Contents

About This Document ............................................................................................................. v
Document Conventions ........................................................................................................ vi

Chapter 1
About the BRM Charging Driver .......................................................... 1
Overview ...................................................................................................................... 1
Overview of the BRM Charging Driver ................................................................. 1
BRM Integration Summary ......................................................................................... 9

Chapter 2
Installing the BRM Charging Driver ......................................................... 11
Overview of Installing the BRM Charging Driver .............................................. 11
About Installing the BRM Charging Driver ...................................................... 11

Chapter 3
Configuring NCC for the BRM Charging Driver ................................. 13
About configuring NCC for the BRM Charging Driver .................................... 13
Summary of NCC Configuration Tasks ................................................................. 13
Creating the BRM Domain ...................................................................................... 14
Configuring Replication ........................................................................................... 14
About Editing eserv.config Parameters ............................................................. 15
Configuring bcdActionHandler .............................................................................. 16
Configuring bcdBillingClient ................................................................................. 29
Configuring Calls, Events, and Vouchers ........................................................... 34
Modifying the BCD Client Startup Script ............................................................ 37
Creating Balance Type Mappings ......................................................................... 38

Chapter 4
Configuring BRM for the BRM Charging Driver ...................................... 43
About Configuring BRM ......................................................................................... 43
Adding Custom Fields .............................................................................................. 43
Adding Storable Classes ......................................................................................... 45
Creating Header and Library Files for the Custom Classes ................................ 46
Generating the Custom JAR File ............................................................................ 46
Modifying the BRM Configuration Files ............................................................... 47
Configuring BRM for ExtendTimeReservation .................................................. 48

Chapter 5
Creating Products and Deals ................................................................. 51
About Creating Products and Deals ................................................................. 51
Creating a Product and Deal for Voice Calls .................................................... 51
Creating a Product and Deal for Data Calls ....................................................... 52
Creating a Product and Deal for Named Events ................................................ 52
Creating BRM Vouchers ......................................................................................... 54
Creating a Customer ............................................................................................... 54
Chapter 6
Using In-Session Notifications with BRM .................................................. 57
  About Using In-Session Notifications ....................................................... 57
  Enabling In-Session Notifications .............................................................. 57
  Configuring In-Session Notifications .......................................................... 58
  Using Subscription Expiry, Credit Threshold, and Streaming Threshold Notifications .... 60

Chapter 7
Generating Statistics and Reports ................................................................. 63
  Overview of Statistics and Reports ............................................................... 63
  About Statistics and Reports ........................................................................ 63

Chapter 8
Usage Scenarios ............................................................................................. 65
  About Usage Scenarios ................................................................................ 65
  A Voice Call Charged Against BRM ............................................................. 65
  Recharge using BRM Vouchers ................................................................. 81
  Recharge using VWS Vouchers ................................................................. 88
  Data Charging ............................................................................................. 95
  Other Scenarios .......................................................................................... 100

Appendix A
Example BCD section of the eserv.config file ............................................. 213
NCC Glossary of Terms ................................................................................ 219
Index ............................................................................................................ 227
About This Document

Scope

This document presents an overview of the integration of Oracle Communications Network Charging and Control (NCC) and Oracle Communications Billing and Revenue Management (BRM) and describes the processes of installing, configuring and administering the NCC BRM Charging Driver. It also presents several message flow scenarios that describe in detail the interactions that occur for various call charging scenarios. In some cases, this document refers you to existing NCC and BRM documentation to perform specific steps that have already been described in those documentation sets.

Audience

This guide was written primarily for system administrators and persons installing, configuring and administering the BRM Charging Driver. However, sections of the document may be useful to anyone requiring an introduction to the application.

Prerequisites

This document assumes that you are familiar with both the NCC system and the BRM application. This document focuses on the implementation and configuration tasks that are required to integrate the two products.

A solid understanding of UNIX and familiarity with IN concepts and with BRM and its system administration are essential prerequisites for safely using the information contained in this technical guide. Attempting to install, remove, configure or otherwise alter the described systems without the appropriate background skills, could cause damage to the system; including temporary or permanent system malfunctions, loss of service, and inability to recover your system.

Although it is not a prerequisite to using this guide, familiarity with the target platform would be an advantage.

This manual describes system tasks that should only be carried out by suitably trained operators.

Related Documents

The following documents are related to this document:

- Installation Guide
- Service Management System User’s Guide
- Service Management System Technical Guide
- Service Logic Execution Environment Technical Guide
- Charging Control Services User's Guide
- Billing and Revenue Management Concepts
- Billing and Revenue Management Developer's Guide
- Billing and Revenue Management System Administrator's Guide
- Billing and Revenue Management Setting Up Pricing and Rating
- Billing and Revenue Management Telco Integration
## Document Conventions

### Typographical Conventions

The following terms and typographical conventions are used in the Oracle Communications Network Charging and Control (NCC) documentation.

<table>
<thead>
<tr>
<th>Formatting convention</th>
<th>Type of information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Special Bold</strong></td>
<td>Items you must select, such as names of tabs. Names of database tables and fields.</td>
</tr>
<tr>
<td><strong>Italics</strong></td>
<td>Name of a document, chapter, topic or other publication. Emphasis within text.</td>
</tr>
<tr>
<td><strong>Button</strong></td>
<td>The name of a button to click or a key to press. Example: To close the window, either click <strong>Close</strong>, or press <strong>Esc</strong>.</td>
</tr>
<tr>
<td><strong>Key+Key</strong></td>
<td>Key combinations for which the user must press and hold down one key and then press another. Example: <strong>Ctrl+P</strong>, or <strong>Alt+F4</strong>.</td>
</tr>
<tr>
<td><strong>Monospace</strong></td>
<td>Examples of code or standard output.</td>
</tr>
<tr>
<td><strong>Monospace Bold</strong></td>
<td>Text that you must enter.</td>
</tr>
<tr>
<td><strong>variable</strong></td>
<td>Used to indicate variables or text that should be replaced.</td>
</tr>
<tr>
<td><strong>menu option &gt; menu option</strong></td>
<td>Used to indicate the cascading menu option to be selected, or the location path of a file. Example: <strong>Operator Functions &gt; Report Functions</strong> Example: <strong>/IN/html/SMS/HelpText/</strong></td>
</tr>
<tr>
<td><strong>hypertext link</strong></td>
<td>Used to indicate a hypertext link on an HTML page.</td>
</tr>
</tbody>
</table>

Specialized terms and acronyms are defined in the *Glossary* at the end of this guide.
Chapter 1

About the BRM Charging Driver

Overview

Introduction

This chapter describes the architecture and the main features of the BRM Charging Driver, which integrates the Oracle Communications Network Charging and Control (NCC) application with the Oracle Communications Billing and Revenue Management (BRM) system to provide a complete network charging and account settlement system.

For architectural overviews and descriptions of the (NCC) and BRM systems, see the documentation sets for each of these products. For an architectural overview of NCC, see System Administrator's Guide. For an overview of BRM, see BRM Concepts.

In this chapter

This chapter contains the following topics.

Overview of the BRM Charging Driver

BRM Integration Summary

Overview of the BRM Charging Driver

About BRM

Oracle Communications Billing and Revenue Management (BRM) system is a revenue management system for communications and media service providers. Some of the services and capabilities that BRM provides include:

- Managing customers
- Creating price lists for calculating customer charges
- Recording billable events for chargeable interactions
- Rating usage by measuring events and calculating charges
- Creating bills
- Managing payments and accounts receivable

For a thorough introduction to BRM, Oracle strongly recommends that you first read BRM Concepts.

About integrating NCC and BRM

Integrating BRM with NCC provides customers with a charging solution from prepaid to advanced online and offline charging and account settlement.

The BRM Charging Driver is the interface that allows NCC to integrate and communicate with BRM. In the NCC configuration, Charging Control Services (CCS) and Advanced Control Services (ACS) software run on the Service Logic Controller (SLC) platforms and ACS communicates with various networks through NCC network interfaces.
BRM stores the wallet and subscriber data and you can choose to store vouchers either on the NCC Voucher and Wallet Server (VWS) or in the BRM database. The CCS software communicates with BRM through the Portal Communications Module (PCM) API.

Some subscriber data is held on NCC and some on BRM. The NCC prepaid charging platform does not access subscriber data on BRM other than essential account information, and subscriber data is not replicated from NCC to BRM. In addition, the integration of BRM with NCC has the following features:

- NCC does not perform any accounts receivable operations.
- Balances are held only on BRM.
- BRM is solely responsible for credit limits for prepaid accounts.
- You can top up BRM balances through interaction with NCC.
- BRM is responsible for any invoices and statements for prepaid accounts.
- BRM manages the pricing catalog and price elements
- BRM defines and applies any recurring charges. All pricing is configured and applied on BRM.
- BRM is responsible for any re-rating.

The BRM Charging Driver does not support the following NCC capabilities on BRM:

- Named events with a negative number of events
- Direct time charge in credit mode

**BRM and NCC components**

The following diagram illustrates the main components of an NCC system that is integrated with BRM. The BRM Charging Driver components that are required to integrate with BRM are the BCD Client and the BCD actions shared library, which are shown in dark shading. The other components are existing components of NCC and BRM.
The following sections describe the main components of an NCC system that is integrated with BRM:

**The BRM Connection Manager**

NCC connects to BRM through the BRM Connection Manager, which runs as a daemon on a dedicated BRM Connection Manager machine. When NCC requests a connection, a parent Connection Manager process spawns a child process to handle the connection. After that, all communication flows from NCC to the child Connection Manager.

The Connection Manager uses a connection pool, which is a set of connections that it maintains with an application. When NCC requests an operation of BRM, the Connection Manager assigns the request to a connection from the pool and uses it to perform the operation. When the operation completes, the connection is returned to the pool.
If an incoming request cannot be assigned a connection immediately, the Connection Manager queues the request to wait for a configurable period of time for a connection to become available. If a connection does not become available during that time, the Connection Manager throws an exception to indicate that the request timed out.

For more information about the BRM Connection Manager, see the section on system architecture in *BRM Concepts*. For information on configuring NCC to utilize the BRM Connection Manager, see Configuring Connections to the BRM Connection Manager.

**The Portal Communications Module API**

All NCC access to BRM and BRM data is done through the BRM Portal Communications Module (PCM) API. Calls to the PCM API are made through a macro interface rather than directly to API functions. The PCM macros pass operations, called *opcodes*, to BRM to perform various operations. An opcode has an identifying name and number that are associated with a particular function that performs the specified operation.

A PCM opcode receives input data and sends output data in the form of field lists (*flists*), which are lists of field name and value pairs. Each opcode requires its input flist to contain the fields required to perform the operation. For example, to request that BRM debit an account balance, NCC sends the PCM_OP_BILL_DEBIT opcode, including all information in the input flist that is required to debit the account balance.

BRM responds to NCC operation requests by returning an output flist.

**About the session ID**

The BRM Charging Driver uses the PIN_FLD_SESSION_ID field in the input flist to specify a sequence number for a PCM opcode request. This enables BRM to detect whether duplicate opcodes are sent within a session.

If the BRM Charging Driver sends the PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode on a BRM connection and the API times out, the BRM Charging Driver will close the connection and try to send it again on a different connection, possibly on a different Connection Manager. If BRM actually processed the request at the same instant that the API timed out, BRM would receive two identical requests but should apply only one of them. The PIN_FLD_SESSION_ID enables BRM to detect that it has received duplicate opcodes and apply only one.

The BRM Charging Driver increments the value in PIN_FLD_SESSION_ID each time it sends a new opcode, enabling BRM to determine whether it has received duplicates requests for a specific operation. For example, if BRM receives two PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcodes with the same session ID, it knows that they are duplicates and only applies the first one. In particular, both the PCM_OP_TCF_AAA_AUTHORIZE and PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcodes support the use of PIN_FLD_SESSION_ID as a container for a request sequence number.

**About PIN_FLD_DIRECTION**

The BRM Charging Driver uses the PIN_FLD_DIRECTION field in the input flist to indicate either an originating or terminating trigger for voice calls and SMS messages.

For voice calls, the BRM Charging Driver sets PIN_FLD_DIRECTION in the input flist based on the received *eventTypeBCSM* in the CAMEL InitialDP message as follows:

- 0 = Originated (MOC)
- 1 = Terminated (MTC)

For SMS, you can set PIN_FLD_DIRECTION in the input flist through the control plan as follows:

- 0 = Originated (MO SMS)
- 1 = Terminated (MT SMS)

For examples, see the messages in *Usage scenarios* (on page 65).
The BCD Client

The BCD Client is a non-blocking SLEE interface process. You can use multiple BCD Client processes to share the load of BRM Charging Driver operations.

The BCD Client uses the Portal Communications Module (PCM) API to set up connections to the BRM Connection Managers and to send and receive messages over the BRM Portal Communications Protocol (PCP). The BCD Client uses a connection pooling mechanism with load balancing and fail-over to communicate with multiple BRM nodes. Once a connection is established, it is used for multiple operations.

At start-up, the BCD Client reads its configuration information from the BCD section of the eserv.config file and also reads the user names and passwords from the SCP database on the SLC. It will re-read the configuration on receipt of a SIGHUP signal (hang up signal) and on receipt of a REREAD_CONFIG SLEE management event.

The BCD Client processes events that are passed to it over the SLEE by the slee_acs process as a result of running a control plan. The BCD Client simply takes BRM Charging Driver events and turns them into calls to the PCM API.

The BCD actions shared library

The BCD actions shared library resides on the SLC server and translates requests from CCS feature nodes into PCM operations for BRM and then handles the responses.

CCS feature nodes initiate charging actions by calling methods in the NCC acsActions API. The API directs these requests to the shared library for the appropriate protocol, based on the control plan’s current domain. For the BRM domain, the BRM Charging Driver implements the BCD actions shared library for the PCM API. The BCD actions shared library communicates with the BCD Client by sending SLEE events that contain PCM operations.

BRM Charging Driver reports

The BRM Charging Driver reports on events either initiated or observed by the BCD Client. Reporting is done through the standard NCC Service Management System (SMS) reporting mechanism.

For information on generating BRM Charging Driver statistics and reports, see Generating Statistics and Reports (on page 63).

Other NCC components

The following NCC components are also relevant in the integration of NCC and BRM:

- **SLEE and slee_acs**
  The Service Logic Execution Environment (SLEE) manages a group of applications that communicate with each other and share resources efficiently. The slee_acs process is the main process of the Advanced Control Services (ACS) software component and it runs the service logic. It is the process that executes control plans and gives instructions to interfaces that communicate with the network, with billing engines such as BRM, or with other entities. It communicates with these interfaces by using the NCC SLEE API to send and receive SLEE events; thus, the name slee_acs. For more information on the slee_acs process, see Advanced Control Services Technical Guide.

- **The Billing Engine (BE) Client**
  The BE Client provides the interface that processes requests received from the slee_acs process and sent to the NCC Voucher and Wallet Server.

- **Voucher and Wallet Server (VWS)**
Voucher and Wallet Server (VWS) is the NCC real-time charging and subscriber account management component. The VWS solution maintains voucher, wallet and reservation details in a database on the Voucher and Wallet Server. The role of VWS is to manage all the billing and charging information associated with call processing. BRM has more advanced charging features that replace the charging features of the VWS.

- **FOX shared library**
  The FOX shared library is linked to and run by the slee_acs process, which implements charging related actions by sending SLEE FOX events to the Billing Engine (BE) client and receiving SLEE FOX events in return.
  The BE client and the FOX shared library perform the same functions as the BCD Client and the BCD actions shared library, except that SLEE FOX events replace SLEE BCD events, FOX messages replace PCM operations, and VWS replaces BRM.

- **CCS service loader shared library**
  The CCS service loader shared library is primarily responsible for loading both the control plan to be run and the subscriber and service provider profiles from the database. The main piece of information it uses to do this is the calling party’s number.

- **Network control agents**
  Network control agents communicate with the Advanced Control Services component using the internal NCC protocol. In doing so, the control agents translate network protocols such as SIP, MAP, or Diameter into INAP so that a common NCC service logic is possible, independent of the network protocol.

### BRM Charging Driver features

The BRM Charging Driver makes available the following features, or capabilities, when it integrates BRM with NCC.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Charging</td>
<td>The ability to charge for voice and data sessions using the NCC Universal Attempt Termination with Billing (UATB) feature node and BRM</td>
</tr>
<tr>
<td>Time Units</td>
<td>The ability to charge for sessions using the time unit type</td>
</tr>
<tr>
<td>Data Units</td>
<td>The ability to charge for sessions using the data unit type</td>
</tr>
<tr>
<td>Funds Exhaustion</td>
<td>Rejection of further quota requests (duration or volume) when BRM indicates that funds are exhausted</td>
</tr>
<tr>
<td>Network Ends Session</td>
<td>The UATB feature node will report used units (data or time) to BRM when the network indicates that the session terminated</td>
</tr>
<tr>
<td>Cost of Session / Remaining Balance</td>
<td>The ability to report remaining balance and cost of session at the end of a session</td>
</tr>
<tr>
<td>Voice Call Cost</td>
<td>The ability to use the Voice Call Cost feature node to request play a message that states the cost of a session</td>
</tr>
<tr>
<td>SMS Call Info</td>
<td>The ability to use the SMS Call Info feature node to request BRM to send the cost of a session</td>
</tr>
<tr>
<td>Cumulative Balances</td>
<td>The ability to use the Cumulative Balances feature node to play a message that states the balances of an account on BRM</td>
</tr>
<tr>
<td>SMS Account Balances</td>
<td>The ability to use the SMS Account Balances feature node to send the balances of an account that resides on BRM</td>
</tr>
<tr>
<td>Account State Branch</td>
<td>The ability to use the Account State Branch feature node to branch on the state of an account that resides on BRM</td>
</tr>
<tr>
<td>Account Status</td>
<td>The ability in the control plan to play the status and balance of an account on BRM</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Billing Failure Treatment</td>
<td>The ability to specify in the control plan the treatment to use when the UATB feature node cannot contact BRM</td>
</tr>
<tr>
<td>Billing Failure Treatment EDRs</td>
<td>When billing failure treatment conditions occur, event detail records (EDRs) are produced and marked with a special tag so you can identify them for post processing</td>
</tr>
<tr>
<td>Direct Named Event</td>
<td>The ability to use the Named Event feature node to create a Direct Named Event against BRM</td>
</tr>
<tr>
<td>Named Event Reservation</td>
<td>The ability to use the Named Event feature node to create a Named Event Reservation against BRM</td>
</tr>
<tr>
<td>Confirm Named Event</td>
<td>The ability to use the Named Event feature node to confirm a Named Event Reservation against BRM</td>
</tr>
<tr>
<td>Revoke Named Event</td>
<td>The ability to use the Named Event feature node to revoke a Named Event Reservation against BRM</td>
</tr>
<tr>
<td>Recharge of BRM Vouchers</td>
<td>The ability to use the Voucher Redeem and Voucher Recharge feature nodes to recharge BRM vouchers against BRM accounts</td>
</tr>
<tr>
<td>IVR Redemption and Recharge of BRM Vouchers</td>
<td>Using Interactive Voice Response (IVR) as the input method, the ability to use the Voucher Redeem and Voucher Recharge feature nodes to redeem and recharge BRM vouchers against BRM accounts. NCC instructs the IVR to play announcements and user responses.</td>
</tr>
<tr>
<td>IVR Playing of Redeemed Accounts</td>
<td>Using Interactive Voice Response (IVR) as the input method, the ability to use the Play Voucher Redeem Balances feature node to play a message that tells the caller the amounts recharged to accounts on BRM</td>
</tr>
<tr>
<td>SMS Redemption and Recharge of BRM Vouchers</td>
<td>Using SMS as the input method, the ability to use the Voucher Redeem and Voucher Recharge feature nodes to redeem and recharge BRM vouchers against BRM accounts</td>
</tr>
<tr>
<td>USSD Redemption and Recharge of BRM Vouchers</td>
<td>Using Unstructured Supplementary Service Data (USSD) as the input method, the ability to use the Voucher Redeem and Voucher Recharge feature nodes to recharge BRM vouchers against BRM accounts. This feature does not include providing information on recharged amounts by way of USSD. A USSD message is part of an interactive text dialog between a subscriber and the server.</td>
</tr>
<tr>
<td>Web Services Redemption and Recharge of BRM Vouchers</td>
<td>Triggered through web services, the ability to use the Voucher Redeem and Voucher Recharge feature nodes to redeem and recharge BRM vouchers against BRM accounts</td>
</tr>
<tr>
<td>Redemption and Recharge of VWS Vouchers</td>
<td>The ability to use the Voucher Redeem and Voucher Recharge feature nodes to redeem and recharge VWS vouchers against BRM accounts</td>
</tr>
<tr>
<td>IVR Redemption and Recharge of VWS Vouchers</td>
<td>Using Interactive Voice Response (IVR) interaction as the input method, the ability to use the Voucher Redeem and Voucher Recharge feature nodes to recharge VWS vouchers against BRM accounts</td>
</tr>
<tr>
<td>IVR Playing of Redeemed Amounts for VWS Vouchers</td>
<td>Using Interactive Voice Response (IVR) as the input method, the ability to use the Play Voucher Redeem Balances feature node to play a message that tells the caller the amounts recharged to accounts on BRM from VWS vouchers</td>
</tr>
<tr>
<td>SMS Recharge of VWS Vouchers</td>
<td>Using SMS (text message) as the input method, the ability to use the Voucher Redeem and Voucher Recharge feature nodes to recharge VWS vouchers against BRM accounts. This does not include the ability to provide information through SMS on amounts recharged</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>USSD Recharge of VWS Vouchers</td>
<td>Using Unstructured Supplementary Service Data (USSD) as the input method, the ability to use the Voucher Redeem and Voucher Recharge feature nodes to recharge VWS vouchers against BRM accounts. This does not include the ability to provide information through USSD on amounts recharged.</td>
</tr>
<tr>
<td>Web Services Recharge of VWS Vouchers</td>
<td>Triggered through Web services, the ability to use the Voucher Redeem and Voucher Recharge feature nodes to recharge VWS vouchers against BRM accounts. This does not include the ability to provide information on the amounts recharged in the Web services response.</td>
</tr>
<tr>
<td>Voucher Redeem Failure Records</td>
<td>The ability to create a record against the BRM account in the BRM database every time an attempt to redeem a VWS voucher against that account fails</td>
</tr>
<tr>
<td>Reservationless Charging or Refund</td>
<td>The ability to use the DUCR feature node to debit based on a usage amount (duration or volume) without reservation of quota</td>
</tr>
<tr>
<td>Post Call Billing</td>
<td>The ability to use the DUCR feature node to charge for voice calls against BRM accounts after the calls have finished</td>
</tr>
<tr>
<td>Credit Transfer</td>
<td>The ability to use the Credit Wallet Transfer feature node to perform a credit transfer between two BRM accounts</td>
</tr>
<tr>
<td>NCC Defined Credit Transfer</td>
<td>The ability to use the Credit Wallet Transfer feature node to perform a credit transfer between two BRM accounts using a Named event that is mapped to a deal in BRM and a voucher type defined in the NCC system</td>
</tr>
<tr>
<td>Rating Guidance</td>
<td>The ability through a feature node to provide guidance for a particular service that is implemented in a control plan so that BRM is able to apply a particular rate or discount based on criteria that is specified in the control plan. For example, a user is calling a Friends and Family number or is in their home zone.</td>
</tr>
<tr>
<td>Usage Tracking</td>
<td>The ability to track service usage for BRM accounts using NCC tracker wallets</td>
</tr>
<tr>
<td>Connection Manager Details</td>
<td>The ability to specify details of how NCC connects to BRM Connection Managers</td>
</tr>
<tr>
<td>Connection Manager Addresses</td>
<td>The ability to specify the IP address of each BRM Connection Manager to which NCC connects</td>
</tr>
<tr>
<td>Connection Manager Maximum Connections</td>
<td>The ability to specify the maximum number of connections from each NCC process to BRM</td>
</tr>
<tr>
<td>PCM API</td>
<td>The ability to use the PCM API to communicate with BRM Connection Managers</td>
</tr>
<tr>
<td>BRM EDRs</td>
<td>The ability to use the Set Billing Engine EDR feature node to include additional information, such as the charge for a voice or SMS call, in the BRM usage record that is associated with a usage event. The additional information is obtained from the control plan or session context.</td>
</tr>
</tbody>
</table>

**Unavailable BRM Charging Driver features**

When BRM is integrated with NCC through the BRM Charging Driver, you cannot use the following NCC Prepaid Charging feature nodes for subscribers who are charged using a BRM domain:

- Account Activation
- Balance Cascade Override
• Friend Dest Discount
• Periodic Charge State Branching
• Periodic Charge Subscription
• Periodic Charge Transfer
• Play Tariff Plan Announcement
• Scratch Card Recharge
• Scratch Card Recharge Alternate Subscriber
• Set Discount
• Set Tariff Plan Rule
• Select Credit Transfer
• Wallet State Update
The following Prepaid Charging features, or capabilities, are *not available* in a combined NCC and BRM system:

• Periodic Service/Charge logic, which is the ability to perform logic based on periodic charge subscriptions in BRM
• Balance cascade override
• Service logic derived discounts
• Text modification of mid-call tariff change
• Set Discount
• Set Tariff Plan Rule
• Select Credit Transfer
• Wallet State Update
For information about the Prepaid Charging feature nodes, see *Feature Nodes Reference Guide*.

