharge card type (a description).
- Recharge card code.

The recharge codes are used by BRM system and recharge card types are displayed in Voucher Administration Center.

You must load the recharge card details before you start using Voucher Manager and Voucher Administration Center. Use the **pin_recharge_card_type** utility to load the recharge card information into the BRM database. See "[Loading Recharge Card Details](#)".

About Voucher Details

You must define voucher details such as batch part number, package part number, package quantity, and batch quantity before using Voucher Manager and Voucher Administration Center:

- *A package* is a set of vouchers with the same recharge amount.
- *A batch* is a carton containing several packages of vouchers.

The batch part number and package part number are used to create the voucher card number. The unique serial number for each voucher card is the concatenation of the batch number, the package part number, and the voucher card number. For example, if you enter 0002 for the batch number, 0003 for the package part number, and 1000 for the starting number, the unique serial number would be 000200031000.

Use the **load_pin_voucher_config** utility to load the voucher information into the BRM database. See "[Loading Voucher Details](#)".

About Managing Orders

When you create and manage voucher orders, BRM assigns a state to the order. You use Voucher Administration Center to display the status of an order.

Note

Voucher orders are stored in the database as **order/voucher** objects.

Voucher Manager sets an order's status as follows:

1. When you create an order, the status is *New*. You can save a new order or cancel it. When you cancel an order, the status is set to *Cancelled*. You can also update an order if it has a status of *New*.
2. After you create the vendor request file, the status is changed to *Request*. When an order has the *Request* status, you cannot cancel or modify it.
3. When you begin processing vendor response files, the status is changed to *Partial Receive*.
4. After you process all the vendor response files in an order, the status is changed to *Received*.

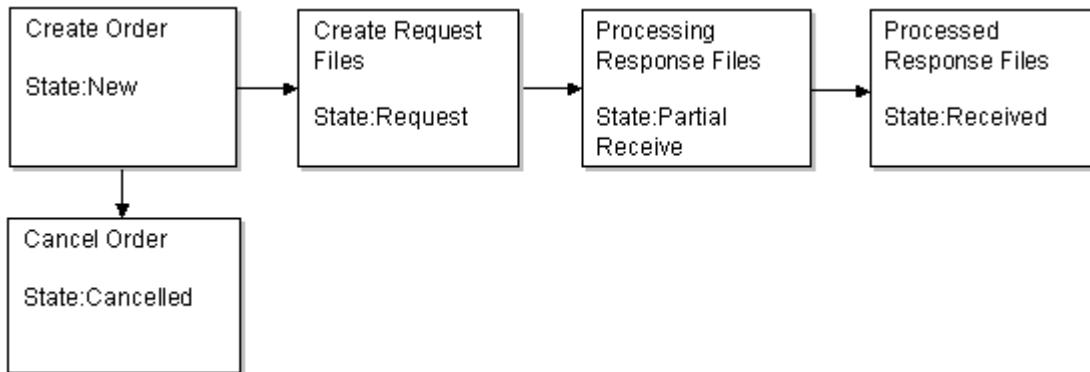
When the order has a *Received* or *Partial Receive* status, you cannot change the order attributes. You can search for orders of any status.

The voucher order states are represented as numbers in Voucher Administration Center.

- 1 = New
- 2 = Request
- 3 = Received
- 4 = Partial Receive
- 5 = Cancelled

[Figure 8-3](#) shows the various order states:

Figure 8-3 Order States



About Modifying Orders

You use Voucher Administration Center to modify an order. You can search for orders by entering a date range, order ID, recharge card type, dealer name, or order state value. For instructions on modifying vouchers, see the Voucher Administration Center Help.

About Voucher Request and Response Files

You manage orders with your voucher vendor by exchanging *request* and *response* files.

- The request files include information about the order, and are used by the vendor to create the vouchers. *Request* files specify the quantity of vouchers and the voucher attributes, such as the customer name and voucher format.
- The response files contain information from the vendor about the vouchers. You load this data into the BRM database.

① Note

Before you create request files or upload response files, you must create vendor request and response file templates. See "[About Request and Response File Templates](#)".

Before creating the request file, you must set the configuration values. The configuration information includes directory path name, the standard naming for the request file, as well as the encryption algorithm.

By default the voucher PIN is encrypted when it is stored in the database. An encrypted request file consists of the serial number, PIN, recharge type, dealer name, quantity, order creation date, order reference number as well as sender and receiver contact information.

 **Note**

Multiple vendor request files are created automatically, based on the size of the order. However, you need to process each vendor *response* file individually.

Vendor Request Files and Voucher Orders

Voucher Administration Center generates vendor request files using the following naming convention:

*CustomerName**nnnnn.inp*

where:

- *CustomerName* is the name in the **Customer name** field when creating the order.
- *nnnnn* specifies the order in which the vendor request files are generated. The number is incremented for each vendor request file generated by an order. For example, the first vendor request file in an order is 00001.

For example, if the customer name is Oracle, the first vendor request file in the order is named **Oracle00001.inp**.

 **Note**

All orders that you create for the same customer use the same file names. Therefore, you should always put request files in a new or empty folder for each order, so files from previous orders are not overwritten.

Vendor request files cannot specify more than 5,000 vouchers. If you create an order for more than 5,000 vouchers, the order is automatically divided into multiple files. For example, if you create an order for 50,000 vouchers, Voucher Administration Center creates 10 vendor request files of 5,000 vouchers each.

You can create a maximum order of 499,995,000 vouchers, which would create 99,999 vendor request files.

You use Voucher Manager to:

- Update the status of the order from *New* to *Request*.
- Set the voucher expiration date.
- Set the request file name and creation date.

You can change the default values for the maximum quantity of vouchers in an order and the maximum number of vouchers in a request file.

Vendor Response Files and Voucher Devices

Vendor response files use the following naming convention:

*CustomerName**_nnnnn.out*

where:

- *CustomerName* is the name in the **Customer name** field when creating the order.
- *nnnnn* is the file order number, which is incremented for each order. The first file in an order uses 00001.

For example, if the customer name is "Oracle," the first file in an order is named **Oracle00001.out**.

You use Voucher Administration Center to process the vendor response files and create devices.

After the vendor response file is processed, the state of the order is changed to Received. In case a partial response file is received, the order state is changed to Partial Receive.

The voucher device created by this process consists of the voucher pin, voucher serial number, dealer name, expiration date, and status. The various states of voucher devices are new, used, and expired. See "[About Voucher Device States](#)".

You can modify voucher devices. See "[About Managing Vouchers](#)".

About Request and Response File Templates

To process vendor request and response files from various vendors, you create vendor file templates. These templates define the request and response file formats for each vendor. You typically use one template for each vendor. A basic format is available in the BRM database, and you can create a new template or open an existing template and modify it.

You create templates by using Voucher Administration Center, and you specify a template when generating the request and response files. The templates are stored in XML format in the BRM database as **/config/inventory_mgmt_template** objects.

[Figure 8-4](#) shows a sample request template:

Figure 8-4 Sample Request Template

[Figure 8-5](#) shows a sample response template:

Figure 8-5 Sample Response Template



About Request and Response File Encryption

You can encrypt the vendor request file, and you can decrypt the vendor response file if your vendor returns response files that are encrypted. Before encrypting files, you need to specify the request and response encryption type in the **EncryptionResources.properties** file.

This file is located in the path: **C:\Program Files\Portal Software\VoucherAdministrationCenter\encrypt.jar**.

About Managing Vouchers

After vouchers are created, you assign them to customer accounts by using Customer Center. See information about topping up accounts in Customer Center Help.

You can search for a single voucher or multiple vouchers and then modify the voucher's state and expiration date. You can also transfer the amount from the voucher to the customer's account.

About Voucher Device States

Voucher devices have the following device states:

- When you process a vendor response file, you create voucher devices in the database. The device state of the vouchers is set to *New*.
- In Customer Center, when an account is created and assigned a voucher, the device state is set to *Used*.
- If an account is inactive or if an amount is transferred from a voucher device to customer's account, the device state is *Used*.
- When the voucher is used up or expires, the device state is set to *Expired*.

Voucher devices have the following states:

- 1 = *New*
- 2 = *Used*
- 3 = *Expired*

The voucher states are represented with their related number in Voucher Administration Center.

Managing Expired Vouchers

If a voucher is not used, it expires. You must use the **pin_voucher_expiration** utility to change the state of the vouchers to *Expired*.

Note

The **pin_voucher_expiration** utility needs a configuration (**pin.conf**) file in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in *BRM System Administrator's Guide*.

Use the following command to run the **pin_voucher_expiration** utility:

```
pin_voucher_expiration [-v] [-d]
```

For more information, see "[pin_voucher_expiration](#)".

About Voucher Customization Options

You can use policy opcodes to customize how vouchers are created and managed.

You can choose to encrypt the vendor request file and decrypt the vendor response file if your vendor returns response files that are encrypted. See "[About Request and Response File Encryption](#)".

Getting Information About Voucher Usage

You can get information about voucher inventory and status by running the Voucher Management reports. See "Voucher Manager Reports" in *BRM Reports*.

How Voucher Association Works

PCM_OP_VOUCHER_ASSOCIATE_VOUCHER calls PCM_OP_DEVICE_ASSOCIATE to perform these operations:

- Calculate the balance impacts of purchasing the deal linked to the voucher device (*device/voucher* object).
- Associate a voucher device with an account or a service.

Before calling the device association opcode, this opcode uses the voucher device ID and PIN specified in the input flist to find the voucher device POID. It passes this POID to the device association opcode.

During voucher top-up operations, the PCM_OP_PYMT_TOPUP opcode calls the PCM_OP_PYMT_POL_VALID_VOUCHER policy opcode, which in turn calls PCM_OP_VOUCHER_ASSOCIATE_VOUCHER.

PCM_OP_VOUCHER_ASSOCIATE_VOUCHER reformats the information it receives from the device association opcode so that it can be used by the top-up opcode. For more information, see *BRM Configuring and Collecting Payments*.

If the device association succeeds, PCM_OP_VOUCHER_ASSOCIATE_VOUCHER returns all currency and noncurrency balance impacts of the voucher.

If the device association fails, the opcode returns an error in the error buffer.

PCM_OP_VOUCHER_ASSOCIATE_VOUCHER returns an error under these circumstances:

- It cannot find a voucher device associated with the input ID and PIN in the BRM database.
- The device cannot be associated with the account or service.

Customizing How Voucher Manager Manages Devices

Use the following opcodes to customize Voucher Manager devices:

- PCM_OP_VOUCHER_POL_DEVICE_CREATE. See "[Customizing Voucher Creation](#)".
- PCM_OP_PYMT_POL_VALID_VOUCHER. See "[Customizing Voucher Validation](#)".
- PCM_OP_VOUCHER_POL_DEVICE_ASSOCIATE. See "[Customizing Voucher Association](#)".
- PCM_OP_VOUCHER_POL_DEVICE_SET_ATTR. See "[Customizing Voucher/Service Association](#)".

Customizing Voucher Creation

PCM_OP_VOUCHER_POL_DEVICE_CREATE validates a device during device creation. For example, this policy opcode verifies that the numbers have the correct number of digits and use the proper syntax. It also verifies that the voucher does not already exist in the database.

You can customize this opcode to change the validation rules for creating voucher devices.

This opcode is called by PCM_OP_DEVICE_CREATE when creating a voucher device.

PCM_OP_VOUCHER_POL_DEVICE_CREATE returns the same values that are in the input flist. Some values might be reformatted, for example, extra spaces might be deleted.

This opcode returns an error when any of the values are not valid or when the voucher already exists in the database.

Customizing Voucher Validation

BRM uses the PCM_OP_PYMT_POL_VALID_VOUCHER policy opcode to validate vouchers. By default, this policy opcode interacts with Voucher Manager. If you use a third-party voucher

management system, you must customize this policy opcode to work with that system. Vouchers are generally used for top-ups and prepayments only.

PCM_OP_PYMT_POL_VALID_VOUCHER is called by PCM_OP_PYMT_TOPUP during voucher top-up operations. To interact with a voucher management system, this policy opcode calls PCM_OP_VOUCHER_ASSOCIATE_VOUCHER. For more information, see *BRM Configuring and Collecting Payments*.

Customizing Voucher Association

PCM_OP_VOUCHER_POL_DEVICE_ASSOCIATE calculates the balance impacts of associating a voucher device (**/device/voucher** object) with an account or a service.