**BRM Integration Summary**

**Integrating NCC and BRM**

The process of integrating NCC and BRM consists of the following general steps, which subsequent chapters describe in detail:

1. Installing the BRM Charging Driver.
   For information on installing the BRM Charging Driver, see *Installing the BRM Charging Driver* (on page 11).
2. Configuring NCC for the BRM Charging Driver.
   For information on configuring NCC, see *Configuring NCC for the BRM Charging Driver* (on page 13).
3. Configuring BRM for the BRM Charging Driver and the specific scenarios that you want to implement.
   For information on configuring BRM, see *Configuring BRM for the BRM Charging Driver* (on page 43).
4. Creating Products and Deals.
   For information on creating products and deals, see *Creating Products and Deals* (on page 51).
5. Using In-Session Notifications in BRM.
   For more information on using In-Session Notifications, see *Using In-Session Notifications with BRM* (on page 57).
For information on generating statistics and reports, see *Generating Statistics and Reports* (on page 63).
Chapter 2

Installing the BRM Charging Driver

Overview of Installing the BRM Charging Driver

Introduction

This chapter explains how to install the BRM Charging Driver components.

In this chapter

This chapter contains the following topics.

About Installing the BRM Charging Driver

About Installing the BRM Charging Driver

Overview of Installing BRM Charging Driver

Note: You must install Portal Development Toolkit 7.5 in /opt/portal/7.5/PortalDevKit/lib as a prerequisite to installing NCC with the BRM Charging Driver.

The NCC BRM Charging Driver supports BRM versions 7.4 and 7.5, with certification for version 7.5.

The general platform requirements for the BRM Charging Driver components are the same as they are for NCC. For information on the general platform requirements for NCC and on installing NCC, see Installation Guide.
Chapter 3

Configuring NCC for the BRM Charging Driver

About configuring NCC for the BRM Charging Driver

Introduction

This chapter explains how to configure NCC for the BRM Charging Driver.

In this chapter

This chapter contains the following topics.

Summary of NCC Configuration Tasks
13
Creating the BRM Domain
14
Configuring Replication
14
About Editing eserv.config Parameters
15
Configuring bcdActionHandler
16
Configuring bcdBillingClient
29
Configuring Calls, Events, and Vouchers
34
Modifying the BCD Client Startup Script
37
Creating Balance Type Mappings
38

Summary of NCC Configuration Tasks

Steps to Configure NCC for the BRM Charging Driver

The following steps summarize what is required to configure NCC for the BRM Charging Driver. The sections that follow describe how to complete these steps in more detail.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create a BRM domain with a domain type of BCD. You can only create one domain of this type. Decide whether to enable both wallets and vouchers or only wallets. If you enable only wallets, NCC will use NCC vouchers to recharge BRM accounts.</td>
</tr>
</tbody>
</table>
| 2    | Configure replication for following SMF database tables, which are used by the BRM Charging Driver:  
   - SMF_NORMALIZATION  
   - SMF_STDEF_BCD |
| 3    | Edit the BRM Charging Driver configuration parameters in the eserv.config file on the SLC server. |
| 4    | (Optional) Set up multiple BCD Client start-up scripts (bcdBeClient.sh) if you have a large system and you want to run more processes. |
| 5    | Start the SLEE. |
| 6    | Create balance types in the NCC Service Management System to match BRM resource IDs and map NCC balance types to BRM resource IDs, which are effectively BRM balance types. For more information, see Creating Balance Type Mappings (on page 38). |
# Creating the BRM Domain

**Procedure to create a BRM domain**

Follow these steps to create a BRM domain.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Log in to the Service Management System (SMS).</td>
</tr>
<tr>
<td>2</td>
<td>From the Services menu, select Prepaid Charging, then Service Management.</td>
</tr>
<tr>
<td>3</td>
<td>Select the Domain tab to view the current list of NCC domains.</td>
</tr>
<tr>
<td>4</td>
<td>Click <strong>New</strong> to create a new domain.</td>
</tr>
</tbody>
</table>
| 5    | In the BE section, enter the following values:  
   - In the Name field, enter **BRM**  
   - From the Type list, select **BCD**  
   - In the Maximum Accounts field, enter the maximum number of accounts in this domain.  
   - Select the **Update Username and Password** check box and enter values for **Username** and **Password**, which are used by the BCD Client to connect to the BRM communication managers. |
| 6    | (Optional) In the **Manages** section, select the **Charging** and **Voucher Management** options. |
| 7    | Click **Save** to save the new domain. |

## Configuring Replication

**Replication for the BRM Charging Driver tables**

Using the SMS UI, configure replication for the SMF_NORMALIZATION and SMF_STDEF_BCD replication groups, which are required for the BRM Charging Driver.
To access the replication settings, select the **Operator Functions** menu in the NCC SMS UI and select **Node Management**. Then select the **Table Replication** tab as shown in the following figure.

For information on using the SMS UI to configure table replication, see *SMS User’s Guide*.

**About Editing eserv.config Parameters**

**Editing eserv.config parameters for the BRM Charging Driver**

The *eserv.config* file is a shared configuration file from which many NCC components read their configuration parameters and data. Each component reads those sections of the file that are relevant to its configuration. The section of the *eserv.config* file for the BRM Charging Driver is labeled BCD and the parameters are divided into two structures: `bcdActionHandler` and `bcdBillingClient`.

The *eserv.config* file resides in the `/IN/service_packages/` directory on each of the SLC servers.

The BRM Charging Driver installation process installs two versions of the *eserv.config* file, a default version and an example version. The default version is installed in the following location:

`/IN/service_packages/BCD/etc/eserv.config.default`

The content of the default version is added to the end of `/IN/service_packages/eserv.config` file. Its BCD section contains only those parameters that are essential to make the BRM Charging Driver functional.

The example version is located here:

`/IN/service_packages/BCD/etc/eserv.config.example`

The BCD section in the example version contains a complete set of NCC configuration parameters for the BRM Charging Driver. It is provided as a reference when you want to change configuration settings.

For a complete listing of the BCD section of the *eserv.config.example* file, see Appendix A.

The following topics in this section describe `bcdActionHandler` and `bcdBillingClient` structures in the BCD section of the *eserv.config* file.
Chapter 3

Configuring bcdActionHandler

bcdActionHandler Content

The bcdActionHandler structure resides in the BCD section of the eserv.config file and defines the following:

- Mapping NCC sessions to BRM services (on page 16)
- Mapping NCC currency codes to BRM values (on page 19)
- Mapping NCC information to BRM fields (on page 19)
- Copying PCM input flist fields to an ACS EDR (on page 20)
- Mapping the location number (on page 21)
- Specifying custom opcodes (on page 21)
- Mapping BRM Piggyback Notifications to NCC Profile Tags (on page 22)
- Configuring additional bcdActionHandler Parameters (on page 25)

Mapping NCC sessions to BRM services

The ServiceProfileTagMapping array allows you to map ServiceKey and BearerID combinations, which identify the type of NCC session, to specific BRM services. If the ServiceKey value or the BearerID values cannot be found, -1 is used. The default configuration, therefore, is one with ServiceKey and BearerID values of -1, which specify a basic voice call. The following example illustrates the array entries:

```plaintext
ServiceProfileTagMapping = [
  {
    # Default config. Basic duration measured voice call
    ServiceKey = -1 # default
    BearerID = -1 # default
    ScalingFactor = 1
    BRMField = "QUANTITY" # or BYTES_UPLINK, BYTES_DOWNLINK
    BRMReqMode = "DURATION" # or VOLUME
    BrmServicePoid = "/service/telco/gsm/telephony"
    BrmObjectType = "gsm"
    UsedUnitsCumulative = false
    DefaultUnitsType = "QUANTITY" # or UP_BYTES, DOWN_BYTES
  }
  {
    # Specific configuration for data calls.
    ServiceKey = 1
    BearerID = 17
    ScalingFactor = 100000 # Bytes per deci-second (=1Mb/second)
    BRMField = "BYTES_UPLINK"
    BRMReqMode = "VOLUME"
    BrmServicePoid = "/service/telco/gsm/data"
    BrmObjectType = "gsm"
    UsedUnitsCumulative = false
    DefaultUnitsType = "UP_BYTES"
  }
]
```

The BRM Charging Driver uses this array when it sends a PCM_OP_TCF_AAA_AUTHORIZE operation to BRM. For information on BRM opcodes, see the BRM Developer's Reference and BRM Telco Integration.

NCC always calls a function called InitialTimeReservation() even when reserving an amount of data. If NCC is reserving data, the DCA program (the Diameter interface) will set the bearer capability to tell the slee_acs process that this is a data call. When slee_acs returns a number of seconds, DCA multiplies the seconds by a scaling factor and grants that many bytes of data.
NCC turns seconds into bytes using a scaling factor when sending PCM_OP_TCF_AAA_AUTHORIZE to BRM, so that BRM can rate the usage. NCC also sets the ReqMode parameter in PCM_OP_TCF_AAA_AUTHORIZE to VOLUME to notify BRM that it is requesting to reserve volume.

The following are the fields in the ServiceProfileTagMapping array:

**ServiceKey**

<table>
<thead>
<tr>
<th>Syntax:</th>
<th>See Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>The service key from the InitialDP that triggered this session. The BRM Charging Driver looks up the ServiceProfileTagMapping section of this configuration based on bearer capability and service key. It defaults to the entry for service key -1 if it does not find an entry. InitialDP is a voice operation but the Messaging Manager product (for SMS) and the DCA product (for data) translate their protocols into INAP and send InitialDPs also.</td>
</tr>
<tr>
<td>Type:</td>
<td>Integer</td>
</tr>
<tr>
<td>Optionality:</td>
<td>Required</td>
</tr>
<tr>
<td>Allowed:</td>
<td>0 to 2147483647</td>
</tr>
<tr>
<td>Default:</td>
<td>-1</td>
</tr>
<tr>
<td>Example:</td>
<td>ServiceKey = 1</td>
</tr>
</tbody>
</table>

**Note:** NCC cannot receive a service key with a value of -1. It is a special value that means 'default' when a service is not found. The meaning of any other service key value is defined by the network operators.

**BearerID**

<table>
<thead>
<tr>
<th>Syntax:</th>
<th>See Example.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>The ITC of the bearer capability provided in the InitialDP and used by NCC to determine the nature of the session - for example, voice or data.</td>
</tr>
<tr>
<td>Type:</td>
<td>Integer</td>
</tr>
<tr>
<td>Optionality:</td>
<td>Required</td>
</tr>
<tr>
<td>Allowed:</td>
<td>0 to 31</td>
</tr>
<tr>
<td>Default:</td>
<td>-1</td>
</tr>
<tr>
<td>Example:</td>
<td>BearerID = 17</td>
</tr>
</tbody>
</table>

**Note:** NCC cannot receive a BearerID with a value of -1. It is a special value that means default when a service is not found. This document uses 0 for voice and 17 for data. For the meaning of other values, see the International Telecommunications Union Telecommunication Standardization Section (ITU-T) Recommendation Q.931. The BRM Charging Driver uses only the 5 bits defined for the information transfer capability.

**ScalingFactor**

<table>
<thead>
<tr>
<th>Syntax:</th>
<th>See Example.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>Determines the conversion ratio between the BRM defined unit for QUANTITY, such as bytes, for example, and decisseconds.</td>
</tr>
<tr>
<td>Type:</td>
<td>Integer</td>
</tr>
<tr>
<td>Optionality:</td>
<td>Required</td>
</tr>
<tr>
<td>Allowed:</td>
<td>Any positive integer</td>
</tr>
<tr>
<td>Default:</td>
<td>None.</td>
</tr>
<tr>
<td>Example:</td>
<td>ScalingFactor = 1</td>
</tr>
</tbody>
</table>
BRMField
Syntax: See Example
Description: This field is required and must be set to QUANTITY, BYTES_UPLINK or BYTES_DOWNLINK. In the PCM_OP_UPDATE_AND_REAUTHORIZE or the PCM_OP_STOP_ACCOUNTING operations, the BRM Charging Driver can put the used units into the QUANTITY, BYTES_UPLINK, or BYTES_DOWNLINK fields. This parameter tells the BRM Charging Driver which field to populate. For more information, see the sections on preparing GSM-specific data in BRM Telco Integration.
Type: String
Optionality: Required
Allowed: QUANTITY, BYTES_UPLINK, BYTES_DOWNLINK
Default: None
Example: BRMField = "QUANTITY"

BRMReqMode
Syntax: See Example
Description: The unit for which to charge.
Type: String
Optionality: Required
Allowed: DURATION or VOLUME
Default: None
Example: BRMReqMode = "DURATION"

BrmServicePoid
Syntax: See Example
Description: This parameter is used in the PIN_FLD_POID parameter of the PCM_OP_TCF_AAA_AUTHORIZE operation.
Type: String
Optionality: Required
Default: None
Example: BrmServicePoid = "service/telco/gsm/data"

BrmObjectType
Syntax: See Example
Description: This parameter is used as a suffix on the PIN_FLD_OBJ_TYPE field of a PCM_OP_TCF_AAA_AUTHORIZE operation.
Type: String
Optionality: Required
Default: None
Example: BrmObjectType = "gsm/ncc"
UsedUnitsCumulative
Syntax: See Example
Description: When slee_acs receives an ApplyChargingReport operation, indicating how many bytes have been used, the value is either cumulative, meaning it is the total bytes used for this session, or it is non-cumulative, meaning it is the number of bytes used this time. A value of true in the UsedUnitsCumulative field indicates to the BRM Charging Driver that the used bytes are cumulative. A value of false indicates the used bytes are non-cumulative.
Type: Boolean
Optionality: Required
Default: None
Example: UsedUnitsCumulative = false

DefaultUnitType
Syntax: See Example
Description: Used to set the UNIT_TYPE parameter when reporting used units to BRM. Specifies which field is used to report usage to BRM in PCM opcodes: PIN_FLD_UP_BYTES, PIN_FLD_DOWN_BYTES, or PIN_FLD_QUANTITY
Type: String
Optionality: Required
Allowed: UP_BYTES, DOWN_BYTES, or QUANTITY
Default: None
Example: DefaultUnitType = "UP_BYTES"

Mapping NCC currency codes to BRM values
The BrmToNccCurrencyMapping array maps BRM resource IDs to NCC currency codes. You must add all currencies that NCC accesses in BRM to this array. BRM resource IDs are defined in the BRM pin_currency.h file, but only those that NCC uses need to be specified here. For the location of the pin_currency.h file, see your BRM documentation.
The following example illustrates the BrmToNccCurrencyMapping array:

```
BrmToNccCurrencyMapping = [ 
  { 
    NCCCode = "NZD" 
    BRMNum = 554 
  } 
  { 
    NCCCode = "EUR" 
    BRMNum = 978 
  } 
]
```
For more information on currencies and synchronizing monetary transactions between NCC and BRM, see Creating Balance Type Mappings (on page 38).

Mapping NCC information to BRM fields
The NccToBrmFieldMapping array maps NCC event detail record (EDR) information items, as well as other NCC items, to BRM fields and their associated types. Each field that is defined can be sent in an NCC information structure within opcodes that are sent to BRM. You trigger the opcodes by using feature nodes such as ChangeEDR or Set Tariff Plan in a control plan.
Note: You must create BRM custom fields to associate with NCC items and perform the related BRM compilations before attempting to use these custom fields. For information on creating BRM custom fields for NCC information items, see About Adding Custom Fields (on page 43).

Each entry in the NccToBrmFieldMapping array must contain the following parameters:

- **NCCItem** - The name of the NCC EDR item or other information item
- **BRMType** - The BRM data type: "STRING, "DECIMAL", "INT", "ENUM", or "TIMESTAMP"
- **BRMField** - The numeric field ID of the custom field, found in the pin_flds.h file on BRM

If an EDR item is not listed here, it will not be copied into a BRM flist.

```plaintext
NccToBrmFieldMapping = [
    {
        NCCItem = "TARIFF_PLAN_ID"
        BRMType = "INT"
        BRMField = 10004
    },
    {
        NCCItem = "NUMBER_OF_EVENTS_ID"
        BRMType = "INT"
        BRMField = 10002
    },
    {
        NCCItem = "EXAMPLE"
        BRMType = "STRING"
        BRMField = 10005
    },
    {
        NCCItem = "VOUCHER"
        BRMType = "STRING"
        BRMField = 10007
    },
    {
        NCCItem = "PIN"
        BRMType = "STRING"
        BRMField = 10008
    }
]
```

### Copying PCM input flist fields to an ACS EDR

The BrmFieldToEdrMapping array is an optional array in the bcdActionHandler structure that enables you to specify any non-array field sent to BRM in an input flist to be copied to an Advanced Control Services event detail record (EDR) for subsequent tracking and analysis. If the BrmFieldToEdrMapping array is not present in the eserv.config file, no flist fields are copied to the EDR.

To specify the flist fields that you want to copy to the EDR, create array items that assign values to the BRMField and EDRItem fields. The BRMField value specifies the integer value of the flist field to be saved and the EDRItem value specifies the name that will be associated with the field in the EDR.

For example, the following entries in BrmFieldToEdrMapping will cause PIN_FLD_AUTHORIZATION_ID to be stored in the EDR as AUTHORIZATION_ID=auth_id_value and PIN_FLD_SESSION_ID to be stored as SESSION_ID=sess_id_value:

```plaintext
BrmFieldToEdrMapping = [
    {
        BRMField = 7450
        EDRItem = "AUTHORIZATION_ID"
    },
    {
        BRMField = 3039
        EDRItem = "SESSION_ID"
    }
]
This adds entries like the following to the end of the EDR:

```
...|AUTHORIZATION_ID=brmClient-username-2013-3-12-2_session_1187270_0|SESSION_ID=3
```

To obtain the integer values of the flist fields that you want to store, see the pin_flds.h file on BRM.

### Mapping the location number

The fields in the `LocationNumberMapping` structure control how the location number is retrieved from ACS and sent to BRM. `BRMField` refers to the BRM field ID of a string field in the GSM_INFO substruct. If this is zero, the location number will not be passed to BRM.

The Primary location profile block and tag specify the primary place to look for the location number in ACS. This defaults to `PT_CC_LOCATION_NUMBER`. If either parameter is zero, the location number will not be passed to BRM.

The secondary location profile block and tag specify the second place to look for the location number in ACS. This defaults to `PT_CC_LOCATION_INFO_LOCATION_NUMBER`.

The secondary location is only used if the primary is specified, but no data is found when retrieving data from the profile block. If either of the secondary location parameters are zero, the profile block will not be searched.

```c
LocationNumberMapping = {
    BRMField = 1251  # Or zero to disable sending of Location Number.
    PrimaryLocationNumberProfileBlock = 18  # Call Context
    PrimaryLocationNumberProfileTag = 327692  # PT_CC_LOCATION_NUMBER. Must be
                                               # non-zero if BRMField is non-zero.
    SecondaryLocationNumberProfileBlock = 18  # Call Context
    SecondaryLocationNumberProfileTag = 327716  # PT_CC_LOCATION_INFO_LOCATION_NUMBER
                                               # or zero to disable secondary choice
}
```

### Specifying custom opcodes

You can specify custom opcodes in place of standard PCM opcodes to allow for the application of custom business rules. To specify custom opcodes, use the `CustomOpCodeMapping` array with `BrmOpCode` and `CustomOpCode` name value pairs, as shown in the following example:

```c
CustomOpCodeMapping = [
    { BrmOpCode = 4007
      CustomOpCode = 11007
    },
    { BrmOpCode = 4026
      CustomOpCode = 11026
    }
]
```

This example would cause the BRM Charging Driver to substitute 11007 as the opcode for a `PCM_OP_TCF_AAA_STOP_ACCOUNTING` (4007) operation and 11026 as the opcode for a `PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE` (4026) operation.

**Note:** Only the operation code is changed. The flists that are sent to and received from BRM will be exactly the same as they would be for the standard opcodes.
Mapping BRM Piggyback Notifications to NCC Profile Tags

The `InSessionNotificationMapping` section allows you to map the data received in the BRM `PIGGYBACK_NOTIFICATIONS` array fields to NCC profile tags to be used in conjunction with an in-session control plan trigger or to be processed when the session has ended (for IVR channels).

This section is optional; if it is not present then In-Session Notifications will be fully disabled.

Sub-sections and individual items are optional; omitting a section or tag will result in that section/item not being copied into the NCC profile tag data to be used by In-Session Notifications. Omitting a section description will also result in the data in that section not being copied.

For more information, see *Using In-Session Notifications with BRM* (on page 57) and BRM documentation on in-session notifications in *Oracle Communications Billing and Revenue Management Telco Integration*.

The following example illustrates the section entries:

```
InSessionNotificationMapping = {
    ProfileBlock = 17    # NCC profile block to populate. Default = 17
    (TEMPORARY STORAGE)

    Language = {
        Description = "Preferred Language"   # Text to match in
        PIGGYBACK_NOTIFICATIONS
        ProfileTag = 37             # NCC profile tag to use (PT_LANGUAGE)
    }

    Channel = {
        Description = "Preferred Channel"    # Text to match in
        PIGGYBACK_NOTIFICATIONS
        ProfileTag = 1312050        # NCC profile tag to use
        (PT_ISN_PREF_CHANNEL)
        InSessionTrigger = [ "Email", "SMS" ] # Which channels require in-
        session trigger
    }

    Time = {
        Description = "Preferred Time"    # Text to match in
        PIGGYBACK_NOTIFICATIONS
        ProfileTag = 1312051        # NCC profile tag to use
        (PT_ISN_PREF_CHANNEL)
    }