This policy opcode is called by the PCM_OP_DEVICE_ASSOCIATE opcode as follows:

- To retrieve the balance impacts but *not* the information that enables PCM_OP_DEVICE_ASSOCIATE to associate the voucher with an account or a service, it is called with the PCM_OPFLG_CALC_ONLY flag *on* (calculation-only mode).
- To retrieve the balance impacts *and* the information that enables PCM_OP_DEVICE_ASSOCIATE to associate the voucher with an account or a service, it is called with the PCM_OPFLG_CALC_ONLY flag *off*.

If the voucher is not expired, PCM_OP_VOUCHER_POL_DEVICE_ASSOCIATE calls PCM_OP_SUBSCRIPTION_PURCHASE_DEAL to calculate the currency and noncurrency balance impacts of the deal linked to the voucher.

Note

Whether the PCM_OPFLG_CALC_ONLY flag is on or off, this policy opcode always calls PCM_OP_SUBSCRIPTION_PURCHASE in calculation-only mode. During top-up operations, this enables PCM_OP_PYMT_TOPUP to perform the actual balance impacts.

In addition, when called with the PCM_OPFLG_CALC_ONLY flag *off*, PCM_OP_VOUCHER_POL_DEVICE_ASSOCIATE calls PCM_OP_DEVICE_SET_STATE to update the voucher device status as follows:

- If the voucher expiration date is passed, updates it from **new** to **expired**.
- If the voucher is not expired, updates it from **new** to **used**.

If successful, PCM_OP_VOUCHER_POL_DEVICE_ASSOCIATE returns the following information:

- Balance impacts of purchasing the deal linked to the voucher device
- Validity dates of the resources associated with the affected balances

PCM_OP_VOUCHER_POL_DEVICE_ASSOCIATE returns an error in the following cases:

- The voucher PIN specified in the input flist is not in the BRM database.
- The specified voucher is expired.
- The specified voucher is used.

Customizing Voucher/Service Association

PCM_OP_VOUCHER_POL_DEVICE_SET_ATTR ensures that the voucher card number (PIN_FLD_DEVICE_ID) cannot be changed during a device update. It validates a deal object available in the database if the deal object is changed.

You can customize this opcode to change how vouchers are associated with services.

This opcode is called by PCM_OP_DEVICE_SET_ATTR when updating a voucher card device.

PCM_OP_VOUCHER_POL_DEVICE_SET_ATTR returns an error if an attempt is made to change voucher pin or device ID, and when a deal object is changed.

Customizing How Voucher Manager Manages Orders

Use the following opcodes to customize Voucher Manager orders:

- PCM_OP_VOUCHER_POL_ORDER_CREATE. See "[Customizing Order Creation](#)".
- PCM_OP_VOUCHER_POL_ORDER_ASSOCIATE. See "[Customizing Order Association](#)".
- PCM_OP_VOUCHER_POL_ORDER_SET_ATTR. See "[Customizing Order Attributes](#)".
- PCM_OP_VOUCHER_POL_ORDER_PROCESS. See "[Canceling Orders](#)".
- PCM_OP_VOUCHER_POL_ORDER_DELETE. See "[Deleting Orders](#)".

Customizing Order Creation

PCM_OP_VOUCHER_POL_ORDER_CREATE validates the information in the input flist before an order object is created.

Validation consists of:

- Checking for duplicate serial numbers.
- Verifying the quantity ordered.

When calculating the quantity of vouchers within each order the following formula is applied:

Quantity = batch quantity * package quantity * total batches

 **Note**

* indicates multiplication.

- Checking that the quantity is same as the quantity specified in the order.
- Checking for the deal object existence.

You can customize this opcode to change the validation rules for creating **order/voucher** objects.

This policy opcode is called by PCM_OP_ORDER_POL_CREATE when the **order/voucher** object is created.

If any of the checks fail, this opcode returns an error message and logs an error in the CM **pinlog** file, indicating the reason for the failure. The **/order/voucher** object creation is terminated.

Customizing Order Association

PCM_OP_VOUCHER_POL_ORDER_ASSOCIATE ensures that a sub-order cannot be associated or disassociated with the primary order when the order state is not **New**.

You can customize this opcode to change any validation for voucher order association.

This policy opcode is called by PCM_OP_ORDER_POL_ASSOCIATE when associating or disassociating a sub-order with the primary order at account creation.

If the opcode is successful:

- In case of association, the device is associated with the account or service and the device state is set to **Assigned**.
- In case of disassociation, the device is disassociated with the account or service and the device state is set to **Unchanged**.

This opcode returns an error when you associate or disassociate a sub-order with the primary order when the order state is not **New**.

Customizing Order Attributes

PCM_OP_VOUCHER_POL_ORDER_SET_ATTR validates the new values passed into the input list before an order is modified.

This policy opcode is called by PCM_OP_ORDER_POL_SET_ATTR when updating the attributes and description of the **/order/voucher** object.

This policy opcode verifies that:

- The order has not yet been sent to the vendor.
- The modified serial number has no duplicate number in the database.
- The order quantity is correct.

When calculating the quantity of vouchers within each order the following formula is applied:

Quantity = batch quantity * package quantity * total batches

Note

* indicates multiplication.

- The order is not modified when it is in the **Received** or **Canceled** state.

During validation, if it is found that the order was not yet sent to the vendor, the values are validated and updated with the new values passed in.

During validation, if it is found that the order has been sent to the vendor, updating or modifying the order is not allowed.

Canceling Orders

PCM_OP_VOUCHER_POL_ORDER_PROCESS reads the status of an order using the order POID and terminates the processing of the order if the order state is **Cancel**.

A request file is sent to the relevant vendor and the vendor returns a vendor response file. The PCM_OP_ORDER_POL_PROCESS calls this opcode while processing the vendor response files.

Deleting Orders

PCM_OP_VOUCHER_POL_ORDER_DELETE ensures that an order cannot be deleted when the order is in the **Received** or **Partial Receive** state.

This policy opcode is called by PCM_OP_ORDER_POL_DELETE when deleting an order.

If deletion is attempted on the order object in **Request** or **Partial Receive** state, this opcode returns an error.

Installing and Configuring Voucher Manager and Voucher Administration Center

Learn how to install and customize Oracle Communications Billing and Revenue Management (BRM) Voucher Manager and Voucher Administration Center.

Topics in this document:

- [Mandatory Configuration Tasks](#)
- [Installing Voucher Manager](#)
- [Installing Voucher Administration Center](#)
- [Loading Recharge Card Details](#)
- [Loading Dealer Details](#)
- [Loading Voucher Device States](#)
- [Loading Voucher Order States](#)
- [Customizing and Loading Voucher Service Associations](#)
- [Loading Voucher Details](#)
- [Creating Vouchers for Testing](#)
- [Uninstalling Voucher Manager](#)

For information about Voucher Manager and Voucher Administration Center, see "[About Managing Voucher Inventory](#)".

Important

Voucher Manager is deprecated and supported only for backward compatibility. Use Oracle Communications Convergent Charging Controller to manage vouchers.

Mandatory Configuration Tasks

In addition to installing the Voucher Manager software, you must configure and load several text files before you can use Voucher Manager and Voucher Administration Center.

Before performing these configuration tasks, you should gather all the information about your voucher dealers, your recharge card types, and the batch and package numbers you will use.

Note

If you already installed the product, you must uninstall its features before reinstalling them.

- Define voucher device states. This involves two tasks:

- Define and load the state transition model.
- Load localized state descriptions.
See "[Loading Voucher Device States](#)".
- Define order device states. See "[Loading Voucher Order States](#)".
- Specify the services with which vouchers can be associated and load this information into the BRM database. See "[Customizing and Loading Voucher Service Associations](#)".
- Associate the voucher devices with batch and part numbers and with batch and package quantities. See "[Loading Voucher Details](#)".
- Define your voucher dealers and load this information into the database. You can also associate dealers with a specific voucher recharge card type. See "[Loading Dealer Details](#)".
- Define your recharge card types. Each recharge card type is associated with an order configuration, which includes voucher dealer information, batch number, and order size. See "[Loading Recharge Card Details](#)".

Installing Voucher Manager

To install Voucher Manager, see "Installing Individual BRM Components" in *BRM Installation Guide*.

Installing Voucher Administration Center

To install Voucher Administration Center, see "Installing Individual BRM Clients" in *BRM Installation Guide*.

Loading Recharge Card Details

Note

This is a mandatory configuration task.

To load recharge card details into the database, run the **load_pin_recharge_card_type** utility. It loads data into the **/config/recharge_card_type** object. See "[About Recharge Card Types](#)".

Note

The utility needs a configuration (**pin.conf**) file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

1. In the **BRM_home/sys/data/config/pin_recharge_card_type** file, enter the recharge card details.
2. Save and close the file.
3. Use the following command to load the **pin_recharge_card_type** file:

```
load_pin_recharge_card_type cardTypeInputFile
```

For more information, see "[load_pin_recharge_card_type](#)".

Loading Dealer Details

Note

This is a mandatory configuration task.

To load dealer details into the database, you run the **load_pin_dealers** utility to load the data into the **/config/dealers** object. See "[About Dealers](#)".

Note

The utility needs a configuration (**pin.conf**) file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

1. In the **BRM_home/sys/data/config/pin_dealers** file, enter the dealer names and codes.
2. Save and close the file.
3. Use the following command to load the **pin_dealers** file:

```
load_pin_dealers dealerInputFile
```

For more information, see "[load_pin_dealers](#)".

Loading Voucher Device States

Note

This is a mandatory configuration task.

Voucher devices have the following states:

- **1** = New
- **2** = Used
- **3** = Expired

For information on voucher device states, see "[About Voucher Device States](#)".

These states are already defined in the **pin_device_state_voucher** file. You run the **load_pin_device_state** utility to load the data into the **/config/device_state/voucher** object.

Note

The **load_pin_device_state** utility needs a configuration (**pin.conf**) file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

1. Edit the **pin_device_state_voucher** file in **BRM_home/sys/data/config**.

2. Save and close the file.
3. Use the following command to load the **pin_device_state** file:

```
load_pin_device_state stateFileDevice
```

For more information, see "load_pin_device_state" in *BRM Developer's Guide*.

Loading Voucher Order States

Note

This is a mandatory configuration task.

Voucher states are defined in the **pin_order_state_voucher** file. You run the **load_pin_order_state** utility to load the data into the **/config/order_state/voucher** object.

For information on voucher order states, see "[About Managing Orders](#)".

Note

The **load_pin_order_state** utility needs a configuration (**pin.conf**) file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

1. Edit the **pin_order_state_voucher** file in *BRM_home/sys/data/config*.
2. Save and close the **file**.
3. Use the following command to load the **pin_order_state** file:

```
load_pin_order_state stateFileOrder
```

For more information, see "load_pin_order_state" in *BRM Developer's Guide*.

Customizing and Loading Voucher Service Associations

Note

This is a mandatory configuration task.

The voucher service association file (*BRM_home/sys/data/config/pin_device_permit_map_voucher*) specifies the services (**/service** objects) with which voucher devices (**/device/voucher** objects) can be associated in your BRM system. By default, vouchers can be associated with these services:

- **/service/telco/gsm/data**
- **/service/telco/gsm/fax**
- **/service/telco/gsm/sms**
- **/service/telco/gsm/telephony**

① Note

To enable vouchers to be associated with accounts, the **pin_device_permit_map_voucher** file also includes the **/account** object in the list of services with which **/device/voucher** objects can be associated.

To use the voucher management features, you must load the voucher service association file into your BRM database. Before loading the file, you can customize the voucher service associations that it contains. See "Defining Device-to-Service Associations" in *BRM Developer's Guide*.

Loading Voucher Details

① Note

This is a mandatory configuration task.

Voucher details are defined in the **pin_voucher_config** file. You run the **load_pin_voucher_config** utility to load the data into the **/config/voucher** object. See "[About Voucher Details](#)".

① Note

The **load_pin_voucher_config** utility needs a configuration (**pin.conf**) file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

1. Edit the **pin_voucher_config** file in *BRM_home/sys/data/config*.
2. Save and close the **pin_voucher_config** file.
3. Use the following command to load the **pin_voucher_config** file:

```
load_pin_voucher_config voucherInputFile
```

For more information, see "[load_pin_voucher_config](#)".

Creating Vouchers for Testing

You need voucher devices in your database so you can create accounts and then create a set of vouchers.

① Note

This procedure creates an order with five vouchers.