    CreditThreshold = {
        Description = "Credit Threshold Breach"   # Text to match in
        PIGGYBACK_NOTIFICATIONS
        ProfileTag = 1312052         # NCC profile tag to use
        (PT_ISN_CT_BALANCE)
        BalanceTypeTag = 1312054    # NCC profile subtag for balance type ID
        (PT_ISN_CT_BAL_TYPE)
        BalanceNameTag = 1312055    # NCC profile subtag for balance type
        name (PT_ISN_CT_BAL_NAME)
        AmountTag = 1312053         # NCC profile subtag for balance amount
        (PT_ISN_CT_BAL_AMOUNT)
        CurrentBalanceTag = 1312056 # NCC profile subtag for current balance
        (PT_ISN_CT_BAL_CURRENT_BAL)
        GroupObjTag = 1312057        # NCC profile subtag for group object
        (PT_ISN_CT_BAL_GROUP_OBJ)
        PercentTag = 1312058        # NCC profile subtag for percent
        (PT_ISN_CT_BAL_PERCENT)
        SourceObjTag = 1312059      # NCC profile subtag for source object
        (PT_ISN_CT_BAL_SOURCE_OBJ)
    }
```

AlertTypeTag = 1312060   # NCC profile subtag for alert type
(PT_ISN_CT_BAL_ALERT_TYPE)
ReasonTag = 1312061     # NCC profile subtag for reason
(PT_ISN_CT_BAL_REASON)
CreditFloorTag = 1312062 # NCC profile subtag for credit floor
(PT_ISN_CT_BAL_CREDIT_FLOOR)
CreditLimitTag = 1312063 # NCC profile subtag for credit limit
(PT_ISN_CT_BAL_CREDIT_LIMIT)
CreditThresholdTag = 1312064 # NCC profile subtag for percent threshold (PT_ISN_CT_BAL_CREDIT_THRESH)
CreditThresholdFixedTag = 1312065 # NCC profile subtag for fixed threshold (PT_ISN_CT_BAL_CREDIT_THRESH_FIXED)
BalanceUnitNameTag = 1312078    # NCC profile subtag for balance unit name (PT_ISN_CT_BAL_UNIT_NAME)
)

SubscriptionExpiry = {
  Description = "Subscription Expired"   # Text to match in PIGGYBACK_NOTIFICATIONS
  ProfileTag = 1312066     # NCC profile tag to use (PT_ISN_SUB_EXPIRY)
  ExpiryDateTag = 1312067  # NCC profile subtag for expiry date (PT_ISN_SUB_EXPIRY_DATE)
}

StreamingThreshold = {
  Description = "Streaming Threshold reached"   # Text to match in PIGGYBACK_NOTIFICATIONS
  ProfileTag = 1312068     # NCC profile tag to use (PT_ISN_STREAM_THRESH)
  CurrentBalanceTag = 1312069 # NCC profile subtag for current balance (PT_ISN_STREAM_THRESH_CURRENT_BAL)
}

Balance = {
  ProfileTag = 1312070     # NCC profile tag to use (PT_ISN_BALANCE)
  BalanceTypeTag = 1312076 # NCC profile subtag for balance type ID (PT_ISN_BALANCE_TYPE)
  BalanceNameTag = 1312077 # NCC profile subtag for balance type name (PT_ISN_BALANCE_NAME)
  AmountTag = 1312071      # NCC profile subtag for amount (PT_ISN_BALANCE_AMOUNT)
  AvailLimitTag = 1312072 # NCC profile subtag for amount (PT_ISN_BALANCE_AVAIL_LIMIT)
  BalanceUnitNameTag = 1312079    # NCC profile subtag for balance unit name (PT_ISN_BALANCE_UNIT_NAME)
}

Status = {
  RatingStatusTag = 1312073 # NCC profile subtag for rating status (PT_ISN_RATING_STATUS)
  LifecycleStateTag = 1312074 # NCC profile subtag for lifecycle state (PT_ISN_LIFECYCLE_STATE)
  FailureReasonTag = 1312075 # NCC profile subtag for failure reason (PT_ISN_FAILURE_REASON)
}

} # End of InSessionNotificationMapping

The following are the fields in the InSessionNotificationMapping array:
Language
The Language array describes the subscriber's preferred language, such as English or Spanish, for example. The BCD actions library matches the preferred language to a language entry in ACS_LANGUAGE and stores the associated NCC language ID in the profile tag.

Channel
The Channel array describes the subscriber's preferred channel for notifications. The preferred channel could be SMS, Email, or IVR, for example. The BCD actions library stores the preferred channel and compares it against the inSessionTrigger list to see if the selected channel requires a real-time control plan trigger to be armed for the notifications.

Time
The Time array contains the subscriber's preferred notification time, specified as a Coordinated Universal Time (UTC) timestamp in string form with three extra digits for milliseconds. The BCD actions library converts the timestamp to a profile date value and stores it in the configured profile tag.

CreditThreshold
The CreditThreshold array contains an entry for each balance that is breaching a credit threshold. A threshold breach can be either a breach up, which is an increased usage of credit or prepaid funds, or a breach down, which is a decreased usage of credit or prepaid funds following a payment or topup. Each balance entry contains fields that are stored in profile tags within a profile array.

Subscription Expiry
The SubscriptionExpriy array stores the timestamp that specifies when the subscription defined by the BRM lifecycle expires. The timestamp is stored as a profile Date.

StreamingThreshold
The StreamingThreshold array contains the current balance value when the notification was triggered. For postpaid balances, this is a positive number and it represents the currently used credit, including the reserved amount for the current call.

Note: NCC represents postpaid balances as negative and adds a minus sign. For example the value 913.67 received from BRM for a balance with scaling factor set to 100 is conveyed to the NCC control plan as -91367

For prepaid balances, this is a negative number that represents the currently available funds after taking into account the amount reserved for the current call.

Note: NCC represents prepaid balances as positive and removes the minus sign. For example the value -45.88 received from BRM for a balance with scaling factor set to 100 is conveyed to the NCC control plan as 4588

Balances
The Balances array contains an entry for each subscriber balance. Each balance entry contains an ID, the reserved amount delta for the call, and the remaining unreserved funds left available.

Status
The Status array contains statuses that indicate why a call was denied, including the result of the rating operation, the reason for an authorization failure, and the lifecycle state, which is included only if lifecycle management is enabled in BRM.

Using output flist fields
The BRM Charging Driver records the following flist fields returned by BRM. The BRM Charging Driver stores these fields in temporary storage profile buffers, using the following profile tags:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCD Result</td>
<td>Integer</td>
<td>7995400</td>
</tr>
</tbody>
</table>
The BRM Charging Driver uses separate profile tags for STOP_ACCOUNTING messages because the UATB feature node can send a PCM_OP_TCF_AAA_AUTHORIZE message followed by a PCM_OP_TCF_AAA_STOP_ACCOUNTING message and receive result, reason, and rating status values for each one. The control plan does not have a chance to branch on these values until the UATB feature node exits, so BCD stores the Stop Accounting values separately.

For an example of how to use these fields, imagine you have a control plan with a UATB feature node. The Billing Failure Treatment (BFT) branch of the feature node goes to a profile branching feature node that branches on BCD Reason. If the reason is 1011 (EXPIRED_CARD), the control plan goes to a play-announcement feature node to announce to the caller, "Sorry. Your card has expired." For any other value, the plan goes to an end feature node.

In addition, you can put an ACS Change CDR feature node in the control plan to write the temporary storage tags to any CDR fields that you want.

### Configuring additional bcdActionHandler Parameters

The `bcdActionHandler` structure also contains the following additional parameters. For examples of these parameters, please see the sample `eserv.config` file in Appendix A.

**accountString**

- **Syntax:** See Example.
- **Description:** The string used for account level on BRM.
- **Type:** String
- **Optionality:** Optional (default used if not set)
- **Allowed:** "account"
- **Default:** "account"
- **Example:**

  ```
  accountString = "account"
  ```

**BrmBadPinEdrActive**

- **Syntax:** See Example
- **Description:** Specifies whether to trigger sending PCM_OP_ACTIVITY to BRM to request BRM to produce an event detail record (EDR) when the subscriber attempts to redeem a voucher using an invalid voucher number or PIN.
- **Type:** Boolean
- **Optionality:** Optional
- **Allowed:** true and false
- **Default:** false
- **Example:**

  ```
  BrmBadPinEdrActive = true
  ```
BrmBadPinResourceId

**Syntax:** parameter = value

**Description:** The CreateEDR action always sends the voucher number and PIN number that were used for a failed attempt to redeem a voucher. If BrmBadPinResourceId is not zero, this resource ID will be used as a counter of bad PIN attempts for this account and it will be incremented.

**Type:** Integer

**Optionality:** Optional

**Allowed:** See Description

**Default:** 0

**Example:** BrmBadPinResourceId = 100001

BrmEdrObjectType

**Syntax:** See Example

**Description:** Specifies the BRM object type to use when sending PCM_OP_ACTIVITY to BRM to request BRM to produce an event detail record (EDR) when the subscriber attempts to redeem a voucher using an invalid voucher number or PIN.

**Type:** String

**Optionality:** Optional

**Allowed:** A valid BRM object type

**Default:** "/voucher"

**Example:** BrmEdrObjectType = "/voucher"

cacheTimeout

**Syntax:** See Example

**Description:** Specifies in seconds the maximum age of cached named event and balance type map data. If the data is older than the number of seconds specified, the cache will be refreshed when the data is needed.

**Type:** Integer

**Optionality:** Optional (default used if not set).

**Allowed:** Any positive integer

**Default:** 60

**Example:** cacheTimeout = 90

clientIDString

**Syntax:** See Example

**Description:** Unique string that identifies a client that will be accessing a BRM server. Prevents multiple clients from accessing BRM with the same authentication IDs.

**Type:** String

**Optionality:** Optional

**Allowed:** See Description

**Default:** Defaults to value of hostname if not specified

**Example:** clientIDString = "client1"
loggedNotificationPeriod
Syntax: See Example
Description: Specifies in seconds how often to announce the number of message parse errors.
Type: Integer
Optionality: Optional (default used if not set).
Allowed: Any positive integer
Default: 30
Example: loggedNotificationPeriod = 300

lowCreditBufferTime
Syntax: See Example
Description: Specifies the number of seconds to hold back as the low credit buffer. For example, if this is set to 10 seconds, the caller will hear a beep 10 seconds before funds expire and the call terminates.
Type: Integer
Optionality: Optional
Allowed: Any positive integer
Default: 10
Example: lowCreditBufferTime = 10

NccInfoFieldDummyEntry
Syntax: See Example
Description: Specifies the BRM field ID of a string field that is configured to be present in the PIN_FLD_NCC_INFO substruct. The field is set to "Present" by the action handler. It is required because PIN_FLD_NCC_INFO must be present in the flist that is passed to opcodes. It is not valid to have an empty substruct; thus the dummy value.
Type: Integer
Optionality: Required
Allowed: See Description
Default: 10001
Example: NccInfoFieldDummyEntry = 10001

NccInfoFieldNumber
Syntax: See Example
Description: Specifies the BRM field ID of the flist substruct field under which all the NCC specific fields get added. This will normally be the numeric value associated with PIN_FLD_NCC_INFO when the BRM instance was customized.
Type: Integer
Optionality: Required
Allowed: See Description
Default: 10000
Example: NccInfoFieldNumber = 10000
poidPrefix

Syntax: See Example
Description: Specifies the first part of the POID string to be used by named events. The event class is appended to it to form the complete POID name. For example, if the poidPrefix is "/service/telco" and the eventClass name is "/gsm/sms", the complete POID name would be "/service/telco/gsm/sms".

When you define a product in the BRM Pricing Center, you specify the BRM object to which it applies. The POID (Portal Object Identifier) is the name of this object and determines which product is used for rating and charging.

Type: String
Optionality: Required
Allowed: See Description
Default: "/service/telco"
Example: poidPrefix = "/service/telco"

roundingScheme

Syntax: See Example
Description: Specifies the rounding scheme for rounding sub-second durations into seconds. The allowable values are:

1 = floor (always round down)
2 = ceiling (always round up)
3 = nearest ( >= .5 rounds up; < than .5 rounds down)

Type: Integer
Optionality: Optional
Allowed: 1, 2, or 3
Default: 3
Example: roundingScheme = 3

serviceDomainInterfaceName

Syntax: See Example
Description: Specifies the interface name of the BCD Client, which is defined in SLEE.cfg. If you create multiple BCD Client start-up scripts, serviceDomainInterfaceName specifies the prefix of the interface name. For example, for the following INTERFACE parameters in SLEE.cfg, which are abbreviated here:

```
INTERFACE=bcdBeClient1 bcdBeClient1.sh ...
INTERFACE=bcdBeClient2 bcdBeClient2.sh ...
```

the value of serviceDomainInterfaceName would be "bcdBeClient". For more information, see Configuring the SLEE to Run New BCD Client Startup Scripts (on page 38).

Type: String
Optionality: Required
Allowed: See Description
Default: None
Example: serviceDomainInterfaceName = "bcdBeClient"

voucherPinLength

Syntax: See Example
Description: Specifies the length of the voucher PIN.
Configuring `bcdBillingClient`

`bcdBillingClient` content

The BCD section of the eserv.conf file also contains the `bcdBillingClient` structure, which defines the following:

- **Mapping opcodes** (on page 29)
- **Configuring BRM connections** (on page 30)
- **Specifying operation timeouts** (on page 30)
- **Configuring additional `bcdBillingClient` parameters** (on page 31)

Mapping opcodes

The `opCodeMapping` section of the `bcdBillingClient` structure specifies the following opcode mapping definitions to associate opcode integers with operation macros that NCC passes to BRM to perform authorization and accounting operations.

**Note:** You should not need to change this section unless you specify custom opcodes. In that case, you should add the custom opcodes to the `opCodeMapping` array for the sake of meaningful debug output. See *Specifying custom opcodes* (on page 21) for more information:

```plaintext
OpCodeMapping = [
    {operation = "PCM_OP_BAL_GET_BALANCE" , opCode = 3701 },
    {operation = "PCM_OP_CUST_MODIFY_CUSTOMER" , opCode = 64 },
    {operation = "PCM_OP_PYMT_TOPUP" , opCode = 3726 },
    {operation = "PCM_OP_SEARCH" , opCode = 7 },
    {operation = "PCM_OP_SUBSCRIPTION_PURCHASE_DEAL" , opCode = 108 },
    {operation = "PCM_OP_SUBSCRIPTION_READ_ACCT_PRODUCTS" , opCode = 81 },
    {operation = "PCM_OP_TCF_AAA_AUTHORIZE" , opCode = 4002 },
    {operation = "PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION" , opCode = 4004 },
    {operation = "PCM_OP_TCF_AAA_QUERY_BALANCE" , opCode = 4104 },
    {operation = "PCM_OP_TCF_AAA_ACCOUNTING" , opCode = 4012 },
    {operation = "PCM_OP_TCF_AAA_STOP_ACCOUNTING" , opCode = 4007 },
    {operation = "PCM_OP_TCF_AAA_STOP_ACCOUNTING_PREP_INPUT" , opCode = 4031 },
    {operation = "PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE" , opCode = 4026 },
    {operation = "PCM_OP_TRANS_ABORT" , opCode = 13 },
    {operation = "PCM_OP_TRANS_OPEN" , opCode = 12 },
    {operation = "PCM_OP_WRITE_FLDS" , opCode = 5 },
    {operation = "PCM_OP_READ_FLDS" , opCode = 4 },
    {operation = "PCM_OP_ACT_ACTIVITY" , opCode = 151 },
    {operation = "PCM_OP_CUST_POL_GET_DEALS" , opCode = 278 },
    {operation = "PCM_OP_BILL_DEBIT" , opCode = 105 }
]
```

For information on BRM opcodes, see the *BRM Developer’s Reference* and *BRM Telco Integration*.
Specifying operation timeouts

The OperationTimeouts array specifies timeout values for the operations in the opCodeMapping array. If the timeout value for an operation is too large, it will increase the delay when a failed connection switches from a failed Connection Manager to a working one. If the timeout value is too small, the operation can time out unnecessarily. Also, some operations will take longer than others. The ideal timeout values will vary from site to site based on network speeds, load, and the speed of the BRM servers. For more information, see the BRM System Administrator's Guide.

The defaultOperationTimeout value specifies a default timeout value for any operation that does not specify a timeout value in OperationTimeouts.

The following example shows the initial values of defaultOperationTimeout and the OperationTimeouts array, as they are provided in the eserv.config file:

```plaintext
# Default operation timeout to use if a specific opcode does not have an
# entry in the OperationTimeouts array
# Default = 250 milliseconds
defaultOperationTimeout = 250

OperationTimeouts = [
    {operation = "PCM_OP_BAL_GET_BALANCE", timeoutMilliseconds = 250 },
    {operation = "PCM_OP_CUST_MODIFY_CUSTOMER", timeoutMilliseconds = 250 },
    {operation = "PCM_OP_PYMT_TOPUP", timeoutMilliseconds = 250 },
    {operation = "PCM_OP_SEARCH", timeoutMilliseconds = 250 },
    {operation = "PCM_OP_SUBSCRIPTION_PURCHASE_DEAL", timeoutMilliseconds = 250 },
    {operation = "PCM_OP_TCF_AAA_AUTHORIZE", timeoutMilliseconds = 100 },
    {operation = "PCM_OP_TCF_AAA_QUERY_BALANCE", timeoutMilliseconds = 20 },
    {operation = "PCM_OP_TCF_AAA_STOP_ACCOUNTING", timeoutMilliseconds = 300 },
    {operation = "PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE", timeoutMilliseconds = 250 },
    {operation = "PCM_OP_TCF_AAA_STOP_ACCOUNTING_PREP_INPUT", timeoutMilliseconds = 250 },
    {operation = "PCM_OP_TRANS_ABORT", timeoutMilliseconds = 250 },
    {operation = "PCM_OP_TRANS_OPEN", timeoutMilliseconds = 250 },
    {operation = "PCM_OP_WRITE_FLD", timeoutMilliseconds = 250 },
    {operation = "PCM_OP_ACT_ACTIVITY", timeoutMilliseconds = 250 },
    {operation = "PCM_OP_READ_FLD", timeoutMilliseconds = 250 }
]
```

Configuring BRM connections

You must also define entries in the BCD section of the eserv.config file to configure connections to BRM Connection Managers. You do this by adding or modifying items in the cmPointers array in the ConnectionManager structure. The ip parameter specifies the IP address of a specific Connection Manager machine. If you supply multiple IP addresses in the cmPointers array, the BCD Client process selects the actual machine and port to use for a particular connection.

At start up, the BCD Client tries to establish all of the connections specified in the cmPointers array, for example 250 to the first Connection Manager, 500 to the second one, 250 to the third one, and so on.

If, for any reason, the BCD Client does not have a full complement of connections to any Connection Manager, the BCD Client tries to re-establish the connections as soon as possible. If an attempt to connect to a Connection Manager fails, the BCD Client process will not attempt to re-connect to that CM for a number of seconds equal to the value specified by the recoverCmPtrSeconds parameter. The attempt to establish the connection will time out after the number of milliseconds that is equal to the value specified by the contextOpenTimeoutMilliseconds parameter.
At any given time, some of the connections will be idle and some will be busy, a busy connection being one that is waiting for a response to an operation that the BCD Client has sent. When the BCD Client wants to choose a connection to send another operation it chooses the Connection Manager with the lowest ratio of busy connections to total connections.

The following example illustrates the definition of the `ConnectionManager` structure:

```plaintext
ConnectionManager = {
    database = 1
    service = "/service/pcm_client"
    cmPointers = [
        { cmPtr = "ip 192.168.111.111 12010", poolSize = 15},
        { cmPtr = "ip 192.168.111.112 12010", poolSize = 50},
        { cmPtr = "ip 192.168.111.111 12011", poolSize = 15},
        { cmPtr = "ip 192.168.111.112 12011", poolSize = 15}
    ]
}
```

The `database` and `service` values should not be changed. NCC passes these values to the BRM Connection Manager when they connect.

The `poolSize` value specifies to the BCD Client the number of connections to create with the Connection Manager specified by the IP address in the `ip` parameter. The Connection Manager accepts connections until it has too many or the BCD Client reaches the limit specified by the `poolSize` parameter. You need one connection for each simultaneous transaction. For example, if machine 192.168.111.112 is twice as fast as machine 192.168.111.111, you might want to set the value of the `poolSize` parameter for 192.168.111.112 to be twice as large.

### Configuring additional `bcdBillingClient` parameters

The `bcdBillingClient` structure also contains the following additional parameters. For examples of these parameters, please see the sample `eserv.config` file in Appendix A.

**contextOpenTimeoutMilliseconds**

**Syntax:** See Example

**Description:** Specifies the length of time in milliseconds after which the BCD Client will stop trying to establish a context (connection) with a given BRM Connection Manager. If the connection is not established within this period, the connection attempt is abandoned and the BCD Client can try to establish a connection with the next Connection Manager.

**Type:** Integer

**Optionality:** Optional (default used if not set).

**Default:** 5000

**Example:**

```
contextOpenTimeoutMilliseconds = 5000
```

**defaultOperationTimeout**

**Syntax:** `parameter = value`

**Description:** Used if a specific opcode does not have an entry in the `OperationTimeouts` array in the `eserv.config` file. Default is 250 milliseconds. For a description of the `OperationTimeouts` array, see *Specifying Operation Timeouts* (on page 30).

**Type:** Integer

**Optionality:** Optional (default used if not set).

**Allowed:** See Description

**Default:** 250
Example: \[ \text{defaultOperationTimeout} = 250 \]

**latencyStatisticsInterval**

Syntax: See Example

Description: A positive integer value causes latency measurements to be logged in the BCD Client log under the bcLatency debug section. This parameter specifies in seconds the interval at which measurements are logged. A value of 0 disables logging of latency measurements.

Type: Integer

Optionality: Optional (default used if not set)

Allowed: See Description.

Default: 300

Example: \[ \text{latencyStatisticsInterval} = 240 \]

**maxContextIdleTimeSeconds**

Syntax: See Example

Description: The number of seconds to leave an idle connection open before closing it.

Type: Integer

Optionality: Optional (default used if not set).

Allowed: Positive integer

Default: 10

Example: \[ \text{maxContextIdleTimeSeconds} = 10 \]

**maxOutstandingRequests**

Syntax: See Example

Description: The maximum number of outstanding events for each BCD Client process. Events will not be sent to a BCD Client with more than this number of events outstanding.

Type: Integer

Optionality: Optional (default used if not set).

Allowed: Any positive integer

Default: 1000

Example: \[ \text{maxOutstandingRequests} = 1000 \]

**maxPollMilliseconds**

Syntax: See Example

Description: The maximum number of milliseconds to elapse before checking SLEE events.

Type: Integer

Optionality: Optional (default used if not set).

Allowed: Any positive integer

Default: 50

Example: \[ \text{maxPollMilliseconds} = 1 \]
maxSelectMicroseconds

Syntax: See Example
Description: The maximum number of microseconds to wait for PCM messages before checking for SLEE events. Oracle recommends that you leave this setting at the default value of 50.
Type: Integer
Optionality: Optional (default used if not set).
Allowed: See Description
Default: 50
Example: maxSelectMicroseconds = 50

maxTries

Syntax: See Example
Description: Specifies the maximum number of attempts, including the first, to send an operation to BRM.
Type: Integer
Optionality: Optional (default used if not set)
Allowed: A positive integer
Default: 3
Example: maxTries = 3

recordCMIPAddressInStats

Syntax: See Example
Description: Specifies whether to include the IP address of the destination communications manager in statistics.
Type: Boolean
Optionality: Optional (default used if not set).
Allowed: true, false
Default: false
Example: recordCMIPAddressInStats = false

recordOpcodeInStats

Syntax: See Example
Description: Specifies whether to include opcodes in statistics.
Type: Boolean
Optionality: Optional (default used if not set).
Allowed: true, false
Default: false
Example: recordOpcodeInStats = false

recordPortInStats

Syntax: See Example
Description: Specifies whether to include the port number of the destination communications manager in statistics.
Type: Boolean
Optionality: Optional (default used if not set).
Allowed: true, false
Default: false
Example: recordPortInStats = false

recoverCmPtrSeconds
Syntax: See Example
Description: The number of seconds for which any IP address and port combination will be marked as not working. After this number of seconds elapses, the connection will be tried again for reuse.
Type: Integer
Optionality: Optional (default used if not set).
Allowed: Any positive integer
Default: 5
Example: recoverCmPtrSeconds = 5

## Configuring Calls, Events, and Vouchers

### Configuring NCC for voice calls

Typically, voice calls are controlled in NCC by the Universal Attempt Termination with Billing (UATB) feature node in the control plan. To charge a BRM account for a voice call, do one of the following:

- Select the BRM domain, which is the BCD domain type, when configuring the subscriber’s wallet.
- Use a Set Active Domain feature node in the control plan with Wallet enabled for the BCD domain type.

To enable voice calls, you must create a voice call entry in the bcdActionHandler.ServiceProfileTagMapping list in the eserv.config file on the SLC. The following example shows a sample entry:

```plaintext
ServiceProfileTagMapping = [
  ... [
    # Prepaid voice
    ServiceKey = 1 # Defined in triggering rules on the HLR
    BearerID = 0 # Information Transfer Capability 0 = Speech per Q.931
    ScalingFactor = 1
    BRMField = "QUANTITY"
    BRMReqMode = "DURATION"
    BrmServicePoid = "/service/telco/gsm/telephony"
    BrmObjectType = "gsm/ncc"
    UsedUnitsCumulative = false
    DefaultUnitType = "QUANTITY"
  ]
]
```

For descriptions of the ServiceProfileTagMapping parameters in the eserv.config file, see Mapping NCC Sessions to BRM Services (on page 16).

### Configuring NCC for data calls

Typically, data calls are controlled in NCC by the Universal Attempt Termination with Billing (UATB) feature node in the control plan. To charge a BRM account for a data call, do one of the following:

- Select the BRM domain, which is the BCD domain type, when configuring the subscriber’s wallet.

34  NCC BRM Charging Driver Technical Guide
Use a Set Active Domain feature node in the control plan with Wallet enabled for the BCD domain type.

To enable data calls, you must create a data call entry in the bcdActionHandler.ServiceProfileTagMapping list in the eserv.config file on the SLC. The following example shows a sample entry for a data call:

```bash
ServiceProfileTagMapping = [
  ...
  {
    # Configuration for Data Calls
    ServiceKey = 1
    BearerID = 17
    ScalingFactor = 100000  # Bytes per deci-second (= 1Mb per second)
    BNMField = "BYTES_UPLINK"
    BNMReqMode = "VOLUME"
    BrmServicePoid = "/service/telco/gsm/data"
    BrmObjectType = "gsm"
    UsedUnitsCumulative = false
    DefaultUnitType = "UP_BYTES"
  }
]
```

For descriptions of the ServiceProfileTagMapping parameters in the eserv.config file, see *Mapping NCC Sessions to BRM Services* (on page 16).

### Configuring NCC for named events

Examples of named events include SMS, MMS, and email.

Named events are controlled by the Named Event feature node in the NCC control plan. To charge a BRM account, do one of the following:

- Select the BRM domain (BCD domain type) when configuring the subscriber's wallet
- Use a Set Active Domain feature node in the control plan with Wallet enabled for the BCD domain type

#### Configuring NCC for named event reservations and reservable direct named events

You make named event reservations using a Named Event feature node. Select the Named Event Feature Selection Reserve Event option, followed by another node set to Confirm Event or Revoke Event.

To create a direct named event, select Named Event Feature Selection Direct Event option. Set the Number of Events field to a positive value.

You must handle named event refunds separately. When NCC sends BRM a PCM_OP_TCF_AAA_ACCOUNTING opcode for a direct named event and receives a response that indicates not all of the events have been charged, it sends a PCM_OP_TCF_AAA_REFUND operation to refund the partial charge and treat the direct named event as a failure.

To enable refunds, NCC must send BRM a one-time opcode that configures it to allow refunds for the desired event. For more information, see *Enabling refunds for direct named events*.

For information on configuring named event refunds, see *Configuring Named Event Refunds* (on page 53).

#### Enabling refunds for direct named events

When the BRM Charging Driver sends PCM_OP_TCF_AAA_ACCOUNTING for a direct named event, it can receive a response indicating that some but not all of the events have been charged. For example, BRM can charge for twenty five percent of an SMS. When this occurs, the BRM Charging Driver sends PCM_OP_TCF_AAA_REFUND to refund the partial charge and treats the direct named event as a failure.
To enable this capability, you must first send a one-time opcode to configure BRM to allow refunds for direct named events.

When the BRM Charging Driver invokes an opcode, it passes a flist and a parameter called flags, which is a bit field with each bit having a different meaning. The BRM code makes this definition for PCM_OPFLAG_ADD_ENTRY:

```c
#define PCM_OPFLAG_ADD_ENTRY 0x0020 /* wfld, incfld only */
```

In C, you would use the following code to set the ADD_ENTRY flag and send an opcode:

```c
flags |= PCM_OPFLAG_ADD_ENTRY
PCM_OP_SEND(ctxtp, opcode, flags, flistp, &ebuf);
```

Follow these steps to configure BRM to allow refunds for direct named events:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Set the ADD_ENTRY flag</td>
</tr>
<tr>
<td>2</td>
<td>Send the PCM opcode PCM_OP_WRITE_FLDS with the following flist:</td>
</tr>
<tr>
<td></td>
<td>0 PIN_FLD_POID POID [0] 0.0.0.1 /config/adjustment/event 301 0</td>
</tr>
<tr>
<td></td>
<td>0 PIN_FLD_EVENTS ARRAY [2] allocated 1, used 1</td>
</tr>
<tr>
<td></td>
<td>1 PIN_FLD_TYPE_STR STR [0] &quot;/event/activity/telco/gsm/ncc&quot;</td>
</tr>
<tr>
<td>3</td>
<td>Restart the BRM Connection Manager</td>
</tr>
</tbody>
</table>

As an alternative, you can use the BRM program `testnap` as follows:

```
/opt/portal/7.5/PortalDevKit/bin/testnap
r << XXX 1
0 PIN_FLD_POID POID [0] 0.0.0.1 /config/adjustment/event 301 0
0 PIN_FLD_EVENTS ARRAY [2] allocated 1, used 1
1 PIN_FLD_TYPE_STR STR [0] "/event/activity/telco/gsm/ncc"
XXX
#
# Perform the opcode
xop PCM_OP_WRITE_FLDS 32 1
```

The number 32 above is the decimal value of PCM_OPFLAG_ADD_ENTRY.

**Configuring NCC for BRM vouchers**

The Voucher Recharge feature node in the ACS Control Plan Editor typically controls voucher refunds. You can refund BRM accounts using either BRM or NCC vouchers.

To use BRM vouchers, the Set Active Domain feature node must have both Voucher and Wallet selected for the BCD domain type. For more information, see *Creating a BRM domain* (on page 14).

If you require EDRs (event detail records) to be sent to BRM when an attempt to redeem a voucher fails see *Tracking BRM Voucher Redemptions with Bad PINs* (on page 36).

**Tracking BRM voucher redemptions with bad PINs**

Each time a subscriber enters a bad PIN when trying to redeem a voucher on BRM, NCC can create a new event detail record (EDR) on BRM by sending a PCM_OP_ACTIVITY operation to track possible fraud.
To enable sending of these EDRs to BRM, set the \texttt{bcdActionHandler.BrmEdrObjectType} parameter in the NCC \texttt{eserv.config} file to match the subclass name of the \texttt{/event/activity} class. For example, if you will be sending EDRs to BRM when a subscriber attempts to redeem a voucher using a bad PIN, set the parameter to \texttt{/voucher}. Likewise, set \texttt{BrmBadPinResourceId} to the BRM resource ID that is used to count bad PIN attempts (the default ID is \texttt{1000011}) and set \texttt{BrmBadPinEdrActive} to \texttt{true}. The following example illustrates these settings:

\begin{verbatim}
bcdActionHandler = {
  ...
  BrmBadPinResourceId = 1000011
  BrmEdrObjectType = "/voucher"
  ...
  BrmBadPinEdrActive = True
  ...
}
\end{verbatim}

For information on creating BRM vouchers, see \textit{Creating BRM Vouchers} (on page 54).

\section*{Modifying the BCD Client Startup Script}

\subsection*{Procedure to modify the script}

When you install the BRM Charging Driver packages, the BCD Client start-up script (\texttt{bcdBeClient.sh}) is added in \texttt{/IN/service_packages/BCD/bin}.

You might want to create multiple copies of this script, for example, \texttt{bcdBeClient1.sh} and \texttt{bcdBeClient2.sh}, to improve performance by allowing each \texttt{bcdBeClient} process to run on a separate CPU or thread. If you do so, you must edit each \texttt{bcdBeClient}n.\texttt{sh} file to specify the correct names of the startup script and its log file. For example, if you change the name of the start-up script to \texttt{bcdBeClient1.sh}, you must also change the file's content to rename the \texttt{bcdBeClient} process \texttt{bcdBeClient1} and its log file \texttt{bcdBeClient1.log}.

The owner of the \texttt{bcdBeClient.sh} file is \texttt{acs Oper}. Follow these steps to modify the \texttt{bcdBeClient.sh} file:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Login to the SMS server as \texttt{acs Oper}.</td>
</tr>
<tr>
<td>2</td>
<td>Change directories to \texttt{/IN/service_packages/BCD/bin}.</td>
</tr>
<tr>
<td>3</td>
<td>Copy the existing start-up script and give the copy a new name. For example, \texttt{bcdBeClient2.sh}.</td>
</tr>
</tbody>
</table>
| 4    | Using a text editor such as vi or vim, open the new start-up script and locate the following lines:
\begin{verbatim}
#!/usr/bin/ksh
exec /IN/service_packages/BCD/bin/bcdBeClient >> /IN/service_packages/BCD/tmp/bcdBeClient.log 2>&1
\end{verbatim}
\| Change the names of the BCD Client start-up script and its corresponding log file to match the name you gave to the new start-up script file. For example:
\begin{verbatim}
#!/usr/bin/ksh
exec /IN/service_packages/BCD/bin/bcdBeClient2 >> /IN/service_packages/BCD/tmp/bcdBeClient2.log 2>&1
\end{verbatim}
| 6    | Save and close the file. |
| 7    | Repeat steps 3 to 6 for each additional start-up script that you want to create. |

\textbf{Note}: Given this example, you must also change the name of the original start-up script to \texttt{bcdBeClient1.sh} and the name of the process to \texttt{bcdBeClient1} and its log file to \texttt{bcdBeClient1.log}. 

Configuring the SLEE to run the scripts

The BRM Charging Driver package installation adds the following line to the file /IN/service_packages/SLEE/etc/SLEE.cfg:

```
INTERFACE=bcdBeClient bcdBeClient.sh /IN/service_packages/BCD/bin EVENT
```

This line tells the SLEE to run the BCD Client start-up script. If you create multiple start-up scripts, you must add an INTERFACE entry to the SLEE.cfg file for each script you create. For example, if you create scripts `bcdBeClient1.sh` and `bcdBeClient2.sh`, you must add the following lines to the SLEE.cfg file:

```
INTERFACE=bcdBeClient1 bcdBeClient1.sh /IN/service_packages/BCD/bin EVENT
```

The BRM Charging Driver installation package also adds the following line to the eserv.config file:

```
serviceDomainInterfaceName="bcdBeClient"
```

The value following the equal sign (bcdBeClient) defines the root name of the BCD Client start-up script.

**Note:** If you create multiple start-up scripts, you do not need to add the corresponding lines for the names of those scripts to the eserv.config file.

The names of the BCD Client in the INTERFACE parameters must be the same as the value of the serviceDomainInterfaceName parameter in the eserv.config file, appended with a number. The numbers must be sequential, beginning with 1 - for example, `bcdBeClient1`, `bcdBeClient2`, and so on.

The SLEE will distribute the load evenly across the specified BCD Client processes. If you leave a gap in the numbering, the SLEE will only distribute the load across the number of processes prior to the gap.

Starting the BCD Client processes

Start the SLEE to initiate the bcdBeClient processes that are defined in the new BCD Client start-up scripts, as well as the other SLEE applications. For information on stopping and starting the SLEE, see Service Logic Execution Environment Technical Guide.

Creating Balance Type Mappings

About Creating Balance Type Mappings

You must create balance types to match the names of the BRM resource IDs for all the resources that are used by both NCC and BRM.

NCC has two concepts with regard to account balances: currency and balance type. For example, Euros would be a currency and General Cash might be a balance type. BRM, however, uses only one concept, which is a resource ID; for example, Euros has a resource ID of 978 and free time has a resource ID of 1000076.

When NCC queries BRM for a balance, it expects both the currency and the balance type.

**Note:** On NCC a balance of $50 means the subscriber has $50 to spend and a balance of -$10 means the subscriber owes $10. On BRM, a balance of -$50 means the subscriber has $50 to spend and a balance of $10 means the subscriber owes $10.

The BRM Charging Driver changes the sign when it presents balances to NCC users. For example, a BRM balance of -$50 will be shown in a text message as $50. However, the balance that appears in an flist for an operation is shown as it exists on BRM because the BRM Charging Driver calls a BRM print function to display it.
The BRM Charging Driver determines the currency by looking up the NCC currency code in the SMS UI. You can access the Currency Code tab in the Service Management window by logging in to SMS and selecting Prepaid Charging from the Services menu, and then Service Management to display the following window:

```
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOP</td>
<td>Dominican Republic, Peso (DOP)</td>
</tr>
<tr>
<td>DZD</td>
<td>Algeria, Algerian Dinar (DZD)</td>
</tr>
<tr>
<td>EEC</td>
<td>Estonia, Kroon (EEK)</td>
</tr>
<tr>
<td>EGP</td>
<td>Egypt, Pound (EGP)</td>
</tr>
<tr>
<td>ERN</td>
<td>Eritrea, Nafka (ERN)</td>
</tr>
<tr>
<td>ESP</td>
<td>American Samoa, Spanish Peseta (ESP)</td>
</tr>
<tr>
<td>ETB</td>
<td>Ethiopia, Birr (ETB)</td>
</tr>
<tr>
<td>EUR</td>
<td>Cyprus, Euro (EUR)</td>
</tr>
<tr>
<td>EUR</td>
<td>Dutch (The Netherlands) Euro (EUR)</td>
</tr>
<tr>
<td>EUR</td>
<td>Eire (Ireland), Euro (EUR)</td>
</tr>
<tr>
<td>EUR</td>
<td>European EMU, Euro (EUR)</td>
</tr>
<tr>
<td>EUR</td>
<td>Finland, Euro (EUR)</td>
</tr>
<tr>
<td>EUR</td>
<td>France, Euro (EUR)</td>
</tr>
<tr>
<td>EUR</td>
<td>French Guiana, Euro (EUR)</td>
</tr>
</tbody>
</table>
```

NCC then obtains the BRM resource ID for that currency by looking up the currency code in the `BrmToNccCurrencyMapping` array in the `eserv.config` file, a sample of which is given here, to:

```python
BrmToNccCurrencyMapping = [
    {  
        NCCCode = "NZD"
        BRMNum = 554  
    },
    {  
        NCCCode = "EUR"
        BRMNum = 978  
    }
]
```
For the balance type, NCC relies on data that has been entered through the SMS Wallet Management screens, which you can access by selecting Prepaid Charging from the Services menu, and then Wallet Management. On the Balance Types tab, you must define each NCC balance type and each BRM resource ID. The following figure illustrates these entries:

The relevant NCC balance types shown are Free Time and General Cash while the relevant BRM resource IDs in this example are 978 and 1000076. Note that for BRM, you specify the resource ID as the balance type name.

To create balance types, login to the Service Management System, select Prepaid Charging from the Services menu, and then select Wallet Management. For more information on creating balance types, see the Charging Control Services User’s Guide.

Next, you must create balance type mappings to associate the NCC balance types with the BRM resource IDs and specify any scaling that is required between the two types.
Create balance type mappings in the Service Management window of the SMS UI. To access the Service Management window, login to the Service Management System, select Prepaid Charging from the Services menu, and select the Service Management menu item and then the Balance Type Mapping tab on the Service Management screen. The following figure illustrates the balance type mapping entries.

The first line maps the NCC General Cash balance type to the BRM resource ID 978, which is Euros. NCC balances are always in small units, which in this case is Euro cents, so the scaling factor is 100.

The second line maps the NCC balance type Free Time to the BRM resource ID 1000076. All time balances in NCC are in hundredths of seconds while BRM minutes are actually in minutes. Because there are 6000 hundredths of a second in a minute, the scaling factor is 6000.

The BRM resource IDs for currency codes are defined in the BRM Resource Center, which is a component of Pricing Center.

For the specific steps to create balance type mappings in the SMS UI, see NCC Charging Control Services User's Guide.
Configuring BRM for the BRM Charging Driver

About Configuring BRM

Introduction

This chapter describes the BRM configuration tasks that you must perform to integrate BRM with NCC.

To integrate BRM with NCC, you must create custom fields and storable classes. You can also create subclasses of existing classes. This section describes the custom fields and storable classes that you must create to integrate BRM with NCC. For more detailed descriptions of the steps to create custom fields and storable classes, see Oracle Communications Billing and Revenue Management Developer's Guide.

Note: Before you can create BRM custom fields and storable classes for NCC, you must edit the Data Manager configuration file, pin.conf, to make the data dictionary writable. See the section on modifying the pin.conf file in the BRM Developer's Guide, and make sure you restore the settings to make the data dictionary unwritable after you are finished.

The BRM documentation describes how to perform the particular steps that are required by some of the tasks presented in this section.

In this chapter

This chapter contains the following topics.

Adding Custom Fields 43
Adding Storable Classes 45
Creating Header and Library Files for the Custom Classes 46
Generating the Custom JAR File 46
Modifying the BRM Configuration Files 47
Configuring BRM for ExtendTimeReservation 48

Adding Custom Fields

About adding custom fields

Creating a custom field in BRM allows you to add a tag and value pair that BRM can add to records that it produces. For example, you can use the NCC Set BE EDR feature node in a control plan to add an arbitrary tag and value pair to an event data record. To enable this capability in BRM, you must create a custom field and configure BRM to add that field to the records it produces.

You must also create the mapping between any NCC information items and the corresponding BRM custom fields by adding the appropriate values to the bcdActionHandler.NccToBrmFieldMapping list in the NCC eserv.config file. For more information on creating the mapping from NCC to BRM fields, see Mapping NCC Information to BRM Fields (on page 19).
You add the custom fields PIN_FLD_NCC_INFO and PIN_FLD_NCC_FIELD to make them available for the storable classes that you will define. Adding PIN_FLD_NCC_INFO creates a container for putting other NCC fields in when NCC sends operations to BRM. The PIN_FLD_NCC_FIELD is required because the PIN_FLD_NCC_INFO container must not be empty when NCC sends an opcode to BRM, but for some operations there is nothing useful to put in the container.

**Note:** When you assign an ID to a custom field, the ID must be greater or equal to 10000 and must not already be in use. Some customization of BRM for other purposes might have already occurred at the time you configure it for integration with NCC, so you cannot assume that ID 10000 will be free.

For examples of NCC operations (opcodes) that include the custom fields you define, see the sections containing messages in *Usage Scenarios* (on page 65).

Use the BRM Storable Class Editor in the BRM Developer Center to create the following fields:

**PIN_FLD_NCC_INFO**

Name: PIN_FLD_NCC_INFO  
Type: PIN_FLD_SUBSTRUCT  
Description: NCC substruct containing custom fields  
Default Field ID: 10000

**Note:** You must set `bcdActionHandler.NccInfoFieldNumber` in the NCC `eserv.config` file to the assigned field ID. The default ID of 10000 is the ID assigned in the default `eserv.config` file during installation. If you need to use a different ID, you must change it in the `eserv.config` file as well.

**PIN_FLD_NCC_FIELD**

Name: PIN_FLD_NCC_FIELD  
Type: PIN_FLDT_STR  
Default Field ID: 10001  
Description: Dummy field required to keep the PIN_FLD_NCC_INFO substruct from being empty when NCC sends an opcode to BRM.

**Note:** You must set the corresponding `bcdActionHandler.NccInfoFieldDummyEntry` value in the NCC `eserv.config` file to the assigned field ID. The default ID of 10001 is the ID assigned in the default `eserv.config` file during installation. If you need to use a different ID, you must change it in the `eserv.config` file as well.

**PIN_FLD_LOCATION**

GSM networks pass NCC a location number, which gives an indication of where the caller is. It does not identify an individual phone, but does indicate where the caller is. For example, the location number for Ipswich in the United Kingdom is 00441473, where 00 is the international dialing prefix, 44 is the country code for the United Kingdom, and 1473 is the area code for Ipswich.

If you want to support location numbers, you must create a location field or use an existing one.

**Note:** If you will not be sending location numbers to BRM, set `bcdActionHandler.LocationNumberMapping.BRMField` in the NCC `eserv.config` file to 0. Otherwise, set `bcdActionHandler.LocationNumberMapping.BRMField` to the assigned field ID.

The ID 1251 is the pre-defined BRM field ID of PIN_FLD_LOCATION. If LocationNumberMapping.BRMField is some other number, you must set the Default Field ID to that number. For more information, see *Mapping the Location Number* (on page 21).

**Name:** PIN_FLD_LOCATION  
**Type:** PIN_FLDT_STR  
**Default Field ID:** 1251
Creating additional custom fields

Create additional custom fields for other information items that you wish to support. For any additional fields that you define, you must also create the mapping between NCC and BRM fields by adding the appropriate values to the `bcdActionHandler.NccToBrmFieldMapping` list in the NCC `eserv.config` file. Each entry should contain the following parameters:

- **NCCItem** - The text name of the NCC concept
- **BRMType** - "STRING, "DECIMAL", "INT", "ENUM", or "TIMESTAMP"
- **BRMField** - The field ID of the custom field

The following custom field definitions are provided as examples.

**PIN_FLD_NCC_NUMBER_OF_EVENTS**

Add the `PIN_FLD_NCC_NUMBER_OF_EVENTS` field if you wish to support named events such as SMS, MMS, or email.

- **Name**: `PIN_FLD_NCC_NUMBER_OF_EVENTS`
- **Type**: `PIN_FLD_T_INT`

Add the following values to the `bcdActionHandler.NccToBrmFieldMapping` list:

- **NCCItem**: "NUMBER_OF_EVENTS_ID"
- **BRMType**: "INT"

**PIN_FLD_NCC_TARIFF_PLAN_ID**

Add the `PIN_FLD_NCC_TARIFF_PLAN_ID` field if you wish to support a tariff plan ID.

- **Name**: `PIN_FLD_NCC_TARIFF_PLAN_ID`
- **Type**: `PIN_FLD_T_INT`

Add the following values to the `bcdActionHandler.NccToBrmFieldMapping` list:

- **NCCItem**: "TARIFF_PLAN_ID"
- **BRMType**: "INT"

Adding Storable Classes

Creating custom classes

The BRM data dictionary uses storable classes to define various types of data. There are storable classes, for example, that define an account, a service object, an activity event, and a session event. To store NCC activities, you must create custom fields and storable classes to define those activities for the BRM data dictionary.

**Note**: Before you can change or add storable classes for NCC, you must edit the Data Manager configuration file, `pin.conf`, to make the data dictionary writable. See the section on modifying the `pin.conf` file in the BRM Developer's Guide, and be sure you restore the settings to make the data dictionary unwritable when you are finished.

The `/event/activity/telco/gsm` class is the standard BRM class for handling GSM mobile calls and the `/active_session/telco/gsm` is a sister class necessary for doing real-time charging. You must create subclasses for both classes so you can add the custom fields to them.

Use the BRM Storable Class Editor, which is part of the BRM Developer Center, to create an NCC subclass for each of the following classes:

- `/active_session/telco/gsm`
The following example illustrates the appropriate subclasses:

```
/event/activity/telco/gsm
/active_session/telco/gsm/ncc
/event/activity/telco/gsm/ncc
```

Using the Storable Class Editor, add the custom fields PIN_FLD_NCC_INFO and PIN_FLD_NCC_FIELD, as well as any others that you have defined, to each of these subclasses. For more information, see the section on creating custom fields and storable classes in *Oracle Communications Billing and Revenue Management Developer's Guide*.

If all of the products, plans and deals that you will define in BRM Pricing Center refer to `/event/activity/telco/gsm`, then extending these two classes is sufficient. If your products, plans, and deals refer to other classes, then you must create subclasses for those classes as well, and add the NCC custom fields to them.

## Creating Header and Library Files for the Custom Classes

### Creating the custom header and library files

Follow these steps to create the BRM header and library files required to make your NCC custom fields available to BRM applications.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Use the BRM Developer Center to generate a header file, for example <code>ncc_flds.h</code>, for the custom NCC fields that you added.</td>
</tr>
<tr>
<td>2</td>
<td>Copy the header file to the appropriate location.</td>
</tr>
<tr>
<td>3</td>
<td>Create a library file from the header file.</td>
</tr>
<tr>
<td>4</td>
<td>Copy the library file to the appropriate location and make it available to your applications.</td>
</tr>
<tr>
<td>5</td>
<td>Restart processes.</td>
</tr>
</tbody>
</table>

See the section on making custom fields available to your application in the *BRM Developer's Guide* for the specific directions to accomplish these general steps.

## Generating the Custom JAR File

### Steps to generate JAR file

In addition to creating the custom header and library files, you must generate a custom `.jar` file for the custom fields and storable classes and configure it for BRM client application such as BRM Developer Center, Pricing Center, or Customer Center.

You created the necessary `.java` files and specified a location for them when you generated the header file for the custom fields in Developer Center.

For information how to compile the `.java` files create the custom `.jar` file, see the section on using custom fields in Java applications in *BRM Developer's Guide*. These are the general steps:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Compile the <code>.java</code> files.</td>
</tr>
<tr>
<td>2</td>
<td>Move the compiled <code>.class</code> files to a directory.</td>
</tr>
<tr>
<td>3</td>
<td>Create a Java archive (.jar) file.</td>
</tr>
<tr>
<td>4</td>
<td>Copy the <code>.jar</code> file into a directory that client applications can access</td>
</tr>
<tr>
<td>5</td>
<td>Edit the Developer Center start-up script, which is platform specific, to include the custom <code>.jar</code> file.</td>
</tr>
</tbody>
</table>
### Modifying the BRM Configuration Files

#### Steps to modify BRM configuration files

To integrate BRM with NCC, you must add entries to the following BRM configuration files: `pin_event_map`, `pin_rum`, and `pin_config_reservation_aaaPrefs`.

The `pin_event_map` file configures the mapping of a service type with an event type. You must add this mapping for any custom service and event object types. This enables BRM to associate the price offers for a specific service with the given event types. As delivered, the `pin_event_map` file contains the mapping for the default service and event types that are supported. For more information, see the section on mapping event types to services in *BRM Setting Up Pricing and Rating*.

The event data that you use to rate an event is called ratable usage metrics. Common examples of ratable usage metrics are duration, in which you rate based on the length of time an event lasts, and occurrence, in which you rate based on the number of events that occur, regardless of their duration. The `pin_rum` file specifies the ratable usage metrics for an event type. For information on specifying ratable usage metrics, see the section on setting up ratable usage metrics in *BRM Setting up Pricing and Rating*.

For information on the `pin_config_reservation_aaaPrefs` file, see the section on specifying default AAA preferences in *BRM Telco Integration*.

Follow these steps to modify the BRM configuration files for integration with NCC.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Add the new custom fields generated by Developer Center to the <code>Infranet.properties</code> file of the relevant BRM client applications. <code>Infranet.properties</code> is a configuration file used by BRM client applications such as Pricing Center, Developer Center and Customer Center. The <code>Infranet.properties</code> file must be changed on every computer on which the client applications run.</td>
</tr>
</tbody>
</table>

#### Step 1

If they are not already present, add the BRM events that you will use with NCC to your `pin_event_map` file, for example, `pin_event_map_telco__gsm`. For the example products used with these instructions, you would add the following events:

- `/service/telco/gsm` : `/event/session/telco/gsm` : Real Time Telco GSM Service
- `/service/telco` : `/event/activity/telco/gsm/ncc` : Real Time Telco GSM NCC Activity

#### Step 2

Load the `pin_event_map` file, following instructions for the `load_pin_event` utility in *BRM Setting Up Pricing and Rating*.

#### Step 3

If you require support for named events, add the following line to the `pin_rum` file, substituting your event class, if it is different than the one shown:

```
/event/activity/telco/gsm/ncc : Number Of Events : PIN_FLD_NCC_INFO.PIN_FLD_NCC_NUMBER_OF_EVENTS : none
```

#### Step 4

Load the `pin_rum` file, following instructions for the `load_pin_rum` utility in *BRM Setting Up Pricing and Rating*. 
If you require support for data calls, modify the `pin_config_reservation_aaa_prefs` file, for example `pin_config_reservation_aaa_prefs_gsm_data`, multiplying appropriate QUANTITY fields by the scaling factor. The scaling factor is defined in the ScalingFactor parameter of the data call entry in the `bcdActionHandler.ServiceProfileTagMapping` list in the NCC `eserv.config` file on the SLC. For more information on the scaling factor, see *Mapping NCC Sessions to BRM Services* (on page 16).

For example, modify the default `pin_config_reservation_aaa_prefs_gsm_data` file to match the scaling factor:

```
0 PIN_FLD_RESERVATION_INFO ARRAY [0]
1 PIN_FLD_QUANTITY DECIMAL [0] <50 x scaling factor>
1 PIN_FLD_MIN_QUANTITY DECIMAL [0] 0
1 PIN_FLD_INCR_QUANTITY DECIMAL [0] <50 x scaling factor>
1 PIN_FLD_RUM_NAME STR [0] "Size"
1 PIN_FLD_REQ_MODE ENUM [0] 4
1 PIN_FLD_UNIT ENUM [0] 0
0 PIN_FLD_RESERVATION_INFO ARRAY [1]
1 PIN_FLD_QUANTITY DECIMAL [0] 50
1 PIN_FLD_MIN_QUANTITY DECIMAL [0] 0
1 PIN_FLD_INCR_QUANTITY DECIMAL [0] 50
1 PIN_FLD_RUM_NAME STR [0] "Amount"
1 PIN_FLD_REQ_MODE ENUM [0] 1
1 PIN_FLD_UNIT ENUM [0] 0
```

6. Load the `pin_config_reservation_aaa_prefs` file following instructions in *BRM Telco Integration* for `load_config_reservation_aaa_prefs`. See the section on telco integration utilities.

7. Restart the BRM Connection Manager.

For more information about modifying these configuration files, see *BRM Setting Up Pricing and Rating*.

### Configuring BRM for ExtendTimeReservation

#### Configuring BRM for ExtendTimeReservation

To configure data charging to work correctly, you must set `PIN_FLD_RATING_MODE` to `USED_AGGR_MODE` in `PCM_OP_TCF_AAA_AUTHORIZE` for ExtendTimeReservation.

Follow these steps to set `PIN_FLD_RATING_MODE` to `USED_AGGR_MODE` in `PCM_OP_TCF_AAA_AUTHORIZE`:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Log on to BRM as the user runs the Connection Manager.</td>
</tr>
<tr>
<td>2</td>
<td>Change directory location to <code>$PIN_HOME/sys/data/config</code>.</td>
</tr>
<tr>
<td>3</td>
<td>Edit the <code>pin_config_opcodemap.tcf</code> file and change the following lines to comments:</td>
</tr>
</tbody>
</table>

```
Framework Opcode: PCM_OP_TELCO_UPDATE_AND_REAUTHOR
Processing Stage: POST_PROCESS
Opcode_Map:/service/telco/gsm/telephony,PCM_OP_GSM_AAA_POL_POST_PROCESS
Opcode_Map:/service/telco/gsm/sms,PCM_OP_GSM_AAA_POL_POST_PROCESS
Opcode_Map:/service/telco/gsm/data,PCM_OP_GSM_AAA_POL_POST_PROCESS
Opcode_Map:/service/telco/gsm/fax, PCM_OP_GSM_AAA_POL_POST_PROCESS
```
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 4    | Run the following command to load the `pin_config_opcodemap_tcf` file:  
      `../../../bin/load_aaa_config_opcodemap_tcf -vf pin_config_opcodemap_tcf` |
| 5    | Restart the Configuration Manager. |
Creating Products and Deals

About Creating Products and Deals

Introduction

This chapter describes how to create BRM products, plans and deals to set up rating and charging on BRM for the features that you require. Add the relevant products, plans and deals and assign the plans to your customers. Plans, for example, allow you to specify charges for local and international voice calls during peak and off peak hours, as well as charges for data and short messages, and discounts for friends and family, and so on.

The following sections provide examples of products and deals for a sampling of features. In general, you can set up the products however you like as long as you create subclasses for the classes referred to by the products and add the NCC custom fields. You must also match the corresponding NCC parameters to the product values as follows:

- The NCC BrmServicePoid parameter must match the product value “Applies to”.
- The NCC BrmObjectType parameter must match the product value “Event”.
- The NCC BrmReqMode parameter must match the product value “Measured by”.

The service and event names used are only examples. The examples, however, are intended to be sensible and usable.

In this chapter

This chapter contains the following topics.

Creating a Product and Deal for Voice Calls
Creating a Product and Deal for Data Calls
Creating a Product and Deal for Named Events
Creating BRM Vouchers
Creating a Customer

Creating a Product and Deal for Voice Calls

Creating the BRM product and deal for voice calls

Using the Pricing Center on BRM, create a product for voice calls with the following properties:

- Applies to: /service/telco/gsm/telephony
  - On NCC, this service is defined in the BrmServicePoid parameter for the voice call entry in the eserv.config file. For example:
    - BrmServicePoid = "/service/telco/gsm/telephony"
- Event set to /event/session/telco/gsm
  - On NCC, the BrmObjectType parameter for the voice call entry must match the suffix of the BRM event that follows the /event/session/telco prefix. In this example BrmObjectType = "gsm".
- Measured by: Duration
  - On NCC, the BRMReqMode parameter in the eserv.config file must be set to "DURATION".
Create a deal for this product and a plan containing the relevant deals.

**Creating a Product and Deal for Data Calls**

**Creating the BRM product and deal for data calls**

This is a real time telco GSM session that is measured by Volume. On BRM, create a product with the following properties:

- Applies to: `/service/telco/gsm/data`
  
  On NCC, this service is defined in the BrmServicePoid parameter for the data call entry. In this case, for example:
  
  \[BrmServicePoid = "/service/telco/gsm/data"\]

- Event set to `/event/session/telco/gsm`
  
  On NCC, the BrmObjectType parameter for the data call must match the suffix of the BRM event that follows the `/event/session/telco` prefix. In this example, `BrmObjectType = "gsm"`

- Measured by: Volume
  
  On NCC, BrmReqMode parameter must be set to "VOLUME"

Create a deal for this product and a plan containing the relevant deals.

**Creating a Product and Deal for Named Events**

**About creating products and deals for direct named events**

NCC uses a concept called *named event* that allows you to name events such as the sending of a short message that costs fifty cents or an event for purchasing the Friends and Family service for thirty dollars. You can define these using the SMS UI by selecting Prepaid Charging from the Services menu, then Rating Management, and then the Named Event tab.

You can define several sets of events and several events within each set. Each set and each event have names. When you create an event you also specify an event type: Direct Event, Reserve Event, or Any. In our example, assume that the short message event is of type Any and the purchase of the Friends and Family service is of type Direct Event.

To use named events, use the Named Event feature node in a control plan and select the class name, the event name, and one of the following named event types: Direct Event, Reserve Event, Confirm Event, Revoke Event, or Cost of Event.

To charge for a short message, you can use the Named Event feature node to send a Reserve Event message when NCC attempts to send the message and a Confirm Event when the message has been delivered. When used with the BRM Charging Driver, the Reserve Event type sends a `PCM_OP_TCF_AAA_AUTHORIZE` operation and the Confirm Event type sends a `PCM_OP_TCF_AAA_STOP_ACCOUNTING` operation.

Alternatively, if you want to charge immediately for the message, without knowing whether it will be delivered, you can choose a Direct Event named event, which is allowed for named event types of Direct Event or Any. In the example, because the named event type is Any, this is a reservable direct named event. In this case, the Named Event feature node sends a `PCM_OP_TCF_AAA_AUTHORIZE` operation immediately followed by a `PCM_OP_TCF_AAA_STOP_ACCOUNTING` operation.

To purchase the Friends and Family service, set the named event type to Direct Event and select the Friends and Family event. Because the event type is Direct Event, the Named Event feature node sends a `PCM_OP_PURCHASE_DEAL` operation.

For examples of the messages sent for direct named events, see Messages: Direct Named Event Reservation and Messages: Direct Named Event for Non-Reservable Event
Configuring BRM for named event reservations and reservable direct named events

On BRM, create a product with the following properties:

- **Applies to: /service/telco/gsm/sms**
  
  On NCC, the prefix of this service, which is common to all named events, is defined in the `bcdActionHandler.poidPrefix` parameter as "/service/telco/".
  
  The NCC named event set, or billable class, must match the suffix of the BRM service that follows the prefix, which in this case is **gsm/sms**.

- **Event: /event/activity/telco/gsm/ncc**
  
  The NCC billable event name must match the suffix of the BRM event that follows the `/event/activity/telco` prefix. In this example, the named event name is **gsm/ncc**. In this example, `BrmObjectType = "gsm/ncc"`. **Note:**

- **Measured by: Number of Events**
  
  On NCC, the `BRMReqMode` parameter must be set to "NUMBER OF EVENTS". The rate plan should be positive.

You can rate a named event without charging the subscriber's account by using the Named Event feature node with Named Event Feature Selection set to Cost of Event. To support this feature, the BRM product must include the following additional event:

- **Event: /event/activity/gsm/ncc**
  
  Measured by: Occurrence. The rate plan amount must match the amount assigned to the session event described above.

Create a deal for this product and a plan containing the relevant deals.

Configuring named event refunds

Named event refunds are named events that are controlled by a Named Event feature node. A refund is accomplished by creating an event of type **SMS Refund** and submitting one **SMS Refund** event.

In addition to the steps provided here, see also *Enabling refunds for direct named events*.

Create a BRM product with the following properties:

- **Applies to: /service/telco/gsm/smsRefund**
  
  **Note:**: The service must be different than the service used by ordinary named events. You can also configure a custom class, such as `/service/telco/gsm/sms/Refund`.

  The prefix of this service, which is common to all named events, is defined in the `bcdActionHandler.poidPrefix` parameter in the NCC `eserv.config` file. In this example, the value of `poidPrefix` is `/service/telco`.

  The NCC named event set, or event class, must match the suffix of the BRM service following the prefix. For example, if you use the `smsRefund` service, the named event set must be `gsm/smsRefund`.

- **Event: /event/session/telco/gsm/ncc**
  
  Measured by: Number of Events

  The NCC named event name must match the suffix of the BRM event following the `/event/session/telco` prefix. In this example, the named event name must be `gsm/ncc`.

  When you create the NCC named event, set Event Type to Reservable Event, which allows only named event reservations, or Any, which allows both named event reservations and reservable direct named events.

  The rate plan amount must be negative.

Create a deal for this product and a plan containing the relevant deals.
Configuring non-reservable direct named events

Non-reservable direct named events are events that are controlled by a Named Event feature node with Named Feature Selection set to Direct Event. When creating these events in the feature node, set Event Type to Direct Event.

Create a BRM product with the following properties:

- Applies to: Account
- Event: Purchase Fee Event
  The NCC named event name must match the name of the BRM deal to be purchased.
- Measured by: Occurrence

Create a deal for this product, but do not add the deal to the plan in the same manner as other types of named events. This deal is purchased directly by the named event chassis action.

Creating BRM Vouchers

Configuring BRM for vouchers

The Voucher Recharge feature node in the NCC Control Plan Editor typically controls voucher refunds. You can refund BRM accounts using either BRM or NCC vouchers.

To use BRM vouchers, the NCC Set Active Domain feature node must have both Voucher and Wallet enabled for the BCD domain. For more information, see Creating a BRM domain (on page 14).

If you require EDRs (event detail records) to be sent to BRM when an attempt to redeem a voucher fails see Tracking BRM Voucher Redemptions with Bad PINs (on page 36).

The following deal is required only for BRM vouchers.

Using Pricing Center on BRM, create a product with the following properties:

- Applies to: /service/telco/gsm/account
  On NCC, this service is defined in the BrmServicePoid parameter for the voucher entry. In this case, for example:
  BrmServicePoid = "/service/telco/gsm/account"
- Event set to /event/session/telco/purchase fee event
  On NCC, the BrmObjectType parameter for the data call must match the suffix of the BRM event that follows the /event/session/telco prefix. In this example BrmObjectType = "purchase fee event"
- Measured by: Occurrence
  On NCC, BrmReqMode parameter must be set to "OCCURRENCE".

The rate plan amount must be negative.

Create a deal for this product. Generate BRM vouchers in the BRM Voucher Administration Center using the selected deal.

Creating a Customer

Creating a customer

For any NCC subscriber who will use BRM for rating and charging, you must create a customer record for the subscriber on BRM.

Before creating a customer on BRM, you must select the BRM Charging Driver domain (for example, BCD) when you create the subscriber's wallet. For information on creating a subscriber's wallet, see Charging Control Services User's Guide.
For information on creating a new BRM customer using the BRM Customer Center, see *BRM Managing Customers* and BRM Customer Center Help. Use the following steps as a general guide:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select the plan containing the deals that you defined in <em>Creating Products and Deals</em> (on page 51).</td>
</tr>
<tr>
<td>2</td>
<td>Assign a number and SIM to each service that is included in the plan.</td>
</tr>
</tbody>
</table>
| 3    | If you require support for vouchers, create a separate BRM customer for the Voucher Administration Center.  
    - Select CSR Plan  
    - Assign an ID and password.  
      You login to Voucher Administration Center using this name and password when generating vouchers. |

In practice, you might want to add NCC subscribers in bulk to the BRM database. For information on mapping data from another system to create new BRM customers in bulk, see *BRM Managing Customers*. 
Chapter 6

Using In-Session Notifications with BRM

About Using In-Session Notifications

Introduction

This chapter explains how to enable and configure BRM to use in-session notifications in conjunction with NCC. This chapter should be read in conjunction with the BRM documentation. You can find BRM documentation on in-session notifications in Oracle Communications Billing and Revenue Management Telco Integration.

For information on configuring NCC for in-session notifications, see Mapping BRM Piggyback Notifications to NCC Profile Tags (on page 22).

In this chapter

This chapter contains the following topics.

- Enabling In-Session Notifications
- Configuring In-Session Notifications
- Using SubscriptionExpiry, Credit Threshold, and Streaming Threshold Notifications

Enabling In-Session Notifications

Prerequisite

The BRM client and server software must be at release 7.5, patch set 4 or greater.

Setting the PiggyBack parameter

Follow these steps to enable in-session notifications in the BRM server business parameters by enabling the PiggyBack parameter:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Edit the following file on the BRM server: $PIN_HOME/sys/data/config/bus_params_AAA.xml</td>
</tr>
<tr>
<td>2</td>
<td>Set the Piggyback parameter to enabled.</td>
</tr>
<tr>
<td>3</td>
<td>Run the following command: pin_bus_params bus_params_AAA.xml</td>
</tr>
</tbody>
</table>

For more information, see "Providing In-Session Notifications for Network Connectivity Applications" in Oracle Communications Billing and Revenue Management Telco Integration.
Loading the Subscriber Profile

To enable in-session notifications in BRM, you must load the `config_subscriber_preferences_map.xml` file into the database. To do so, run the following command in the directory

```
$PIN_HOME/apps/load_config:
load_config -dv
$PIN_HOME/sys/data/config/config_subscriber_preferences_map.xml
```

When you create a new customer account, BRM uses a default set of preferences to create a `profile/subscriber_preferences` object. If no profile object is created, no piggyback notifications will be sent.

**Note:** To create a customer, you must use Customer Center from release 7.5, patch set 1 or later.

Setting the SubscriberLifeCycle parameter

Follow these steps to set the `SubscriberLifeCycle` BRM business parameter, which is required to use SubscriptionExpiry notifications:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | Edit the following file on the BRM server:  
      | $PIN_HOME/sys/data/config bus_params_customer.xml |
| 2    | Set the SubscriberLifeCycle parameter to enabled, as follows:  
      | `<SubscriberLifeCycle>enabled</SubscriberLifeCycle>` |
| 3    | Run the following command:  
      | `pin_bus_params bus_params_customer.xml` |

Restarting BRM

Run the following commands to restart BRM:

```
stop_all
start_all
```

Configuring In-Session Notifications

Configuring the Subscriber in the Customer Center

To configure subscribers to use in-session notifications, you must use Customer Center release 7.5, patch set 1 or later. To assign SIM and IMSI numbers, you must also install the GSM_Mgr_CustCirExt client extension packages.

You configure in-session notifications per subscriber.

Follow these steps in the Customer Center to configure a subscriber for in-session notifications:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Edit an existing subscriber or create a new one.</td>
</tr>
<tr>
<td>2</td>
<td>Select the Subscriber Preferences tab.</td>
</tr>
<tr>
<td>3</td>
<td>Select the Enabled checkbox for the account or service to display the Preference Details panel.</td>
</tr>
</tbody>
</table>
### Configuring Credit Threshold Notifications

Follow these steps to configure Credit Threshold notifications in the relevant Pricing Center Plan.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In the <strong>Track Balances</strong> tab, select a service.</td>
</tr>
<tr>
<td>2</td>
<td>In <strong>Credit Limits</strong>, choose <strong>Add</strong> or <strong>Modify</strong> a resource.</td>
</tr>
<tr>
<td>3</td>
<td>Set the following notifications in the Credit Limit panel, as desired:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Credit Limit/Floor</strong> - For a purely prepaid service, set Limit to 0.0 and Floor to the maximum topup value, specified as a negative value. For a purely postpaid service, set Floor to 0.0 and set Limit to the maximum credit limit, specified as a positive value.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Percentage Threshold Settings</strong> - Set the Credit Threshold breach points expressed as a percentage of the range between the configured Credit Floor and Limit. The meaning of the percentage is different for prepaid and postpaid scenarios. For example, a 90% breach on a prepaid service means that the remaining topup balance is less than 10% of the range between the Limit and the Floor. For a postpaid service, it means that the used credit has exceeded 90% of the range between the Limit and the Floor.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Fixed Threshold Settings</strong> - Sets the Credit Threshold breach points to an actual resource amount, which is a negative value for a prepaid service.</td>
</tr>
</tbody>
</table>

Once you've set the desired notifications, commit them to the database. To be able to view and select new Plans in the Customer Center, they must be added to the CSR Plan List.
Credit Threshold notification settings are only picked up for a subscriber when the Plan is assigned to the subscriber. Subsequent changes to the settings in the Plan are not propagated to the subscriber. Beware, too, that pricing data is cached in Customer Center so it might be necessary to restart Customer Center to pick up changes.

Note: The currency in the Plan and the primary currency in the subscriber account must be the same.

Using SubscriptionExpiry, Credit Threshold, and Streaming Threshold Notifications

Using SubscriptionExpiry Notifications

Note the following considerations in using SubscriptionExpiry Notifications:

- BRM calculates the subscription expiry date from the service lifecycle model (SLM) that was used when the subscriber was created. The SLM sets this value in the \service object. When a new service is activated, SLM sets the \EXP_T field based on the validity of the service and the service state. See "Managing Service Lifecycles" in Oracle Communications Billing and Revenue Management Managing Customers for more information.
- The subscription expiry date is held in the \service object from the PIN_FLD_SERVICE_EXPIRATION field. If the service life cycle has not been enabled, this field will not be set and expiry notifications will not be generated.
- BRM uses its own virtual time to calculate expiry events. To set the BRM virtual time, use the pin_virtual_time command. For example, the following command sets the virtual date and time to Thursday, February 28, 2013 and 03:58:20 using the format MMDDHHMMYYYY.SS:
  
  ```
  pin_virtual_time -m 2 022803582013.20
  ```

Using Credit Threshold Notifications

Note the following considerations in using Credit Threshold notifications:

- For piggyback notifications, BRM sets ALERT_TYPE to 0x0, REASON to 0x1 for a breach up, which implies increased use of the resource, and 0x2 for a breach down.
- BRM might send a Credit Threshold notification in a STOP_ACCOUNTING response that duplicates information in previous AUTHORIZE or UPDATE_AND_REAUTHORIZE messages for a call. The reason is that duplicates would be filtered out in the case of AUTHORIZE, REAUTHORIZE, and UPDATE_AND_REAUTHORIZE messages but not in the case of STOP_ACCOUNTING because STOP_ACCOUNTING is also sent in the case of direct debit. The breach notification can be different depending on the snapshot of the current balance, which will be different in the case of parallel sessions.

Using Streaming Threshold Notifications

Note the following considerations in using Streaming Threshold notifications:

- Use Streaming Threshold notifications like /gsm/data or gprs services. Set the resource ID to a non-currency counter resource in the /profile/subscriber_preferences when creating the account. This is used to track the streaming threshold breaches. For example, if a streaming threshold resource is set to MBytesUsed(1000009) and the streaming thresholds value is specified in megabytes, and a notification for each 1 gigabyte (GB) of data is required, the streaming threshold value 1024 would be configured.
• A full rate plan needs to contain a currency balance or a chargeable balance. For example, a simple case might have a currency resource ID of Euro (978), with the scaled amount set to 1 euro for 1 MB of usage, and a counter resource ID of MBytesUsed (1000009) with the scaled amount set to 1 for 1 megabyte of usage, which implies that this counter gives the total megabytes used in a session. Note that Credit Floor/Limit and Credit Threshold settings apply only to currency, or chargeable resources.
• In calculating streaming thresholds, the BRM server considers the reserved amount in calculating the breach. It also considers any reserved amount across parallel sessions.
Chapter 7 Generating Statistics and Reports

Overview of Statistics and Reports

Introduction

This chapter explains the statistics collection and reporting that the BRM Charging Driver performs.

In this chapter

This chapter contains the following topics.

About Statistics and Reports

Generating statistics

The BRM Charging Driver uses the SMS statistics mechanism to store statistics. To collect BRM Charging Driver statistics, you must first ensure that the replication group for the SMF_STDEF_BCD table is replicated to all SLC machines. You configured replication for the SMF_STDEF_BCD table as part of configuring NCC for the BRM Charging Driver. For more information, see Configuring Replication for the BRM Charging Driver Tables (on page 14).

For information on the SMS statistics mechanism and on configuring replication for SMF database tables, see Service Management System Technical Guide and Service Management System User's Guide.

To begin collecting statistics on all SLC machines, log in to each SLC machine as smf_oper and execute the following command:

```
kill -HUP process ID of smsStatsDaemon
```

NCC immediately begins collecting BRM Charging Driver statistics for the SLC machines.

By default, NCC collects all BRM Charging Driver statistics every five minutes. You can set the frequency through the SMS UI. Select the Operator Functions menu, then Statistics Management, and then Statistics.

Generating reports

To generate reports on BRM Charging Driver statistics, go to the Operator Functions menu in the NCC SMS UI and select Report Functions. On the Report Selection tab, select one of the following three reports from the Statistics branch under BCD.

- BCD System Stats Summary
- BCD System Stats by Connection Manager
- BCD System Stats by Operation Code
The following figure shows the Report Functions window and the Report Selection tab.

For more information about generating reports, see the discussion on report functions in *Service Management System User's Guide*.
Chapter 8
Usage Scenarios

About Usage Scenarios

Introduction

This chapter presents some common usage scenarios that describe the interactions of key components when NCC integrates BRM into the charging and account settlement process.

In this chapter

This chapter contains the following topics.

A Voice Call Charged Against BRM 65
Recharge using BRM Vouchers 81
Recharge using VWS Vouchers 88
Data Charging 95
Other Scenarios 100

A Voice Call Charged Against BRM

Preconditions for voice call scenarios

The scenarios in this section describe a voice call that is charged against BRM. These scenarios assume the following preconditions:

- The subscriber is provisioned to have voice calls run a control plan with a UATB feature node.
- The subscriber's MSISDN is provisioned on BRM with prepaid GSM (voice call) service enabled.
- The subscriber has an account balance that is sufficient for 10 minutes of talk time.
- The configured grace period, specified by the eserv.config parameter CCS.ccsMacroNodes.BFTGracePeriodLength is set to 30 seconds.
- The NCC eserv.config parameter bcdActionHandler.lowCreditBuffertime is set to 30 seconds.
- BRM has been configured to charge different rates for different GSM locations.
- The appropriate entry for a voice call has been created in the ServiceProfileTagMapping array in the bcdActionHandler structure in the BCD section of the eserv.config file. For more information, see Configuring NCC for Voice Calls.
- A product and deal for voice calls has been set up. For more information, see Creating a Product and Deal for Voice Calls (on page 51).
Caller hangs up

Caller hangs up flow
Here is an example message flow covering the caller hangs up scenario.

Caller hangs up scenario
This scenario describes the actions that are taken when a subscriber makes a voice call on a CAMEL network. The voice call is handled by NCC and charged against BRM. In this sequence, the call is terminated when the caller hangs up.

See Preconditions for Voice Call Scenarios for information about the preconditions for this scenario.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | - The subscriber, Party A, makes a voice call.  
          - The MSC sends an InitialDP operation to the SLC. |
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 2      | The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.  
|        | The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. It also determines the control plan to run.  
|        | The slee_acs process runs the control plan.  
|        | The slee_acs process reaches a Universal Attempt Terminate with Billing (UATB) feature node in the control plan and invokes an InitialTimeReservation action on the BCD actions library.  
|        | The BCD actions library creates an event (BcdSleeEvent) to invoke a PCM_OP_TCF_AAA_AUTHORIZE operation and sends the event to the BCD Client that is the least busy.  
|        | The BCD Client finds a free connection on the BRM Connection Manager that has the lowest proportion of its connections currently in use.  
|        | The BCD Client invokes the PCM_OP_TCF_AAA_AUTHORIZE opcode and starts a timer for the configured value for this type of operation.  
| 3      | BRM responds to the operation and authorizes 50 seconds of time to be used.  
|        | The BCD Client receives the BRM output flist from PCM_OP_TCF_AAA_AUTHORIZE and packages it in a BcdSleeEvent and sends it to slee_acs. It also marks the BRM connection as available and cancels the operation timer.  
|        | The BCD actions library takes the output flist and translates it into the response to the InitialTimerReservation action.  
| 4      | The slee_acs process sends a RequestReportBCSMEvent operation, a Connect or a Continue operation, and an ApplyCharging operation. The ApplyCharging operation allows 45 seconds of call time (50 seconds minus the 5-second buffer previously defined).  
|        | The MSC connects the call to party B.  
|        | Party B answers the call.  
| 5      | After 45 seconds, the MSC sends ApplyChargingReport(callActive=true, 45 seconds) to slee_acs.  
|        | The slee_acs invokes an ExtendTimeReservation action on the BCD actions library.  
|        | The BCD actions library constructs an event (BcdSleeEvent) and invokes a PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode and sends the event to the least busy BCD Client.  
|        | The BCD Client finds a free connection on the least busy connection manager.  
|        | The BCD Client sends the PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode and sets the timer to the configured value for this type of operation.  
| 6      | BRM responds by indicating that the caller can use 50 more seconds.  
|        | The BCD Client sends the output flist from the operation to slee_acs in an event (BcdSleeEvent). It also marks the BRM connection as available and cancels the operation timer.  
|        | The BCD actions library translates the output flist to the response of the ExtendTimeReservation action.  
| 7      | The slee_acs sends an ApplyCharging(45 seconds) operation to the MSC.  
|        | After 18 more seconds, party A hangs up.  
| 8      | MSC sends an ApplyChargingReport(callActive=false, 63 seconds) and an EventReportBCSM(Disconnect) to slee_acs.  |
Chapter 8

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 10     | The slee_acs invokes a ConfirmTimeReservation action on the BCD actions library.  
  - The BCD actions library constructs an event (BcdSleeEvent) and invokes a PCM_OP_TCF_AAA_STOP_ACCOUNTING operation to indicate that 18 seconds of total talk time were used and sends it to the least busy Billing Client.  
  - The BCD Client finds a free connection on the least busy connection manager.  
  - The BCD Client calls the PCM_OP_SEND() function to invoke a PCM_OP_TCF_AAA_STOP_ACCOUNTING operation and sets a timer to the configured value for this type of operation. |
| 11     | BRM responds by indicating that 63 seconds of talk time has been deducted from the account.  
  - The BCD Client sends the output flist from the operation and to slee_acs packaged in an event (BcdSleeEvent). It also marks the BRM connection as available and cancels the operation timer.  
  - The BCD actions library translates the output flist into the response of the ConfirmTimeReservation action. It also stores the charge information in a call context for possible later use. |

Messages: caller hangs up

The following messages include operations sent to BRM and results returned by BRM for a voice call that is completed when the caller hangs up. The general message format is: nesting level (0, 1, or 2); field; data type; value.

**Operation:** send PCM_OP_TCF_AAA_AUTHORIZE (4002)