To create a set of vouchers for testing:

1. In Voucher Administration Center, click the **New Order** icon or choose **File - New Order**.
2. Create an order using the following entries:

- **Customer Info tab:**

Customer name: Telecom Services

Contact name: John Brown

Email address: jbrown@example.com

Address: 100 Main St

City: Fremont

State/Province: CA

Zip/Postal: 99999

Country: USA

- **Vendor Info tab:**

Vendor name: Cell Corp

Contact name: Mary John

Email address: jmary@example.com

Address: 200 Main St

City: RedWood City

State/Province: CA

Zip/Postal: 12234

Country: USA

- **Order Details tab:**

Recharge card: V02 Voucher

Order number: 1.35667

 **Note**

You can select either **Recharge Card** or **Order number**, but not both.

Batch part number: BTH001

Package part number: PT02

Start serial number: 734536

This is the voucher card starting number for this order, which you enter in the last field of the **Start serial number** box.

Dealer: D001Airtel

Total Vouchers: 2500

If you select **Total Vouchers** and enter the quantity, the batch quantity is displayed in the **Number of batches** field.

Number of batches: 1

If you select **Number of batches** and enter the batch quantity, the package quantity is displayed in the **Total Vouchers** box.

Days: 10

Date: 2/19/04

 **Note**

You can select either **Days** or **Date**, but not both.

Deal Selected: Voucher_\$300

3. Click **Generate File** to generate the request file.
4. Select or create a directory in which to generate the request files. The file name is **XXXXX.inp**. This file will be the request file sent to the voucher card vendor.
5. Save the file as **XXXXX.out**. This file will be the vendor response file.
6. Process the vendor response file.
7. Load the response file. For complete instructions on using Voucher Administration Center, see Voucher Administration Center Help.

Uninstalling Voucher Manager

To uninstall Voucher Manager, see "Uninstalling Optional Components" in *BRM Installation Guide*.

Voucher Manager Utilities

Learn about the syntax and parameters for the Oracle Communications Billing and Revenue Management (BRM) Voucher Manager utilities.

Topics in this document:

- [load_pin_dealers](#)
- [load_pin_recharge_card_type](#)
- [load_pin_voucher_config](#)
- [pin_voucher_expiration](#)

 **Important**

Voucher Manager is deprecated and supported only for backward compatibility. Use Oracle Communications Convergent Charging Controller to manage vouchers.

[load_pin_dealers](#)

Use this utility to load dealer information into the BRM database. This utility loads dealer information from the **pin_dealer** file to a **/config/dealers** object.

You use this utility to configure a set of dealers for vouchers and identify the dealers associated with any specific card. The dealer names and codes are loaded into the database. See "[About Dealers](#)" and "[Loading Dealer Details](#)".

 **Note**

- When you run the **load_pin_dealers** utility, it overwrites the existing dealers. If you are updating dealers, you cannot load new dealers only. You must load complete sets of dealers each time you run the **load_pin_dealers** utility.
- To connect to the BRM database, the **load_pin_dealers** utility requires a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

Location

BRM_home/bin

Syntax

`load_pin_dealers [-v] [-d] dealerInputFile`

Parameters

-v

Displays information about successful or failed processing as the utility runs.

ⓘ Note

This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter **-v** at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace **filename.log** with the name of the log file:

load_pin_dealers any_other_parameter -v > filename.log

-d

Writes error information for debugging purposes to the utility log file.

dealerInputFile

Specifies the name and location of the file that contains the dealer information. For example, the default file for dealer information is **pin_dealers**. A sample definition file is provided in **BRM_home/sys/data/config/pin_dealers**.

Use this syntax for entries:

```
dealer_name : dealer_code
```

Results

Reports success or displays an error.

By default, the file is located in the same directory as the utility and is called **default.pinlog**. You can specify a different name and location in the **Infranet.properties** file.

load_pin_recharge_card_type

Use this utility to load voucher information into the BRM database. This utility loads voucher information such as dealer name, dealer code, recharge card type, and recharge card code from the **pin_recharge_card_type** file to a **/config/recharge_card_type** object. See "[About Recharge Card Types](#)" and "[Loading Recharge Card Details](#)".

This utility associates card types with their dealers.

ⓘ Note

- When you run the **load_pin_recharge_card_type** utility, it overwrites the existing recharge card types. If you are updating a set of recharge card types, you cannot load new recharge card types only. You must load complete sets of recharge card types each time you run the **load_pin_recharge_card_type** utility.
- To connect to the BRM database, the **load_pin_recharge_card_type** utility requires a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

Location**BRM_home/bin****Syntax****load_pin_recharge_card_type [-v] [-d] cardTypeInputFile****Parameters****-v**

Displays information about successful or failed processing as the utility runs.

 ⓘ Note

This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter **-v** at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace **filename.log** with the name of the log file:

load_pin_recharge_card_type any_other_parameter -v > filename.log**-d**

Writes error information for debugging purposes to the utility log file.

cardTypeInputFile

Specifies the name and location of the file that contains the recharge card type information. For example, the default file for recharge card type information is **pin_recharge_card_type**. A sample definition file is provided in **BRM_home/sys/data/config/pin_recharge_card_type**. Use this syntax for entries:

```
recharge_card_type : recharge_card_code : dealer_name : dealer_code
```

Results

Reports success or displays an error.

By default, the file is located in the same directory as the utility and is called **default.pinlog**. You can specify a different name and location in the **Infranet.properties** file.

load_pin_voucher_config

Use this utility to load voucher information into the BRM database. This utility loads voucher specific information such as batch part no, pack part number, pack quantity and batch quantity from the **pin_voucher_config** file to a **/config/voucher** object. See "[About Voucher Details](#)" and "[Loading Voucher Details](#)".

① Note

- When you run the **load_pin_voucher_config** utility, it overwrites the existing dealers. If you are updating dealers, you cannot load new dealers only. You must load complete sets of dealers each time you run the **load_pin_voucher_config** utility.
- To connect to the BRM database, the **load_pin_voucher_config** utility requires a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

Location

BRM_home/bin

Syntax

load_pin_voucher_config [-d] [-v] voucher_input_file

Parameters

-v

Displays information about successful or failed processing as the utility runs.

① Note

This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter **-v** at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace *filename.log* with the name of the log file:

load_pin_voucher_config any_other_parameter -v > filename.log

-d

Writes error information for debugging purposes to the utility log file.

voucher_input_file

Specifies the name and location of the file that contains the voucher information. For example, the default file for voucher information is **pin_voucher_config**. A sample definition file is provided in *BRM_home/sys/data/config/pin_voucher_config*.

Use this syntax for entries:

batch_part_no : pack_part_no : pack_quantity : batch_quantity

Results

Reports success or displays an error.

By default, the file is located in the same directory as the utility and is called **default.pinlog**. You can specify a different name and location in the **Infranet.properties** file.

pin_voucher_expiration

Use this utility to move BRM vouchers from the New state to the Expired state. See "[Managing Expired Vouchers](#)".

Note

To connect to the BRM database, the **pin_voucher_expiration** utility requires a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

Location

BRM_home/bin

Syntax

`pin_voucher_expiration [-v] [-d]`

Parameters

-v

Displays information about successful or failed processing as the utility runs.

Note

This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter **-v** at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace **filename.log** with the name of the log file:

`pin_voucher_expiration any_other_parameter -v > filename.log`

-d

Writes error information for debugging purposes to the utility log file.

Results

Reports success or displays an error.

By default, the file is located in the same directory as the utility and is called **default.pinlog**. You can specify a different name and location in the **Infranet.properties** file.

Part V

Managing IP Address and APN Inventories

Part V describes how to manage IP address and APN inventories in an Oracle Communications Billing and Revenue Management (BRM) system. It contains the following chapters:

- [About IP Address Manager](#)
- [Installing and Configuring IP Address Manager and IP Address Administration Center](#)

For information about IP Address Manager opcodes, see *BRM Opcode Guide*.

About IP Address Manager

Learn how to use Oracle Communications Billing and Revenue Management (BRM) IP Address Manager to manage your inventories of Internet Protocol (IP) addresses and Access Point Names (APNs). IP Address Manager includes the IP Address Administration Center GUI tool, which provides a graphical interface for managing these devices.

Topics in this document:

- [About Managing IP Address and APN Device Inventories](#)
- [About IP Address Manager](#)
- [IP Address Manager Configuration](#)

 **Note**

IP Address Manager and IP Address Administration Center are optional components, not part of base BRM.

Before setting up IP Address Manager, you need to know the following information:

- Basic BRM concepts. See *BRM Concepts*.
- BRM system architecture. See "BRM System Architecture" in *BRM Concepts*.
- Basic information about how BRM manages devices. See "Managing Devices with BRM" in *BRM Developer's Guide*.
- How to create and edit BRM configuration files. See "Using Configuration Files to Connect and Configure Components" in *BRM System Administrator's Guide*.

About Managing IP Address and APN Device Inventories

BRM stores IP addresses and APNs as devices in the BRM database, so you can use them with the BRM Device Management framework. These devices are stored as BRM objects that can be associated with each other, as well as services and accounts. These IP addresses and APN devices use device states to control their life cycles. You can use the default states and state restrictions, or create new ones to satisfy your business requirements.

You use these IP Manager components to manage IP address and APN devices:

- *IP Address Administration Center* is a GUI application used by operations personnel to manage IP address and APN inventories. For details, see the IP Address Administration Center online Help.
- IP Address Manager also includes command-line interfaces to manage your IP address and APN inventories, including creating and changing IP and APN devices, and associating them with accounts or services. For details, see *BRM Opcode Guide*.

About IP Address Manager

IP Address Manager stores IP addresses and APNs as device objects in your BRM database:

- **/device/ip** objects store information for a specific IP address, including the IP address number, its state, and associations to other objects.
- **/device/apn** objects store information for specific APN devices.

 **Note**

The **/device/apn** subclass of the **/device** storable class does not contain any additional fields. The fields in **/device** are sufficient to use for APN addresses.

This object includes the APN name and ID, its state, and associations to other objects.

Using the IP Address Manager APIs you can associate an IP address or APN with a customer account or any number of services. For details, see *BRM Opcode Guide*.

About IP and APN Device States

IP address and APN devices are controlled by a device states.

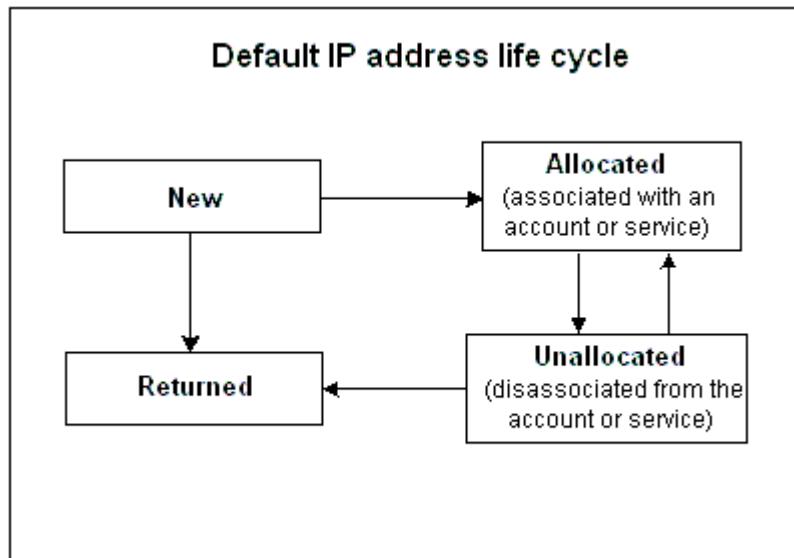
IP Address Device States

By default, the state of an IP device controls whether you can do the following:

- Associate or disassociate an IP address with a customer account or service.
- Delete an IP address.

[Figure 11-1](#) shows the default IP address states and their relationships.

Figure 11-1 Default IP Address States and Relationships



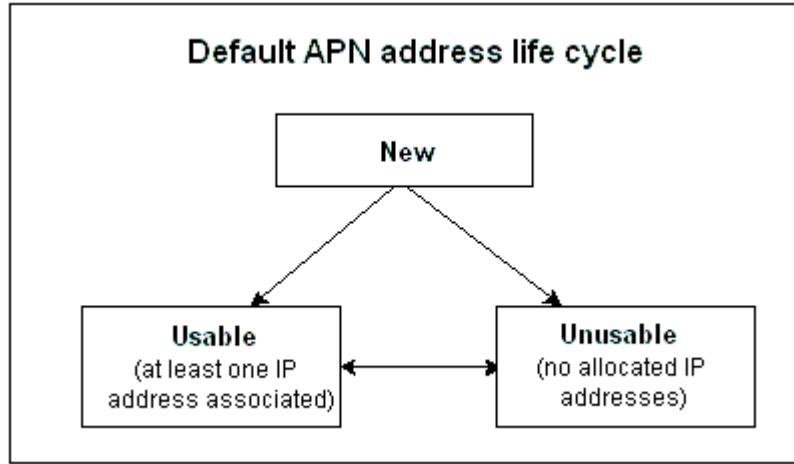
You can add additional states and customize their relationships to suit your business requirements. For details, see "[Customizing IP Address and APN Device Life Cycles](#)".