```
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0  
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"  
0 PIN_FLD_SESSION_ID INT [0] 0  
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-3_session_1348249_0"  
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"  
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"  
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"  
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2  
1   PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3  
   2    PIN_FLD_DIRECTION ENUM [0] 0  
   2    PIN_FLD_CELL_ID STR [0] "000c"  
   2      PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"  
1  10000 SUBSTRUCT [0] allocated 20, used 2  
2   10001 STR [0] "Present"  
2   PIN_FLD_MSID STR [0] "004085752159"  
0 PIN_FLD_REQ_MODE ENUM [0] 2
```

**Note:** REQ_MODE 2 = DURATION according to the following definitions:

```
#define PIN_TCF_AAA_REQ_MODE_AMOUNT 1  
#define PIN_TCF_AAA_REQ_MODE_DURATION 2  
#define PIN_TCF_AAA_REQ_MODE_VOLUME 4
```

**Result:** received for operation PCM_OP_TCF_AAA_AUTHORIZE (4002)

```
0 PIN_FLD_POID POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 173417 0  
0 PIN_FLD_EXPIRATION_T TSTAMP [0] (1363143998) Wed Mar 13 03:06:38 2013  
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/telephony 112551 18  
0 PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1 /reservation 175465 0  
0 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 114087 714  
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2  
1   PIN_FLD_AMOUNT DECIMAL [0] 0.500  
1   PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 9.500  
0 PIN_FLD_RESULT ENUM [0] 1
```
Chapter 8

Note: PIN_FLD_QUANTITY = 50. This means 50 seconds.

We are charging one Euro cent per second and, on BRM, a positive balance means that the subscriber owes money.

This is a limited credit account with the credit limit set to 10 Euros.

Operation: Send PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

Flags = 0

0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_PAYMENT_SESSION_ID INT [0] 1
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-3_session_1348249_0"
0 PIN_FLD_PAYMENT_OBJ POID [0] 0.0.0.1 /payment 114343 0

Note:: PIN_FLD_QUANTITY=45.0 means the subscriber has used 45 seconds.

Field ID 10000 is NCC_INFO and 10001 is NCC_FIELD. The names do not come out because they are not standard BRM fields. On a production system, the IDs may be different because 10000 and 10001 may already be used.

Result: received for operation PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

0 PIN_FLD_POID POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 173417 1
0 PIN_FLDExpiration_TS TSTAMP [0] (1363144004) Wed Mar 13 03:06:44 2013
0 PIN_FLD_RESPONSE DECIMAL [0] 50.000000000000000
0 PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1 /reservation 175465 0
0 PIN_FLD_BALANCE_GRP_OBJ POID [0] 0.0.0.1 /balance_group 114087 715
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
1 PIN_FLD_AMOUNT DECIMAL [0] 0.950
1 PIN_FLD_AVAILABLE_Resource_LIMIT DECIMAL [0] 9.050
0 PIN_FLD_RESULT ENUM [0] 1
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-3_session_1348249_0"

Operation: Send PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

Flags = 0

0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 3
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-3_session_1348249_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587930000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2 PIN_FLD_DIRECTION ENUM [0] 0
2 PIN_FLD_CELL_ID STR [0] "000c"
2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
1 10000 SUBSTRUCT [0] allocated 20, used 2
2 10001 STR [0] "Present"
2 PIN_FLD_LOCATION STR [0] "004085752159"
0 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_REQ_MODE ENUM [0] 2
0 PIN_FLD_RATING_MODE ENUM [0] 1
0 PIN_FLD_QUANTITY DECIMAL [0] 45.000000000000000
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-3_session_1348249_0"
Note: The subscriber has used 18 more seconds and that’s the end of the call.

Times reported to BRM are not cumulative.

**Result:** received for operation PCM_OP_TCF_AAA_AUTHORIZE (4007)

```plaintext
0 PIN_FLD_POID          POID [0] 0.0.0.1 /event/session/telco/gsm/ncc
277516734850845521 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-
3_session_1348249_0"
0 PIN_FLD_ACCOUNT_OBJ   POID [0] 0.0.0.1 /account 114343 0
0 PIN_FLD_SERVICE_OBJ   POID [0] 0.0.0.1 /service/telco/gsm/telephony 112551 18
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_BALANCES      ARRAY [978] allocated 20, used 2
1 PIN_FLD_AMOUNT        DECIMAL [0] 1.130
1 PIN_FLD_AVAILABLERESOURCE_LIMIT DECIMAL [0] 8.870
```
Funds expire

Funds expire flow
Here is an example message flow covering the funds expire scenario.
A Voice Call When Funds Expire
This scenario describes the actions that are taken when a voice call is made by a subscriber on a CAMEL network. The voice call is handled by NCC and charged against BRM. In this sequence, the call terminates when funds expire.
See Preconditions for Voice Call Scenarios for information about the preconditions for this scenario.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 7</td>
<td><strong>See Caller hangs up scenario</strong> (on page 66).</td>
</tr>
<tr>
<td>8</td>
<td>Slee_acs sends an ApplyCharging(45 seconds) operation to the MSC.</td>
</tr>
<tr>
<td>9</td>
<td>After 45 seconds, the MSC sends ApplyChargingReport(callActive=true, 90 seconds) to slee_acs.</td>
</tr>
<tr>
<td></td>
<td>- The slee_acs invokes an ExtendTimeReservation action on the BCD actions library.</td>
</tr>
<tr>
<td></td>
<td>- The BCD actions library constructs an event (BcdSleeEvent) to invoke a PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZ operation and sends it to the least busy BCD Client.</td>
</tr>
<tr>
<td></td>
<td>- The BCD Client finds a free connection on the least busy Connection Manager.</td>
</tr>
<tr>
<td></td>
<td>- The BCD Client calls the PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode and sets a timer to the configured value for this type of operation.</td>
</tr>
<tr>
<td>10</td>
<td>BRM determines that there are no funds left on this account and sends a response indicating that no more time can be used.</td>
</tr>
<tr>
<td></td>
<td>- The BCD Client calls sends the output flist from the operation in an event (BcdSleeEvent) to slee_acs. It also marks the BRM connection as free and cancels the operation timer.</td>
</tr>
<tr>
<td></td>
<td>- The BCD actions library translates the output flist to the failure response of the ExtendTimeReservation action.</td>
</tr>
<tr>
<td>11</td>
<td>The slee_acs process sends to the MSC an ApplyCharging operation for 5 seconds (held from the original buffer).</td>
</tr>
<tr>
<td></td>
<td>- 5 seconds elapse.</td>
</tr>
<tr>
<td>12</td>
<td>MSC sends TC_END(ApplyChargingReport(callActive=true, 95 seconds)) to slee_acs.</td>
</tr>
<tr>
<td></td>
<td>- The slee_acs invokes an ExtendTimeReservation action on the BCD actions library.</td>
</tr>
<tr>
<td></td>
<td>- The BCD actions library constructs an event (BcdSleeEvent) to invoke a PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE operation requesting 5 seconds and sends it to the least busy BCD Client.</td>
</tr>
<tr>
<td></td>
<td>- The BCD Client finds a free connection on the least busy Connection Manager.</td>
</tr>
<tr>
<td></td>
<td>- The BCD Client calls the PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode and sets a timer to the configured value for this type of operation.</td>
</tr>
<tr>
<td>13</td>
<td>BRM indicates that 5 more seconds can be used because only 5 of the 10 seconds from last time have been used.</td>
</tr>
<tr>
<td></td>
<td>- The BCD Client sends the output flist from the operation in an event (BcdSleeEvent) to slee_acs. It also marks the BRM connection as free and cancels the operation timer.</td>
</tr>
<tr>
<td></td>
<td>- The BCD actions library translates the output flist to the failure response of the ExtendTimeReservation action.</td>
</tr>
<tr>
<td>14</td>
<td>The slee_acs process sends to the MSC an ApplyCharging operation for 5 seconds, with both tone and release-if-duration-exceeded set.</td>
</tr>
<tr>
<td></td>
<td>- MSC plays a tone to the caller.</td>
</tr>
<tr>
<td></td>
<td>- 5 seconds elapse.</td>
</tr>
<tr>
<td>15</td>
<td>MSC disconnects the call and sends TC_END(ApplyChargingReport(callActive=false, 100 seconds)) to slee_acs.</td>
</tr>
</tbody>
</table>
Action | Description
--- | ---
18 | The BCD actions library constructs an event (BcdSleeEvent) for a PCM_OP_TCF_AAA_STOP_ACCOUNTING operation, indicating that 5 seconds of talk time have been used and sends it to the least busy BCD Client.
| The BCD Client finds a free connection on the least busy connection manager.
| The BCD Client calls the PCM_OP_TCF_AAA_STOP_ACCOUNTING opcode and sets a timer to the configured value for this type of operation.

19 | BRM responds to the operation, reporting that 100 seconds of total talk time has been deducted from the caller’s account.
| The BCD Client sends the output flist from the operation in an event (BcdSleeEvent) to slee_acs. It also marks the BRM connection as free and cancels the operation timer.
| The BCD actions library translates the output flist to the response of the ConfirmTimeReservation action. It also stores the charge information in the call context for possible later use.slee_acs. It also marks the BRM connection as free and cancels the operation timer.

Messages: funds expire
The following messages include operations sent to BRM and results returned by BRM for a voice call that is terminated when the caller’s funds expire. The general message format is: nesting level (0, 1, or 2); field; data type; value.

**Operation: send PCM_OP_TCF_AAA_AUTHORIZE (4002)**

```
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-3_session_1349250_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
  1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
  2 PIN_FLD_DIRECTION ENUM [0] 0
  2 PIN_FLD_CELL_ID STR [0] "000c"
  2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
  1 10000 SUBSTRUCT [0] allocated 20, used 2
  2 10001 STR [0] "Present"
  2 PIN_FLD_LOCATION STR [0] "004085752159"
0 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_REQ_MODE ENUM [0] 2
```

**Result: received for operation PCM_OP_TCF_AAA_AUTHORIZE (4002)**

```
0 PIN_FLD_POID POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 182225 0
0 PIN_FLD_EXPIRATION_T TSTAMP [0] (1363144881) Wed Mar 13 03:21:21 2013
0 PIN_FLD_QUANTITY DECIMAL [0] 50
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/telephony 112551 18
0 PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1 /reservation 184273 0
0 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 114087 721
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
  1 PIN_FLD_AMOUNT DECIMAL [0] 0.500
  1 PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 0.500
0 PIN_FLD_RESULT ENUM [0] 1
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-3_session_1349250_0"
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 114343 0
```

**Operation: send PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)**
Flags = 0
0 PIN_FLD_POID [0] 0.0.0.1/service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID [0] 1
0 PIN_FLD_AUTHORIZATION_ID [0] "brmClient-tbalcomb-2013-3-12-
 3_session_1349250_0"
0 PIN_FLD_OBJ_TYPE [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO [0] allocated 20, used 2
 1 PIN_FLD_GSM_INFO [0] allocated 20, used 3
  2 PIN_FLD_DIRECTION [0] 0
  2 PIN_FLD_CELL_ID [0] "000c"
  2 PIN_FLD_LOC_AREA_CODE [0] "064001000f"
  1 10000 [0] SUBSTRUCT [0] allocated 20, used 2
    2 10001 [0] "Present"
  2 PIN_FLD_LOCATION [0] "004085752159"
0 PIN_FLD_MSID [0] "004085752159"
0 PIN_FLD_REQ_MODE [0] 2
0 PIN_FLD_RATING_MODE [0] 1
0 PIN_FLD_QUANTITY [0] 45.000000000000000
Result: received for operation PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)
Chapter 8

Note: RATING_STATUS 26 above is insufficient funds, meaning insufficient funds for BRM to grant a whole chunk, which is set to 50 seconds in this example. BRM still grants 10 seconds which BCD treats as a success.

Operation: send PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 3
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-3_session_1349250_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
  1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
    2 PIN_FLD_DIRECTION ENUM [0] 0
    2 PIN_FLD_CELL_ID STR [0] "000c"
    2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
  1 10000 SUBSTRUCT [0] allocated 20, used 2
    2 10001 STR [0] "Present"
    2 PIN_FLD_LOCATION STR [0] "004085752159"
0 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_REQ_MODE ENUM [0] 2
0 PIN_FLD_RATING_MODE ENUM [0] 1
0 PIN_FLD_QUANTITY DECIMAL [0] 5.000000000000000

Note: BCD only reports 5 seconds used because the UATB node is withholding 5 seconds so that it can play a beep 5 seconds before the money runs out.

Result: received for operation PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

0 PIN_FLD_POID POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 182225 3
0 PIN_FLD_EXPIRATION_T TSTAMP [0] (1363144892) Wed Mar 13 03:21:32 2013
0 PIN_FLD_QUANTITY DECIMAL [0] 5.000000000000000
0 PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1 /reservation 184273 2
0 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 114087 724
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
  1 PIN_FLD_AMOUNT DECIMAL [0] 1.000
  1 PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 0
0 PIN_FLD_RESULT ENUM [0] 1
0 PIN_FLD_REASON ENUM [0] 3
0 PIN_FLD_RATING_STATUS ENUM [0] 26
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-3_session_1349250_0"

Note: Insufficient funds again, but <= the withheld time of 5 seconds so it is time to end the call. BCD returns a "no funds" result to the UATB node.
Operation: send PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 4
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-
3_session_1349250_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2 PIN_FLD_DIRECTION ENUM [0] 0
2 PIN_FLD_CELL_ID STR [0] "000c"
2 PIN_FLD_LOC_AREA_CODE STR [0] "06400100f"
1 10000 SUBSTRUCT [0] allocated 20, used 3
2 10001 STR [0] "Present"
2 10007 STR [0] "55587390000"
0 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_REQ_MODE ENUM [0] 2
0 PIN_FLD_QUANTITY DECIMAL [0] 5.000

Note: The last 5 seconds have been used, after the beep, and the caller has been disconnected. BCD reports the last 5 seconds of usage in PCM_OP_TCF_AAA_STOP_ACCOUNTING.

Result: received for operation PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

0 PIN_FLD_POID POID [0] 0.0.0.1 /event/session/telco/gsm/ncc
277516734850838505 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-
3_session_1349250_0"
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 114343 0
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/telephony 112551 18
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
1 PIN_FLD_AMOUNT DECIMAL [0] 1.000
1 PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 0
**SMS call delivered successfully**

**SMS call delivered successfully flow**

Here is an example message flow covering the SMS call delivered successfully scenario.

**SMS call delivered successfully scenario**

This scenario describes a short message (SMS) that a subscriber sends from a GSM phone. The message is handled by NCC and charged against BRM and is delivered successfully to the destination.

See Preconditions for SMS Call Scenarios for information about the preconditions for this scenario.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | The subscriber sends a short message from a GSM (voice) phone.  
      | MSC sends MOForwardSM to xmsTrigger |
| 2    | xmsTrigger sends InitialDP to slee_acs |
| 3    | The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.  
      | The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines the control plan to run.  
      | The slee_acs process runs the control plan.  
      | The slee_acs process reaches a Named Event feature node specifying Direct Event and name of the event as SMS.  
      | The Named Event feature node invokes the DirectNamedEvent action in the BCD actions library.  
      | The BCD actions library constructs an event (FlistSleeEvent) containing a PCM_OP_TCF_AAA_ACCOUNTING operation with the quantity set to 1.  
      | The BCD Client finds a free connection on the Connection Manager with the lowest proportion of its connections currently in use.  
      | The BCD Client calls the PCM_OP_TCF_AAA_ACCOUNTING opcode and sets a timer to the configured value for this type of operation. |
### Step 4
- BRM responds to the operation by indicating that the SMS message has been successfully charged.
- The BCD Client sends the output flist from the operation in an event (BcdSleeEvent) to slee_acs. It also marks the BRM connection as free and cancels the timer for the operation.
- The BRM Charging Driver actions library translates the output flist to the response of the DirectNamedEvent action.

### Step 5
- The control plan reaches an Attempt Delivery to Pending (ADP) feature node, which is the main MMX node for delivering short messages.
- The ADP feature node sends RequestReportBCSMEvent and INAP Continue operations to xmsTrigger.

### Step 6
- xmsTrigger sends MTFowardSM to the MSC serving the called subscriber.

### Step 7
- Destination MSC successfully delivers SMS to called subscriber and sends success result of MTFowardSM to xmsTrigger.

### Step 8
- xmsTrigger sends EventReportBCSM(oDisconnect) to slee_acs.
- The slee_acs process reaches an end node and clears the call context.

### Step 9
- xmsTrigger sends a success result of MOFowardSM to the originating MSC.
SMS call that fails permanently

SMS call that fails permanently flow
Here is an example message flow covering the SMS call that fails permanently scenario.

**SMS call that fails permanently scenario**
This scenario describes a short message (SMS) that a subscriber sends from a GSM phone. The message is handled by NCC and charged against BRM but delivery fails permanently.

See Preconditions for SMS Call Scenarios for information about the preconditions for this scenario.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 5</td>
<td>See steps for <em>SMS Call Delivered Successfully Scenario</em> (on page 78).</td>
</tr>
<tr>
<td>6</td>
<td>xmsTrigger sends SendRoutingInfoForSMS to HLR to find the location of the destination handset.</td>
</tr>
<tr>
<td>7</td>
<td>The destination number is invalid so HLR sends SendRoutingInfoForSMS error to xmsTrigger.</td>
</tr>
<tr>
<td>8</td>
<td>xmsTrigger sends EventReportBCSM(RouteSelectFailure) to the slee_acs process.</td>
</tr>
<tr>
<td>9</td>
<td>The slee_acs process sends ReleaseCall to xmsTrigger.</td>
</tr>
<tr>
<td>10</td>
<td>xmsTrigger sends a MTForwardSM error to the originating MSC.</td>
</tr>
</tbody>
</table>
Chapter 8

### Step 11
- The slee_acs process takes the Delivery Failure branch of the ADP feature node and reaches a Named Event feature node.
- The Named Event feature node is configured to send 1 event of type Refund SMS. It invokes a DirectNamedEvent action on the BCD actions library.
- The BCD actions library constructs an FlistSleeEvent containing PCM_OP_TCF_AAA_ACCOUNTING(1 Refund SMS).
- The BCD Client has a free BRM connection and calls the PCM_OP_TCF_AAA_ACCOUNTING opcode and sets the timer for the operation to the configured value for this type of operation.

### Step 12
- BRM responds to the operation indicating that the cost of the SMS has been refunded successfully.
- The BCD Client sends the output flist for the operation in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection and free and cancels the operation timer.
- The BCD actions library takes the output flist and translates it to the response of the DirectNamedEvent action.

---

### Recharge using BRM Vouchers

#### Preconditions for recharge using BRM vouchers

The scenarios in this section describe a subscriber who has a BRM voucher and uses it through the NCC platform to recharge the amount on a BRM account, either by way of IVR or SMS. These scenarios assume the following preconditions:

- The subscriber is provisioned so that voice calls run a control plan with a UATB feature node.
- The subscriber's MSISDN is provisioned on BRM with prepaid GSM (voice) service enabled.
- The voucher is provisioned on BRM.

For information on creating vouchers, see *Creating Vouchers* (on page 54).
### IVR BRM voucher recharge

#### IVR BRM voucher recharge flow

Here is an example message flow covering the IVR, BRM voucher recharge scenario.

### IVR BRM voucher recharge scenario

This scenario describes a subscriber who uses IVR to successfully recharge the amount on a BRM account using a BRM voucher.

See Preconditions for Recharge Using BRM Vouchers for information about the preconditions for this scenario.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | - The subscriber dials the voucher recharge number, which is toll free.  
      - MSC sends InitialDP to the slee_acs process with ServiceKey set to a special value that indicates voucher recharge. |
| 2    | - The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines the control plan to run, using the serviceKey to select the voucher recharge control plan.  
      - The slee_acs process runs the control plan.  
      - The slee_acs process reaches a Voucher Recharge feature node.  
      - The Voucher Recharge feature node sends ConnectToResource, PromptAndCollectUserInformation to the MSC, instructing the MSC to prompt the caller for the voucher number and PIN. |
Step | Action
--- | ---
3 | • MSC plays the specified announcement to the caller and collects the voucher number and PIN.
   • MSC sends PromptAndCollectUserInfo result, containing voucher number and PIN to slee_acs.
4 | • The Voucher Recharge feature node invokes the VoucherRedeem action in the BCD actions library.
   • The BCD actions library constructs an event (FlistSleeEvent) that contains PCM_OP_PYMT_TOPUP and sends it to the BCD Client.
   • The BCD Client calls the PCM_OP_PYMT_TOPUP opcode and sets a timer to the configured value for this type of operation.
5 | • BRM responds to the operation indicating that the account has been successfully recharged.
   • The BCD Client sends the output flist from the operation in an event (BcdSleeEvent) to slee_acs. It also marks the BRM connection as free and cancels the operation timer.
   • The BCD actions library translates the output flist to the response of the VoucherRedeem action.
   • The Voucher Recharge feature node sends Play Announcement to the MSC.
6 | MSC plays an announcement to the caller, stating that the recharge was successful.
7 | The announcement finishes and MSC sends Specialized ResourceReport to slee_acs.
8 | • The control plan reaches a Play Voucher Redeem Balances feature node that invokes a VoucherInfo action on the BCD actions library.
   • The BCD actions library returns the balance update information returned in the output flist for the PCM_OP_PYMT_TOPUP operation.
   • The Play Voucher Redeem Balances feature node sends Play Announcement to the MSC.
9 | • MSC plays the balance update information to the caller in an announcement.
   • The announcement ends and the MSC sends Specialized ResourceReport to slee_acs.
10 | • The control plan reaches an end node and ACS sends DisconnectForwardConnection, ReleaseCall to the MSC and clears the call context.
   • The caller is disconnected.

Messages: redemption of BRM voucher
The following messages include operations sent to BRM and results returned by BRM for the redemption of a BRM voucher. The general message format is: nesting level (0; 1, or 2); field; data type; value.

Operation: Send PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

Flags - 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
  1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
  2 PIN_FLD_DIRECTION ENUM [0] 0
  2 PIN_FLD_CELL_ID STR [0] "000c"
  2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
  1 100000 SUBSTRUCT [0] allocated 20, used 2
  2 100001 STR [0] "Present"
  2 PIN_FLD_LOCATION STR [0] "7390002"
Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