APN Device States

By default the state of an APN device controls whether the APN is usable or unusable.

[Figure 11-2](#) shows the APN device states and their relationships. These changes are made automatically by IP Address Manager.

Figure 11-2 APN Device States and Relationships



You can add additional states and customize their relationships to suit your business requirements. For details, see "[Customizing IP Address and APN Device Life Cycles](#)".

Associated Services Lists

You can associate services with IP addresses and APNs to satisfy your business requirements. For details, see "[Customizing IP Address and APN Service Association Lists](#)".

IP Address Manager Configuration

Configuring IP Address Manager involves setting or changing IP address and APN life cycles.

To change IP address life cycles:

1. Set IP address states and give them name values in the `pin_device_state_ip` file.
2. Map IP address state name values to strings in the `ip_device_states.locale` file.
3. Create the list of services (as objects) that are permitted to associate with IP addresses in the `pin_device_permit_map_ip` file.

For more information, see "[Installing and Configuring IP Address Manager and IP Address Administration Center](#)".

To change APN life cycles:

1. Set APN states and give them name values in the `pin_device_state_apn` file.

2. Map APN state name values to strings in the **apn_device_states.locale** file.
3. Create the list of services (as objects) that are permitted to associate with APNs in the **pin_device_permit_map_apn** file.

For more information, see "[Installing and Configuring IP Address Manager and IP Address Administration Center](#)".

To customize IP Address Manager, use the IP Manager device opcodes, such as `PCM_OP_IP_POL_DEVICE_CREATE`. See *BRM Opcode Guide*.

Important /device Object Fields

[Table 11-1](#) lists the most important **/device** object fields used by IP Address Manager API.

Table 11-1 Important /device Object Fields

/device Object Field	IP Address Device Use	APN Device Use
PIN_FLD_DEVICE_ID	Stores the IP address (mandatory).	Stores the APN name (mandatory).
PIN_FLD_DESCR	Short description of the IP device.	Short description of the APN device.
PIN_FLD_STATE	Stores the IP device state.	Stores the APN device state.
PIN_FLD_ACCOUNT_OBJ	Stores the account used to create the IP device.	Stores the account used to create the APN device.

Installing and Configuring IP Address Manager and IP Address Administration Center

Learn how to install Oracle Communications Billing and Revenue Management (BRM) IP Address Manager and IP Address Administration Center and how to configure and customize IP Address Manager.

Topics in this document:

- [Installing IP Address Manager](#)
- [Installing IP Address Administration Center](#)
- [Configuring IP Address Manager](#)
- [Uninstalling IP Address Manager](#)

For an overview of IP Address Manager and IP Address Administration Center, see "[About IP Address Manager](#)".

Installing IP Address Manager

 **Note**

If you already installed the product, you must uninstall its features before reinstalling them.

To install IP Address Manager, see "Installing Individual BRM Components" in *BRM Installation Guide*

 **Note**

You should verify that the IP Address Manager Facilities Modules (FMs) were added to the Connection Manager (CM) configuration file.

Installing IP Address Administration Center

To install IP Address Administration Center, see "Installing Individual BRM Clients" in *BRM Installation Guide*.

Configuring IP Address Manager

After installing the IP Address Manager software, you will probably want to configure it to meet your business needs. This section explains the tasks necessary to configure IP Address Manager.

Note

You can perform these tasks in any order, but you must install IP Address Manager first.

- You can define and use new IP address and APN states. This requires two procedures: defining and loading the state transition model, and loading the localized state names. See "[Customizing APN Device States](#)".
- You can define new service and account associations for IP address and APNs. See "[Customizing IP Address and APN Service Association Lists](#)".

Customizing IP Address and APN Device Life Cycles

This section explains how to control device life cycles by changing the default IP address and APN device states to meet your business needs. For a list of the default states and their relationships, see "[About IP and APN Device States](#)". You can change the state names and their relationships to each other. This section explains how to create new IP address and APN device states and state relationships. For information on changing the IP Address Manager opcodes to set these new states, see *BRM Opcode Guide*.

Customizing IP Address Device States

Customizing IP address states includes adding or changing the device states, their names, and their relationships. These tasks are explained in the sections below.

Adding or Changing IP Address States

To customize device states, edit the `pin_device_state_ip` file in `BRM_home/sys/data/config`, and then load it by running the `load_pin_device_state` utility. See "Defining the Device Life Cycle" in *BRM Developer's Guide*.

Adding or Changing Device State Names

When you customize device states, you need to add the new device state name to the list of device states that is displayed in IP Address Administration Center. To do so, you edit the `ip_device_states.en_US` sample file in the `BRM_home/sys/msgs/ipdevicestates` directory. You then use the `load_localized_strings` utility to load the contents of the file into the `/strings` objects. See "load_localized_strings" in *BRM Developer's Guide*.

Customizing APN Device States

Customizing an APN device state includes adding or changing the device states, their names, and their relationships. These tasks are explained in the sections below.

Adding or Changing Device States

To add or change device states, edit the `pin_device_state_apn` file in `BRM_home/sys/data/config`, and then load it by running the `load_pin_device_state` utility. See "Defining the Device Life Cycle" in *BRM Developer's Guide*.

Adding or Changing Device State Names

When you customize device states, you need to add the new device state names to the list displayed in IP Address Administration Center. To do so, you edit the

apn_device_states.en_US sample file in the *BRM_home/sysmsgs/apndevicestates* directory. You then use the **load_localized_strings** utility to load the contents of the file into the **Istrings** objects. See "load_localized_strings" in *BRM Developer's Guide*.

Customizing IP Address and APN Service Association Lists

The following sections explain how to customize the lists of acceptable service and account associations.

Customizing the IP Service Association List

IP Address Manager allows you to associate services with an IP address. To do so, edit the *BRM_home/sys/data/config/pin_device_permit_map_ip* file and then load it into the database by running the *BRM_home/bin/load_pin_device_permit_map* utility.

For more information, see "Defining Device-to-Service Associations" and "load_pin_device_permit_map" in *BRM Developer's Guide*.