```
0 PIN_FLDF_MSID STR [0] "004085752158"
0 PIN_FLDF_FLAGS INT [0] 4

Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)
```
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_POID</td>
<td>POID [0] 0.0.0.1 /event/billing/payment/voucher 271148363502715095</td>
</tr>
<tr>
<td>PIN_FLD_CREATED_T</td>
<td>TSTAMP [0] (1331746228) Wed Mar 14 17:30:28 2012</td>
</tr>
<tr>
<td>PIN_FLD_MODIFIED_T</td>
<td>TSTAMP [0] (1331746228) Wed Mar 14 17:30:28 2012</td>
</tr>
<tr>
<td>PIN_FLD_READ_ACCESS</td>
<td>STR [0] &quot;L&quot;</td>
</tr>
<tr>
<td>PIN_FLD_WRITE_ACCESS</td>
<td>STR [0] &quot;L&quot;</td>
</tr>
<tr>
<td>PIN_FLD_ACCOUNT_OBJ</td>
<td>POID [0] 0.0.0.1 /account 135267 0</td>
</tr>
<tr>
<td>PIN_FLD_ARCHIVE_STATUS</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_BATCH_ID</td>
<td>STR [0]</td>
</tr>
<tr>
<td>PIN_FLD_CURRENCY</td>
<td>INT [0] 978</td>
</tr>
<tr>
<td>PIN_FLD_DESCR</td>
<td>STR [0]</td>
</tr>
<tr>
<td>PIN_FLD_EFFECTIVE_T</td>
<td>TSTAMP [0] (1331746228) Wed Mar 14 17:30:28 2012</td>
</tr>
<tr>
<td>PIN_FLD_EVENT_NO</td>
<td>STR [0]</td>
</tr>
<tr>
<td>PIN_FLD_FLAGS</td>
<td>INT [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_GROUP_OBJ</td>
<td>POID [0] 0.0.0.0 0 0</td>
</tr>
<tr>
<td>PIN_FLD_INCR_QUANTITY</td>
<td>DECIMAL [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_INCR_UNIT</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_ITEM_OBJ</td>
<td>POID [0] 0.0.0.1/item/payment 134999 0</td>
</tr>
<tr>
<td>PIN_FLD_MIN_QUANTITY_DECIMAL</td>
<td>[0] 0</td>
</tr>
<tr>
<td>PIN_FLD_MIN_UNIT</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_MINUnitOfWork</td>
<td>STR [0] &quot;Billing Event Log&quot;</td>
</tr>
<tr>
<td>PIN_FLD_NET_QUANTITY_DECIMAL</td>
<td>[0] 0</td>
</tr>
<tr>
<td>PIN_FLD_ORIGINAL_BATCH_ID</td>
<td>STR [0] &quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_PROFILE_LABEL_LIST</td>
<td>STR [0] &quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_PROGRAM_NAME</td>
<td>STR [0] &quot;NCC_BCD_Client&quot;</td>
</tr>
<tr>
<td>PIN_FLD_PROVIDER_DESCR</td>
<td>STR [0] &quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_PROVIDER_ID</td>
<td>POID [0] 0.0.0.0 0 0</td>
</tr>
<tr>
<td>PIN_FLD_PROVIDER_IPADDR</td>
<td>BINSTR [0] 1 00</td>
</tr>
<tr>
<td>PIN_FLD_RATED_TIMEZONE_ID</td>
<td>STR [0] &quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_RERATE_OBJ</td>
<td>POID [0] 0.0.0.0 0 0</td>
</tr>
<tr>
<td>PIN_FLD_ROUNDING_MODE</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_RUM_NAME</td>
<td>STR [0] &quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_SERVICE_OBJ</td>
<td>POID [0] 0.0.0.1/item/payment 271148363502716759 0</td>
</tr>
<tr>
<td>PIN_FLD_SPACE_USAGE</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_SESSION_ID</td>
<td>STR [0]</td>
</tr>
<tr>
<td>PIN_FLD_SESSION_OBJ</td>
<td>POID [0] 0.0.0.1 /event/billing/batch/payment 271148363502715095 0</td>
</tr>
<tr>
<td>PIN_FLD_START_T</td>
<td>TSTAMP [0] (1331746228) Wed Mar 14 17:30:28 2012</td>
</tr>
<tr>
<td>PIN_FLD_SYS_DESCR</td>
<td>STR [0] &quot;Payment - Thank you&quot;</td>
</tr>
<tr>
<td>PIN_FLD_TAX_LOCALES</td>
<td>STR [0] &quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_TAX_SUPPLIER</td>
<td>INT [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_TIMEZONE_ADJ_END_T</td>
<td>TSTAMP [0] (1331746228) Wed Mar 14 17:30:28 2012</td>
</tr>
<tr>
<td>PIN_FLD_TIMEZONE_ADJ_START_T</td>
<td>TSTAMP [0] (1331746228) Wed Mar 14 17:30:28 2012</td>
</tr>
<tr>
<td>PIN_FLD_TIMEZONE_ID</td>
<td>STR [0] &quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_TIMEZONE_MODE</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_TOD_MODE</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_UNIT</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_UNRATED_QUANTITY_DECIMAL</td>
<td>[0] 0</td>
</tr>
<tr>
<td>PIN_FLD_USAGE_TYPE</td>
<td>STR [0] &quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_USERID</td>
<td>POID [0] 0.0.0.1/service/pcm_client 1 18307</td>
</tr>
<tr>
<td>PIN_FLD_BAL_IMPACTS</td>
<td>ARRAY [0] allocated 20, used 3</td>
</tr>
<tr>
<td>PIN_FLD_BAL_GRP_OBJ</td>
<td>POID [0] 0.0.0.1/balance_group 136803 0</td>
</tr>
<tr>
<td>PIN_FLD_BALANCE_ID</td>
<td>INT [0] 978</td>
</tr>
<tr>
<td>PIN_FLD_BALANCES</td>
<td>ARRAY [0] allocated 20, used 8</td>
</tr>
<tr>
<td>PIN_FLD_AMOUNT</td>
<td>DECIMAL [0] -50</td>
</tr>
<tr>
<td>PIN_FLD_BROKER</td>
<td>STR [0] &quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_CONTRIBUTOR</td>
<td>STR [0]</td>
</tr>
<tr>
<td>PIN_FLD_GRANTOR_OBJ</td>
<td>POID [0] 0.0.0.0 0 0</td>
</tr>
<tr>
<td>PIN_FLD_ROLLOVER_DATA</td>
<td>INT [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_VALID_FROM</td>
<td>TSTAMP [0] (0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_VALID_FROM_DETAILS</td>
<td>INT [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_VALID_TO</td>
<td>TSTAMP [0] (0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_VALID_TO_DETAILS</td>
<td>INT [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_BAL_IMPACTS</td>
<td>ARRAY [0] allocated 22, used 22</td>
</tr>
<tr>
<td>PIN_FLD_ACCOUNT_OBJ</td>
<td>POID [0] 0.0.0.1/account 135267 8</td>
</tr>
<tr>
<td>PIN_FLD_AMOUNT</td>
<td>DECIMAL [0] -50</td>
</tr>
</tbody>
</table>
Chapter 8

2. PIN_FLD_AMOUNT_DEFERRED DECIMAL [0] 0
2. PIN_FLD_AMOUNT_ORIG DECIMAL [0] 0
2. PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 136803 159
2. PIN_FLD_DISCOUNT DECIMAL [0] 0
2. PIN_FLD_DISCOUNT_INFO STR [0] ""
2. PIN_FLD_GL_ID INT [0] 0
2. PIN_FLD_IMPACT_CATEGORY STR [0] "default"
2. PIN_FLD_IMPACT_TYPE ENUM [0] 2
2. PIN_FLD_ITEM_OBJ POID [0] 0.0.0.1 /item/payment 134999 0
2. PIN_FLD_LINEAGE STR [0] ""
2. PIN_FLD_OFFERING_OBJ POID [0] 0.0.0.0 0 0
2. PIN_FLD_PERCENT DECIMAL [0] 1
2. PIN_FLD_RATE_OBJ POID [0] 0.0.0.1 /rate 79195 1
2. PIN_FLD_RATE_TAG STR [0] "Rate 1"
2. PIN_FLD_RESOURCE_ID INT [0] 978
2. PIN_FLD_RESOURCE_ID_ORIG INT [0] 0
2. PIN_FLD_RUM_ID INT [0] 0
2. PIN_FLD_TAX_CODE STR [0] ""
1. PIN_FLD_EVENT_MISC_DETAILS ARRAY [0] allocated 20, used 2
2. PIN_FLD_EVENT_MISC_DETAILS ARRAY [0] allocated 20, used 2
2. PIN_FLD_REASON_DOMAIN_ID INT [0] 100
2. PIN_FLD_REASON_ID INT [0] 0
1. PIN_FLD_PAYMENT SUBSTRUCT [0] allocated 20, used 13
2. PIN_FLD_ACCOUNT_NO STR [0] ""
2. PIN_FLD_ACH INT [0] 0
2. PIN_FLD_AMOUNT DECIMAL [0] 50
2. PIN_FLD_AMOUNT_ORIGINAL_PAYMENT DECIMAL [0] 0
2. PIN_FLD_BIL NO STR [0] ""
2. PIN_FLD_CHANNEL_ID INT [0] 0
2. PIN_FLD_COMMAND ENUM [0] 0
2. PIN_FLD_CURRENCY INT [0] 978
2. PIN_FLD_MERCHANT STR [0] ""
2. PIN_FLD_PAY_TYPE ENUM [0] 10016
2. PIN_FLD_STATUS ENUM [0] 0
2. PIN_FLD_SUB_TRANS_ID STR [0] ""
2. PIN_FLD_TRANS_ID STR [0] "T1, 20.0"
1. PIN_FLD_VOUCHERS_INFO ARRAY [0] allocated 20, used 5
2. PIN_FLD_VOUCHERS_INFO ARRAY [0] allocated 20, used 5
2. PIN_FLD_CARD_EXPIRATION INT [0] 0
2. PIN_FLD_DEVICE_ID STR [0] "1000400104"
2. PIN_FLD_VALID_FROM TSTAMP [0] (0) <null>
2. PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
2. PIN_FLD_VOUCHER_PIN STR [0] "0971"
1. PIN_FLD_RESULT ENUM [0] 1
1. PIN_FLD_TYPE ENUM [0] 0
1. PIN_FLD_SELECT_RESULT INT [0] 0
1. PIN_FLD_SELECT_STATUS INT [0] 4
1. PIN_FLD_ITEM_NO STR [0] "P1-32"
0. PIN_FLD_VOUCHERS_INFO ARRAY [0] allocated 20, used 6
1. PIN_FLD_BILLINFO_OBJ POID [0] 0.0.0.1 /billinfo 138339 0
1. PIN_FLD_DEVICE_ID STR [0] "1000400104"
1. PIN_FLD_VOUCHER_PIN STR [0] "0971"
1. PIN_FLD_EXPIRATION_T TSTAMP [0] (1804935600) Sat Mar 13 11:00:00 2027
1. PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
1. PIN_FLD_VOUCHER_PIN STR [0] "0971"
0. PIN_FLD_AMOUNT DECIMAL [0] 50.000
0. PIN_FLD_CURRENCY INT [0] 978
0. PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
0. PIN_FLD_VALID_FROM TSTAMP [0] (0) <null>
0. PIN_FLD_DEAL_INFO SUBSTRUCT [0] allocated 20, used 9
1. PIN_FLD_POID POID [0] 0.0.0.1 /account 135267 0
1. PIN_FLD_DEAL_OBJ POID [0] NULL poid pointer
1. PIN_FLD_TOPUP_RESOURCE_INFO SUBSTRUCT [0] allocated 20, used 1
2. PIN_FLD_BAL_IMPACTS ARRAY [0] allocated 20, used 4
3. PIN_FLD_RESOURCE_ID INT [0] 978
3. PIN_FLD_AMOUNT DECIMAL [0] 50.000
3. PIN_FLD_VALID_FROM TSTAMP [0] (0) <null>
3. PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
1. PIN_FLD_BILLINFO_OBJ POID [0] 0.0.0.1 /billinfo 138339 0
1. PIN_FLD_AMOUNT DECIMAL [0] 50.000
1. PIN_FLD_CURRENCY INT [0] 978
1. PIN_FLD.START_T TSTAMP [0] (1331746228) Wed Mar 14 17:30:28 2012

NCC BRM Charging Driver Technical Guide
SMS BRM voucher recharge - invalid PIN

Here is an example message flow covering the SMS, BRM voucher recharge, invalid PIN scenario.

### SMS BRM voucher recharge - invalid PIN flow

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | Subscriber sends an SMS from a GSM phone to the SMS recharge number. The SMS contains the voucher number and PIN.  
|      | MSC sends MOForwardSM to xmsTrigger. |
| 2    | xmsTrigger sends InitialDP to slee_acs with the service key set to a special value indicating SMS recharge. |
### Recharge using VWS Vouchers

#### Preconditions for recharge using VWS vouchers

The scenarios in this section describe a subscriber who has a VWS voucher and uses it to recharge the amount on a BRM account using either IVR or USSD. These scenarios assume the following preconditions:

- The subscriber is provisioned so that voice calls run a control plan with a UATB feature node.
- The subscriber's MSISDN is provisioned on the BRM with prepaid GSM service enabled.
- The voucher is provisioned on the VWS.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 3    | The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.  
- The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. It also determines the control plan to run, based on the serviceKey.  
- The slee_acs process runs the SMS recharge control plan.  
- The slee_acs process reaches an Extract Content node which extracts the voucher number and PIN from the SMS and places them in temporary storage.  
- The slee_acs process reaches a Voucher Recharge node which extracts the voucher number and PIN from temporary storage and uses them to invoke a VoucherRedeem action on the BCD actions library.  
- The BCD actions library constructs an FlistSleeEvent containing PCM_OP_PYMT_TOPUP and sends it to the BCD Client.  
- The BCD Client calls the PCM_OP_PYMT_TOPUP opcode and sets a timer to the configured value for this type of operation. |
| 4    | BRM responds to the operation indicating that the PIN was invalid.  
- The BCD Client sends the output flist from the operation to slee_acs in an event (BcdSleeEvent). It also marks the BRM connection as available and cancels the operation timer.  
- The BCD actions library translates the output flist to the error response of the VoucherRedeem action. |
| 5    | The slee_acs process takes the Voucher Invalid branch of the Voucher Recharge feature node and reaches a Send Short Message Notification node, which specifies sending a failure text message.  
- The SSMN node constructs an MMX GenericMessage containing the failure message and sends it to xmsTrigger. |
| 6    | xmsTrigger sends an MTForwardSM operation to the MSC, containing the failure message. |
| 7    | The MSC sends an SMS containing the failure message to the caller. |
| 8    | The slee_acs process reaches an Accept node that sends ReleaseCall to the xmsTrigger.  
- The slee_acs reaches an end node and clears the call context. |
| 9    | xmsTrigger sends an MTForwardSM result to the MSC.  
- The MSC sends a notification of successful delivery of the original SMS to the caller. |
IVR VWS voucher recharge

IVR VWS voucher recharge flow
Here is an example message flow covering the IVR, VWS voucher recharge scenario.

IVR VWS voucher recharge scenario
This scenario describes subscriber who uses IVR to successfully recharge the amount on a BRM account using a VWS voucher.

See Preconditions for Recharge Using VWS Vouchers for information about the preconditions for this scenario.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | - The subscriber calls the toll free number for a voucher recharge.  
         - MSC sends InitialDP to the slee_acs process with serviceKey set to a special value indicating voucher recharge.  
         - xmsTrigger sends InitialDP to the slee_acs process. |
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 2      | The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain and VWS is used for the voucher. It also uses the serviceKey to select the voucher recharge control plan.  
- The slee_acs process runs the control plan.  
- The slee_acs process reaches a Voucher Recharge feature node.  
- The Voucher Recharge feature node sends ConnectToResource, PromptAndCollectUserInformation to the MSC, instructing the MSC to prompt the caller for the voucher number and PIN. |
| 3      | MSC plays the specified announcement to the caller and collects the voucher number and PIN.  
MSC sends PromptAndCollectUserInformation result, containing voucher number and PIN to the slee_acs process. |
| 4      | The Voucher Recharge feature node invokes the VoucherRedeem action on the FOX actions library.  
FOX actions library sends VR_Req to BeClient.  
BeClient sends VR_Req to VWS. |
| 5      | VWS sends VR_Ack to BeClient, indicating that the voucher has been reserved, and returning the recharge amounts for each balance type.  
BeClient sends VR_Ack to the slee_acs process. |
| 6      | The Voucher Recharge feature node invokes the WalletRecharge action on the BCD actions library.  
BCD actions library constructs an event (BcdSleeEvent) containing PCM_OPBILL_DEBIT and sends it to the BCD Client.  
The BCD Client calls the PCM.OPBILL_DEBIT opcode and sets a timer to the configured value for this type of operation. |
| 7      | BRM responds to the operation, indicating that the account has been successfully recharged.  
The BCD Client sends the output flist from the operation in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as available and cancels the operation timer.  
The BCD actions library takes the output flist and translates it to the response of the WalletRecharge action. |
| 8      | The VoucherRecharge feature node invokes the VoucherConfirm action on the FOX actions library.  
The FOX actions library sends CVR_Req to BeClient.  
BeClient sends CVR_Req to VWS. |
| 9      | VWS sends CVR_Ack to BeClient, indicating that the voucher has been permanently marked as redeemed.  
BeClient sends CVR_Ack to the slee_acs process. |
| 10     | The VoucherRecharge feature node sends PlayAnnouncement to MSC.  
MSC plays an announcement to the caller, stating that the recharge was successful. |
| 11     | The announcement finishes and MSC sends SpecializedResourceReport to the slee_acs process.  
The control plan reaches a Play Voucher Redeem Balances feature node, which invokes a VoucherInfo action on the FOX actions library.  
The FOX actions library returns the balance update information returned in the VR_Ack. |
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 12     | - The Play Voucher Redeem Balances feature node sends a PlayAnnouncement to the MSC.  
|        | - MSC plays the balance update information to the caller in an announcement.  
| 13     | The announcement ends and the MSC sends SpecializedResourceReport to the slee_acs process.  
| 14     | - The control plan reaches an end node and ACS sends DisconnectForwardConnection, ReleaseCall to the MSC and clears the call context.  
|        | - The caller is disconnected.  

**Messages: redemption of VWS voucher against BRM account**

The following messages include operations sent to BRM and results returned by BRM for the redemption of a VWS voucher against a BRM account. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation: Send PCM_OP_BILL_DEBIT (105)**

Flags - 0  
0 PIN_FLD_POID POID [0] 0.0.0.1 /account 135267 0  
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"  
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"  
0 PIN_FLD_DEBIT ARRAY [978] allocated 20, used 1  
1 PIN_FLD_BAL_OPERAND DECIMAL [0] -0.500000000000000  

**Note:** we are debiting -50 Euro cents which is the same as crediting 50 Euro cents because this is a 50 Euro cent voucher.

**Result: Received for operation PCM_OP_BILL_DEBIT (105)**

0 PIN_FLD_POID POID [0] 0.0.0.1 /account 135267 0  
0 PIN_FLD_RESULTS ARRAY [0] allocated 20, used 4  
1 PIN_FLD_BAL_IMPACTS ARRAY [0] allocated 20, used 17  
2 PIN_FLD_IMPACT_TYPE ENUM [0] 2  
2 PIN_FLD_RESOURCE_ID INT [0] 978  
2 PIN_FLD_RESOURCE_ID_ORIG INT [0] 0  
2 PIN_FLD_RATE_TAG STR [0] ""  
2 PIN_FLD/account_OBJ POID [0] 0.0.0.1 /account 135267 0  
2 PIN_FLD_RATE_OBJ POID [0] 0.0.0.0 0 0  
2 PIN_FLD_DISCOUNT DECIMAL [0] 0  
2 PIN_FLD_PERCENT DECIMAL [0] 0  
2 PIN_FLD_QUANTITY DECIMAL [0] 0  
2 PIN_FLD_AMOUNT DEFERRED DECIMAL [0] 0  
2 PIN_FLD_AMOUNT DECIMAL [0] -0.500  
2 PIN_FLD_AMOUNT_ORIG DECIMAL [0] NULL pin_decimal_t ptr  
2 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 136803 0  
2 PIN_FLD_GL_ID INT [0] 0  
2 PIN_FLD_ITEM_OBJ POID [0] 0.0.0.1 /item/misc 138851 36  
2 PIN_FLD_LINEAGE STR [0] NULL str ptr  
1 PIN_FLD_SUB_BAL_IMPACTS ARRAY [0] allocated 20, used 3  
2 PIN_FLD_SUB_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 136803 166  
2 PIN_FLD_SUB_RESOURCE_ID INT [0] 978  
2 PIN_FLD_SUB_BALANCES ARRAY [0] allocated 20, used 3  
3 PIN_FLD_AMOUNT DECIMAL [0] -0.500  
3 PIN_FLD_VALID_FROM TSTAMP [0] (0) <null>  
3 PIN_FLD_VALID_TO TSTAMP [0] (0) <null>  
1 PIN_FLD/account_OBJ POID [0] 0.0.0.1 /account 135267 0  
1 PIN_FLD_POID POID [0] 0.0.0.1 /event/billing/debit 271148363502728448 0
Chapter 8

Messages: redemption of VWS voucher using bad PIN

The following messages include operations sent to BRM and results returned by BRM for the redemption of a VWS voucher against a BRM account using a bad voucher number or PIN. The general message format is: nesting level (0; 1, or 2); field; data type; value.

Operation: Send PCM_OP_ACT_ACTIVITY (151)

Flags = 0
0 PIN_FLD_POID            POID [0] 0.0.0.1 /account 135267 0
0 PIN_FLD_PROGRAM_NAME    STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE        STR [0] "/voucher"
0 PIN_FLD_INHERITED_INFO  SUBSTRUCT [0] allocated 20, used 1
  1 10000    SUBSTRUCT [0] allocated 20, used 2
  2 10007    STR [0] "9876543214"
  2 10008    STR [0] "3210"
0 PIN_FLD_BAL_IMPACTS     ARRAY [1000011] allocated 20, used 4
  1 PIN_FLD_AMOUNT         DECIMAL [0] 1
  1 PIN_FLD_RESOURCE_ID    INT [0] 1000001
  1 PIN_FLD_ACCOUNT_OBJ    POID [0] 0.0.0.1 /account 135267 0
  1 PIN_FLD_IMPACT_TYPE    ENUM [0] 2
  2 PIN_FLD_GL_ID          INT [0] 0
  2 PIN_FLD_ITEM_OBJ       POID [0] 0.0.0.1 /item/misc 139539 0
  2 PIN_FLD_BAL_GRP_OBJ    POID [0] 0.0.0.1 /balance_group 136803 247
  2 PIN_FLD_LINEAGE        STR [0] NULL str ptr
  1 PIN_FLD_SUB_BAL_IMPACTS ARRAY [1000011] allocated 20, used 3
  2 PIN_FLD_BAL_GRP_OBJ    POID [0] 0.0.0.1 /balance_group 136803 248
  2 PIN_FLD_RESOURCE_ID    INT [0] 1000001
  2 PIN_FLD_SUB_BALANCES   ARRAY [4] allocated 20, used 8
    3 PIN_FLD_VALID_TO      TSTAMP [0] (0) <null>
    3 PIN_FLD_ROLLOVER_DATA INT [0] 0
    3 PIN_FLD_VALID_FROMDETAILS INT [0] 0
    3 PIN_FLD_VALID_TODETAILS INT [0] 0
    3 PIN_FLD_GRANTOR_OBJ   POID [0] 0.0.0.0 0 0
    3 PIN_FLD_CONTRIBUTOR_STR STR [0] "" 0
    1 PIN_FLD_ACCOUNT_OBJ   POID [0] 0.0.0.1 /account 135267 0
  1 PIN_FLD_POID           POID [0] 0.0.0.1 /event/activity/voucher 271148363502725395 0

Note: Fields 10007 and 10008 have been defined as "VOUCHER" and "PIN" respectively in the NccToBrmFieldMapping section of eserv.config. They have also been defined in BRM using Developer Center. BRM can be configured to produce an EDR showing the voucher number and PIN.

Result: Received for operation PCM_OP_ACT_ACTIVITY (151)

0 PIN_FLD_POID            POID [0] 0.0.0.1 /account 135267 0
0 PIN_FLD_RESULTS         ARRAY [0] allocated 20, used 4
  1 PIN_FLD_BAL_IMPACTS    ARRAY [1000011] allocated 20, used 8
    2 PIN_FLD_ACCOUNT_OBJ   POID [0] 0.0.0.1 /account 135267 0
    2 PIN_FLD_IMPACT_TYPE   ENUM [0] 2
    2 PIN_FLD_GL_ID         INT [0] 0
    2 PIN_FLD_ITEM_OBJ      POID [0] 0.0.0.1 /item/misc 139539 0
    2 PIN_FLD_BAL_GRP_OBJ   POID [0] 0.0.0.1 /balance_group 136803 247
    2 PIN_FLD_LINEAGE       STR [0] NULL str ptr
  1 PIN_FLD_SUB_BAL_IMPACTS ARRAY [1000011] allocated 20, used 3
    2 PIN_FLD_BAL_GRP_OBJ   POID [0] 0.0.0.1 /balance_group 136803 248
    2 PIN_FLD_RESOURCE_ID   INT [0] 1000001
    2 PIN_FLD_SUB_BALANCES  ARRAY [4] allocated 20, used 8
      3 PIN_FLD_VALID_TO     TSTAMP [0] (0) <null>
      3 PIN_FLD_ROLLOVER_DATA INT [0] 0
      3 PIN_FLD_VALID_FROMDETAILS INT [0] 0
      3 PIN_FLD_VALID_TODETAILS INT [0] 0
      3 PIN_FLD_GRANTOR_OBJ  POID [0] 0.0.0.0 0 0
      3 PIN_FLD_CONTRIBUTOR_STR STR [0] ""
    1 PIN_FLD_ACCOUNT_OBJ   POID [0] 0.0.0.1 /account 135267 0
  1 PIN_FLD_POID           POID [0] 0.0.0.1 /event/activity/voucher 271148363502725395 0
**IVR VWS multi-balance recharge scenario**

This scenario describes the sequence of messages that occurs when a subscriber uses IVR to successfully recharge more than one balance type on a BRM account using a VWS voucher, creating a new bucket for each type and setting expiry and offset dates.

See Preconditions for Recharge Using VWS Vouchers for information about the preconditions for this scenario.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | The subscriber calls the toll free number for a voucher recharge.  
MSC sends InitialDP to the slee_acs process with serviceKey set to a special value indicating a voucher recharge.  
XmsTrigger sends InitialDP to the slee_acs process. |
| 2      | The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain and VWS is used for the voucher. It also uses the serviceKey to select the voucher recharge control plan.  
The slee_acs process runs the control plan.  
The slee_acs process reaches a Voucher Recharge feature node. The Voucher Recharge feature node sends ConnectToResource, PromptAndCollectUserInformation to the MSC, instructing the MSC to prompt the caller for the voucher number and PIN. |
| 3      | The MSC plays the specified announcement to the caller and collects the voucher number and PIN.  
The MSC sends the PromptAndCollectUserInformation result, containing voucher number and PIN to the slee_acs process. |
| 4      | The Voucher Recharge feature node invokes the VoucherRedeem action on the FOX actions library.  
The FOX actions library sends VR_Req to BeClient.  
The BeClient sends VR_Req to VWS. |
| 5      | VWS sends VR_Ack to BeClient, indicating that the voucher has been reserved, returning the recharge amounts for each balance type.  
The BeClient sends VR_Ack to the slee_acs process. |
| 6      | The Voucher Recharge feature node invokes the WalletRecharge action on the BCD actions library.  
The BCD actions library constructs an event (BcdSleeEvent) containing PCM_OP_BILL_DEBIT and sends it to the BCD Client.  
The BCD Client calls the PCM_OP_BILL_DEBIT opcode and sets a timer to the configured value for this type of operation. |
| 7      | BRM responds to the operation, indicating that the account has been successfully recharged.  
The BCD Client sends the output flist from the operation in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as available and cancels the operation timer.  
The BCD actions library takes the output flist and translates it to the response of the WalletRecharge action. |
| 8      | The VoucherRecharge feature node invokes the VoucherConfirm action on the FOX actions library.  
The FOX actions library sends CVR_Req to BeClient.  
The BeClient sends CVR_Req to VWS. |
<p>| 9      | VWS sends CVR_Ack to BeClient, indicating that the voucher has been permanently marked as redeemed. |</p>
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 10     | - The VoucherRecharge feature node sends PlayAnnouncement to MSC.  
       | - MSC plays an announcement to the caller, stating that the recharge was  
       |   successful.  
| 11     | - The announcement finishes and MSC sends SpecializedResourceReport to the  
       |   slee_acs process.  
|        | - The control plan reaches a Play Voucher Redeem Balances feature node, which  
       |   invokes a VoucherInfo action on the FOX actions library.  
|        | - The FOX actions library returns the balance update information returned in the  
       |   VR_Ack.  
| 12     | - The Play Voucher Redeem Balances feature node sends a PlayAnnouncement to  
       |   the MSC.  
|        | - The MSC plays the balance update information to the caller in an announcement.  
| 13     | The announcement ends and the MSC sends SpecializedResourceReport to the slee_acs  
       |   process.  
| 14     | - The control plan reaches an end node and ACS sends  
       |   DisconnectForwardConnection, ReleaseCall to the MSC and clears the call  
       |   context.  
|        | - The caller is disconnected.  

**Data Charging**

**Preconditions for data session charging on BRM**

The scenario in this section describes a subscriber who initiates and later closes a data session using a mobile device. The GGSN communicates with the NCC software on the SLC by way of Diameter credit control operations. NCC charges the data session against an account on BRM. For information on creating a BRM product and deal for data calls, see *Creating a Product and Deal for Data Calls* (on page 52).

This scenario assumes the following preconditions:

- The subscriber is provisioned so that data calls run a control plan with a UATB feature node.
- The subscriber’s MSISDN is provisioned on BRM with prepaid GSM service enabled.
- The subscriber has sufficient money on his account for the data session.
Data session charging

Data session charging flow
Here is an example message flow covering the data session charging scenario.

### Charging for a Data Session
This scenario describes a data session that is initiated and later closed by a subscriber using a mobile device.

See Preconditions for data session charging on BRM for information about the preconditions for this scenario.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | - The subscriber starts a data session.  
         - The GGSN sends Diameter Credit Control Request (CCR), INITIAL_REQUEST to DCA |
<p>| 2      | The DCA sends an InitialDP operation to the slee_acs. |</p>
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3      |  The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.  
      |  The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. It also determines the control plan to run.  
      |  The slee_process runs the control plan.  
      |  The slee_acs process reaches a Universal Attempt Terminate with Billing (UATB) node in the control plan and invokes the InitialTimeReservation action on the BCD actions library.  
      |  The BCD actions library constructs an event (BcdSleeEvent) to invoke a PCM_OP_TCF_AAA_AUTHORIZE operation and sends it to the least busy BCD Client.  
      |  The BCD Client finds a free connection on the least busy Connection Manager  
      |  The BCD Client calls the PCM_OP_TCF_AAA_AUTHORIZE opcode and sets a timer to the configured value for this type of operation.  
      |  BRM responds to the operation by indicating that 5 million bytes of data can be used.  
      |  The BCD Client sends the output flist for this operation in an event (BcdSleeEvent) to slee_acs. It also marks the BRM connection as available and cancels the operation timer.  
      |  The BCD actions library translates the output flist into the response to the InitialTimeReservation action. It also copies the value of 5 million bytes into the tag used for this purpose by the DCA software. |
| 4      |  The slee_acs process sends RequestreportBCSMEvent, ApplyCharging, and Connect operations to DCA. The 5 million bytes of available data is held in an extension in ApplyCharging.  
      |  DCA sends Credit Control Answer (CCA) to the GGSN to specify that Granted Units = 5 million bytes.  
      |  After some time, the GGSN sends CCR(UPDATE_REQUEST) to DCA, indicating that 5 million bytes have been used.  
      |  DCA sends ApplyChargingReport(callActive=true) to the slee_acs process.  
      |  The slee_acs process invokes an ExtendTimeReservation action on the BCD actions library.  
      |  The BCD actions library constructs an event (BcdSleeEvent) for the PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode for 5 million bytes and sends it to the least busy BCD Client.  
      |  The BCD Client finds a connection on the least busy Connection Manager.  
      |  The BCD Client calls the PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode and sets a timer to the configured value for this type of operation.  
      |  BRM responds to the operation indicating that an additional 5 million bytes may be used.  
      |  The BCD Client sends the output flist from the operation in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as available and cancels the operation timer.  
      |  The BCD actions library translates the output flist into the response to the ExtendTimeReservation action. It also places the 5 million bytes of allowed usage in the standard profile tag used by DCA. |
| 11     |  The slee_acs process sends an ApplyCharging operation to the DCA.  
      |  DCA sends CCA to the GGSN to specify Granted Units = 5 million bytes. |
Chapter 8

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>After using an additional 1.3 million bytes, the user terminates the session. GGSN sends CCR(TERMINATION_REQUEST) to DCA indicating that an additional 1.3 million bytes have been used.</td>
</tr>
<tr>
<td>14</td>
<td>DCA sends ApplyChargingReport(callActive=false), EventReportBCSM(oDisconnect) to the slee_acs process. The value of 6.3 million bytes is passed in an extension in ApplyChargingReport.</td>
</tr>
<tr>
<td>15</td>
<td>The slee_acs process invokes a ConfirmTimeReservation action on the BCD actions library. It stores the value of 1.3 million bytes in context. The BCD actions library constructs an event (BcdSleeEvent) for invoking the PCM_OP_TCF_AAA_STOP_ACCOUNTING opcode, indicating that 6.3 million bytes have been used and sends it to the least busy BCD Client. The BCD Client finds a free connection on the least busy Connection Manager. The BCD Client calls the PCM_OP_TCF_AAA_STOP_ACCOUNTING opcode and sets a timer to the configured value for this type of operation.</td>
</tr>
<tr>
<td>16</td>
<td>BRM responds to the operation indicating that the cost of 1.3 million bytes of data has been permanently deducted from the account. The BCD Client sends the output flist from the operation in an event (BcdSleeEvent) to the slee_acs process. It also marks the connection as available and cancels the operation timer. The BCD actions library translates the output flist into the response to the ConfirmTimeReservation action. It also stores the charge information in call context for possible use in the future.</td>
</tr>
</tbody>
</table>

Messages: data session
The following messages include operations sent to BRM and results returned by BRM for a data session. The general message format is: nesting level (0; 1, or 2); field; data type; value.

Operation: Send PCM_OP_TCF_AAA_AUTHORIZE (4002)
Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1/service/telco/gsm/data -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-cleejoh_session_126042_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752158"
0 PIN_FLD_CALLED_NUMBER STR [0] "555873900000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2 PIN_FLD_DIRECTION ENUM [0] 0
2 PIN_FLD_CELL_ID STR [0] "000c"
2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
1 10000 SUBSTRUCT [0] allocated 20, used 2
2 10001 STR [0] "Present"
2 PIN_FLD_LOCATION STR [0] "7390002"
0 PIN_FLD_MSID STR [0] "004085752158"
0 PIN_FLD_REQ_MODE ENUM [0] 4

Note: REQ_MODE now indicates VOLUME

Result: received for operation PCM_OP_TCF_AAA_AUTHORIZE (4002)
0 PIN_FLD_POID POID [0] 0.0.0.1/active_session/telco/gsm/ncc 138601 0
0 PIN_FLD_EXPIRATION_T TSTAMP [0] (1331837307) Thu Mar 15 18:48:27 2012
0 PIN_FLD_QUANTITY DECIMAL [0] 5000000
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1/service/telco/gsm/data 135651 10
0 PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1/reservation 136041 0
0 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1/balance_group 136803 203
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
1 PIN_FLD_AMOUNT DECIMAL [0] 0.048
1 PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 9.952
0 PIN_FLD_RESULT ENUM [0] 1

98 NCC BRM Charging Driver Technical Guide
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-cleejoh_session_126042_0"
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 135267 0

<table>
<thead>
<tr>
<th>PIN_FLD_QUANTITY says 5 million bytes.</th>
</tr>
</thead>
</table>

**Operation:** Send PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

<table>
<thead>
<tr>
<th>Flags = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/data -1 0</td>
</tr>
<tr>
<td>0 PIN_FLD_PROGRAM_NAME STR [0] &quot;NCC_BCD_Client&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_AUTHORIZATION_ID STR [0] &quot;brmClient-cleejoh_session_126042_0&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_OBJ_TYPE STR [0] &quot;gsm/ncc&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_CALLING_NUMBER STR [0] &quot;004085752158&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_CALLED_NUMBER STR [0] &quot;55587390000&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2</td>
</tr>
<tr>
<td>1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3</td>
</tr>
<tr>
<td>2 PIN_FLD_DIRECTION ENUM [0] 0</td>
</tr>
<tr>
<td>2 PIN_FLD_CELL_ID STR [0] &quot;000c&quot;</td>
</tr>
<tr>
<td>2 PIN_FLD_LOC_AREA_CODE STR [0] &quot;064001000f&quot;</td>
</tr>
<tr>
<td>1 10000 SUBSTRUCT [0] allocated 20, used 2</td>
</tr>
<tr>
<td>2 10001 STR [0] &quot;Present&quot;</td>
</tr>
<tr>
<td>2 PIN_FLD_LOCATION STR [0] &quot;7390002&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_MSID STR [0] &quot;004085752158&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_REQ_MODE ENUM [0] 4</td>
</tr>
<tr>
<td>0 PIN_FLD_BYTES_UPLINK DECIMAL [0] 5000000.000000000000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Note: subscriber has uploaded 5 million bytes.</th>
</tr>
</thead>
</table>

**Result:**

Received for operation PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

| PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/ncc 138601 2 |
| PIN_FLD_QUANTITY DECIMAL [0] 500000 |
| PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1 /reservation 136041 1 |
| PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 136803 205 |
| PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2 |
| 1 PIN_FLD_AMOUNT DECIMAL [0] 0.048 |
| 1 PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 9.857 |
| 0 PIN_FLD_RESULT ENUM [0] 1 |
| 0 PIN_FLD_RATING_STATUS ENUM [0] 0 |
| 0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-cleejoh_session_126042_0" |

<table>
<thead>
<tr>
<th>Note: The subscriber is now allowed to upload 5 million more bytes</th>
</tr>
</thead>
</table>

**Operation:** Send PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

<table>
<thead>
<tr>
<th>Flags = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/data -1 0</td>
</tr>
<tr>
<td>0 PIN_FLD_PROGRAM_NAME STR [0] &quot;NCC_BCD_Client&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_AUTHORIZATION_ID STR [0] &quot;brmClient-cleejoh_session_126042_0&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_OBJ_TYPE STR [0] &quot;gsm/ncc&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_CALLING_NUMBER STR [0] &quot;004085752158&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_CALLED_NUMBER STR [0] &quot;55587390000&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2</td>
</tr>
<tr>
<td>1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3</td>
</tr>
<tr>
<td>2 PIN_FLD_DIRECTION ENUM [0] 0</td>
</tr>
<tr>
<td>2 PIN_FLD_CELL_ID STR [0] &quot;000c&quot;</td>
</tr>
<tr>
<td>2 PIN_FLD_LOC_AREA_CODE STR [0] &quot;064001000f&quot;</td>
</tr>
<tr>
<td>1 10000 SUBSTRUCT [0] allocated 20, used 3</td>
</tr>
<tr>
<td>2 10001 STR [0] &quot;Present&quot;</td>
</tr>
<tr>
<td>2 PIN_FLD_LOCATION STR [0] &quot;7390002&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_MSID STR [0] &quot;004085752158&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_REQ_MODE ENUM [0] 4</td>
</tr>
<tr>
<td>0 PIN_FLD_BYTES_UPLINK DECIMAL [0] 1300000.000000000000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Note: the subscriber has uploaded 1.3 million more bytes and ended the session.</th>
</tr>
</thead>
</table>

**Result:**

Received for operation PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

| PIN_FLD_POID POID [0] 0.0.0.1 /event/session/telco/gsm/ncc 271148363502721926 0 |
| PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-cleejoh_session_126042_0" |
Other Scenarios

All BCD clients busy

All BCD clients busy flow
Here is an example message flow that illustrates the sequence of messages when all BCD Clients are busy.