Customizing the APN Service Association List

You can customize which services are allowed to be associated with IP address devices. To do so, edit the *BRM_home/sys/data/config/pin_device_permit_map_apn* file and then load it into the database by running the *BRM_home/bin/load_pin_device_permit_map* utility.

For more information, see "Defining Device-to-Service Associations" and "load_pin_device_permit_map" in *BRM Developer's Guide*.

Uninstalling IP Address Manager

To uninstall IP Address Manager, see "Uninstalling Optional Components" in *BRM Installation Guide*.

Part VI

Telco Manager Utilities

Part VI describes Oracle Communications Billing and Revenue Management (BRM) telco service utilities. It contains the following chapters:

- [Telco Service Manager Utilities](#)

Telco Service Manager Utilities

Learn about the syntax and parameters in the Oracle Communications Billing and Revenue Management (BRM) Services Framework utilities.

Topics in this document:

- [load_pin_network_elements](#)
- [load_pin_num_config](#)
- [load_pin_sim_config](#)
- [pin_change_num_quarantine](#)

[load_pin_network_elements](#)

Use the **load_pin_network_elements** utility to load network elements used by SIM Manager and Number Manager into the **/config/network_element** object.

For more information, see "[Installing and Configuring Number Manager and Number Administration Center](#)" and "[Installing and Configuring SIM Manager and SIM Administration Center](#)".

Note

To connect to the BRM database, the **load_pin_network_elements** utility needs a **pin.conf** configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

Location

BRM_home/bin

Syntax

`load_pin_network_elements [-d] [-v] [-i|-r] fileName`

Parameters

-d

Creates a log file for debugging purposes. Use this parameter for debugging when the utility appears to have run with no errors but the data has not been loaded into the database.

-v

Displays information about successful or failed processing as the utility runs.

This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter **-v** at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace *filename.log* with the name of the log file:

pin_change_num_quarantine other_parameter **-v > filename.log**

-i|-r

Specifies whether to load network elements in incremental or replace mode:

- **-i**: Specifies to add the input file elements to existing **/config/network_element** objects.
- **-r**: Specifies to replace the existing **/config/network_element** objects with one specified in the input file.

fileName

The file (**pin_num_config**) that includes the value to use for the default quarantine period.

Results

If the utility does not notify you that it was successful, look in the **default.pinlog** file to find any errors. This file is either in the directory from which the utility was run or in a directory specified in the **pin.conf** configuration file.

To verify that the network elements were loaded, you can display the **/config/network_element** object by using Object Browser, or use the **robj** command with the **testnap** utility. See "Reading an Object and Writing Its Contents to a File" in *BRM Developer's Guide*.

Note

You must restart the Connection Manager (CM) to make new or changed settings available to BRM.

load_pin_num_config

Use this utility to change the default telephone number quarantine period.

For more information, see:

- [Changing the Default Quarantine Period](#)
- [About Number Device States](#)

Note

To connect to the BRM database, the **load_pin_num_config** utility needs a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

Location

BRM_home/bin

Syntax

load_pin_num_config [-d] [-v] fileName

Parameters

-d

Enables debugging mode.

-v

Displays information about successful or failed processing as the utility runs.

fileName

Specifies the name of the file defining the default quarantine period. The default file name is **pin_num_config**.

Results

Reports success or displays an error.

load_pin_sim_config

Use this utility to specify the SIM card pre-provisioning service.

For more information, see:

- [Specifying the Pre-Provisioning Service](#)
- [About SIM Card Pre-Provisioning](#)

Note

To connect to the BRM database, the **load_pin_sim_config** utility needs a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

Location

BRM_home/bin

Syntax

```
load_pin_sim_config [-d] [-v] fileName
```

Parameters

-d

Enables debugging mode.

-v

Displays information about successful or failed processing as the utility runs.

fileName

Specifies the name of the file, such as **pin_sim_config**, that includes the value to use for the pre-provisioning service.

Results

Reports success or displays an error.

pin_change_num_quarantine

Use this utility to remove telephone numbers from quarantine before the default period is over.

For more information, see:

- [Changing the Quarantine Status Manually](#)
- [About Number Device States](#)

ⓘ Note

To connect to the BRM database, the **pin_change_num_quarantine** utility needs a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

Location

BRM_home/bin

Syntax

```
pin_change_num_quarantine [-d] [-v] [-l] [-a] quantityOfNumbers
```

Parameters

-d

Enables debugging mode.

-v

Displays information about successful or failed processing as the utility runs.

This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter **-v** at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace *filename.log* with the name of the log file:

```
pin_change_num_quarantine other_parameter -v > filename.log
```

-l

Displays the results, but does not change the quarantine period.

-a

Includes all numbers, even those that are ready for unquarantine. For example, you might want to unquarantine 50 numbers, but you have 30 numbers ready for unquarantine already. If you use the **-a** parameter, the utility affects only 20 numbers, because 30 are ready to be unquarantined already. Your total quantity is 50.

If you *do not* use the **-a** parameter, the utility unquarantines 50 numbers in addition to the 30 that are already available for unquarantine. Your total quantity is 80.

For more information, see "[Examples](#)".

quantityOfNumbers

The quantity of numbers that you want to make available for unquarantine. You can also use this number in combination with the **-l** parameter to find out how many numbers are available for quarantine.

To find out how many numbers are available for unquarantine, enter a very large number.

Examples

- Listing the available quantity of numbers that can be unquarantined

Command:

```
pin_change_num_quarantine -l 100
```

Results:

```
100 number(s) available for premature unquarantining
```

- Listing the available quantity of numbers that can be unquarantined, including numbers that are already available for unquarantine

Command:

```
pin_change_num_quarantine -la 100
```

Results:

```
40 number(s) found ready to be unquarantined
```

```
60 number(s) available for premature unquarantining
```

- Make 100 numbers available for unquarantine, in addition to any numbers already available for unquarantine

Command:

```
pin_change_num_quarantine 100
```

Results:

```
100 number(s) updated successfully
```

- Make 100 numbers available for unquarantine, taking into account numbers that are already available

Command:

```
pin_change_num_quarantine -a 100
```

Results:

```
40 number(s) found ready to be unquarantined
```

```
60 number(s) updated successfully
```

Results

The utility makes the specified quantity of numbers available, or reports how many numbers are available.

If there are fewer numbers available than specified, this utility displays a warning.