```
All BCD clients busy flow

1. InitialDP

2. Can't find a free BCD client
   - RequestReportBCSMEvent
   - ApplyCharging(10 minutes, callReleaseIfTcpExpire=true), Connect
   - Caller hangs up after 10 minutes

3. BFT EDR Written
   - (duration 5 minutes)
```

```
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 135267 0
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/data 135651 10
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
1 PIN_FLD_AMOUNT DECIMAL [0] 0.108
1 PIN_FLD_AVAILABLERESOURCELIMIT DECIMAL [0] 9.892
```
All BCD clients busy scenario
This scenario describes the sequence of messages that occurs when a subscriber makes a call and all BCD Clients are busy.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | The subscriber makes a voice call.  
     | The MSC sends an InitialDP operation to the SLC.  
     | The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.  
     | The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that wallet information is on the BRM domain. It also determines the control plan to run.  
     | The slee_acs process runs the control plan.  
     | The slee_acs process reaches a Universal Attempt Terminate with Billing (UATB) node in the control plan and so invokes an InitialTimeReservation action on the BCD actions library.  
     | The BCD actions library constructs a BcdSleeEvent to invoke a PCM_OP_TCF_AAA_AUTHORIZE operation but is unable to send it to the least busy BCD Client. Therefore, the BCD actions library returns an error response to the InitialTimeReservation action.  
     | The UATB node exits the Billing Failure Treatment feature node.  
     | The control plan runs an Attempt Terminate with Duration node, which grants the user 10 minutes of time.  |
| 2    | The slee_acs process sends an ApplyCharging(releaseOnTcpExpiry=true, tone=false, 10 minutes) operation to MSC.  
     | 5 minutes elapse |
| 3    | MSC cuts caller off  
     | MSC sends ApplyChargingReport(callActive=false) to slee_acs  
     | The slee_acs process writes a BFT EDR for 5 minutes and releases the call connection. |
No free connections to BRM

No free connections to BRM flow
Here is an example message flow that illustrates the sequence of messages when there are no free connections to BRM.

No free connections to BRM scenario
This scenario describes the sequence of messages that occurs when a subscriber's wallet is in the BRM domain and there are no free connections to BRM.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | The subscriber makes a voice call.  
     | The MSC sends an InitialDP operation to the SLC.  
     | The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.  
     | The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that wallet information is on the BRM domain. It also determines the control plan to run.  
     | The slee_acs process reaches a Universal Attempt Terminate with Billing (UATB) node in the control plan and so invokes an InitialTimeReservation action on the BCD actions library. |
| 2    | The BCD actions library constructs a BcdSleeEvent for the invoke of a PCM_OP_TCF_AAA_AUTHORIZE operation and sends it to the least busy BCD Billing Client. |
| 3    | The BCD Client has no free BRM connections so sends a BcdSleeEvent to slee_acs with a status indicating that the BRM could not be contacted.  
     | The UATB feature node takes the Billing Failure Treatment exit.  
     | The control plan runs an Attempt Terminate with Duration node which grants the user 10 minutes of time. |
### Chapter 8, Usage Scenarios

#### PCM operation timeout

**PCM operation timeout flow**

Here is an example message flow that illustrates the message sequence when a PCM operation timeout occurs.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 4    | The `slee_acs` process sends an `ApplyCharging(releaseOnTcpExpiry=true, tone=false, 10 minutes)` operation to MSC.  
      | The caller hangs up after 5 minutes. |
| 5    | MSC cuts off caller  
      | MSC sends `ApplyChargingReport(callActive=false)` to `slee_acs`  
      | The `slee_acs` process writes a BFT EDR for 5 minutes and releases the call connection. |
PCM operation timeout scenario
This scenario describes a scenario in which a subscriber makes a call and a PCM operation that the BCD Client sends to BRM times out.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | The subscriber makes a voice call.  
The MSC sends an InitialDP operation to the SLC.  
The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.  
The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that wallet information is on the BRM domain. It also determines the control plan to run.  
The slee_acs process runs the control plan.  
The slee_acs process reaches a Universal Attempt Terminate with Billing (UATB) node in the control plan and so invokes an InitialTimeReservation action on the BCD actions library.  
The BCD actions library constructs a BcdSleeEvent to invoke a PCM_OP_TCF_AAA_AUTHORIZE operation and sends it to the least busy BCD Billing Client.  
The BCD Client finds a free connection – the one with the lowest proportion of its connections currently in use.  
The BCD Billing Client invokes the PCM_OP_TCF_AAA_AUTHORIZE operation via the PCM_OP_SEND() function and sets a timer to the configured value for this type of operation.  
The operation timer expires.  
The BCD Client closes the connection and sends a BcdSleeEvent to the slee_acs process with a status indicating that the operation timed out.  
The UATB feature node takes the Billing Failure Treatment exit.  
The control plan runs an Attempt Terminate with Duration feature node, which grants the user 10 minutes.  
The slee_acs process sends ApplyCharging(releaseOnTcpExpiry=true, tone=false, 10 minutes) to MSC.  
The caller hangs up after 5 minutes.  
The MSC cuts off caller  
MSC sends ApplyChargingReport(callActive=false) to slee_acs  
The slee_acs process writes a BFT EDR for 5 minutes and releases the call context. |

Messages: PCM operation timeout
The following messages include operations sent to BRM and results returned by BRM for a PCM operation timeout. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation:** send PCM_OP_TCF_AAA_AUTHORIZE (4002)

```plaintext
Flags = 0
0 PIN_FLD_POID       POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID  INT [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-thalcomb-2013-3-12-6_session_1371273_0"
0 PIN_FLD_OBJ_TYPE    STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_GSM_INFO    SUBSTRUCT [0] allocated 20, used 3
2 PIN_FLD_DIRECTION   ENUM [0] 0
2 PIN_FLD_CELL_ID     STR [0] "000c"
2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
```
Mid-call tariff change

Mid-call tariff change flow
Here is an example message flow that illustrates the message sequence when a mid-call tariff change occurs.

1. InitialDP
2. RequestReportBOSCMEvent
   ApplyCharging(45 seconds)
   Connect
3. CallActive=true
   45 seconds
4. MAP AnyTimeInterrogation Request
5. MAP AnyTimeInterrogation Response
6. POM_OP_TCF_AAA_AUTHORIZE
    (location information)
7. POM_OP_TCF_AAA_AUTHORIZE
    response (80 seconds)
8. POM_OP_TCF_AAA_UPDATE_AND_REQUEST
    AUTHORIZATION (45 seconds, new location)
9. POM_OP_TCF_AAA_UPDATE_AND_REQUEST
    response (80 seconds)
10. Caller talks for 45 seconds
    & changes location
11. ApplyChargingReport
    CallActive=false
    53 seconds
    EventReportBCSM(disconnect)
12. POM_OP_TCF_AAA_STOP_ACCOUNTING
    (18 seconds)
13. POM_OP_TCF_AAA_STOP_ACCOUNTING

Mid-call tariff change scenario
This scenario describes the sequence of messages that occurs when the subscriber changes location during the call, causing a different rate to take effect. This scenario assumes that NCC has been configured to query the location of the caller when the location has not been checked for a period of time.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | The subscriber makes a call.  
MSC sends an InitialDP operation, which contains location information, to the slee_acs process.  
The slee_acs process receives the InitialDP and passes it to the CCS service loader. It also stores the location information in context.  
The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is in the BRM domain. It also determines which control plan to run.  
The slee_acs process runs the control plan.  
The slee_acs process reaches a Universal Attempt Terminate with Billing (UATB) node in the control plan and so invokes an InitialTimeReservation action on the BCD actions library. |
| 2    | The BCD actions library constructs a BcdSleeEvent to invoke a PCM_OP_TCF_AAA_AUTHORIZE operation (including the location information) and sends it to the least busy BCD Client.  
The BCD Client finds a free connection on the least busy Connection Manager.  
The BCD Client calls the PCM_OP_SEND() function to invoke the PCM_OP_TCF_AAA_AUTHORIZE operation and sets a timer to the configured value for this type of operation. |
| 3    | BRM responds to the operation, indicating that 50 seconds of time can be used.  
The BCD Client receives the operation output flist from the PCM_OP_RECEIVE() function and sends it in a BcdSleeEvent to the slee_acs process. It also free the BRM connection for reuse and cancels the operation timer.  
The BCD actions library takes the output flist and translates it into the response of the InitialTimeReservation action. |
| 4    | The slee_acs process sends RequestReportBCSMEvent, ApplyCharging(45 seconds), and Connect operations to MSC.  
MSC connects the call to called party.  
The called party answers. |
| 5    | After 45 seconds, MSC sends ApplyChargingReport(callActive=true, 45 seconds) to the slee_acs process. |
| 6    | The subscriber location timer has expired so the slee_acs process sends MAP AnyTimeInterrogationRequest to MSC to determine the caller’s new location. |
| 7    | MSC sends the result of AnyTimeInterrogationRequest to the slee_acs process to indicate the subscriber’s new location.  
The slee_acs process invokes the ExtendTimeReservation action on the BCD actions library.  
The BCD actions library constructs a BcdSleeEvent to invoke the PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE operation, containing the new location, and sends it to the least busy BCD Client.  
The BCD Client finds a free connection on the least busy Connection Manager. |
| 8    | The BCD Client sends BRM a PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE operation and sets a timer to the configured value for this type of operation. |
### Step 9
- The BCD Client receives the operation output flist and sends it in a BcdSleeEvent to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.
- The BCD actions library translates the output flist into the response of the ExtendTimeReservation action.

### Step 10
- The slee_acs process sends an `ApplyCharging(45 seconds)` operation to MSC.
- After 18 seconds the calling party hangs up.

### Step 11
- MSC sends `ApplyChargingReport(callActive=false,63 seconds)`, `EventReportBCSM(oDiConnect)` to the slee_acs process.
- The slee_acs process invokes a `ConfirmTimeReservation` action on the BCD actions library.

### Step 12
- The BCD actions library constructs a BcdSleeEvent to invoke a `PCM_OP_TCF_AAA_STOPACCOUNTING` operation, indicating that 63 seconds of talk time has been used and sends it to the least busy BCD Client.
- The BCD Client finds a free connection on the least busy Connection Manager.
- The BCD Client invokes the `PCM_OP_TCF_AAA_STOPACCOUNTING` operation and sets a timer to the configured value for this type of operation.

### Step 13
- BRM responds to the operation by indicating that the cost of the 63 seconds of talk time has been permanently deducted from the account. The amount reflects that the first 45 seconds of the call were made from the first location and the remaining 18 seconds of the call were made from the new location.
- The BCD Client receives the operation output flist and sends it in a BcdSleeEvent to the slee_acs process. It also marks the connection as free for reuse and cancels the operation timer.
- The BCD actions library translates the output flist into the response to the `ConfirmTimeReservation` action. It also stores the charge information in the call context for possible later use.
- The slee_acs process proceeds with the control plan. If there are no more feature nodes, processing ends for this call.

### Messages: mid-call tariff change
The following messages include operations sent to BRM and results returned by BRM when a mid-call tariff change occurs. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation**: send `PCM_OP_TCF_AAA_AUTHORIZE (4002)`

```plaintext
Flags  =  0
0 PIN_FLD_POID        POID [0]  0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0]  "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID  INT [0]   0
0 PIN_FLD_AUTHORIZATION_ID STR [0]  "brmClient-tbalcomb-2013-3-12-6_session_1447355_0"
0 PIN_FLD_OBJ_TYPE     STR [0]  "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0]  "644F220021002C"
0 PIN_FLD_CALLED_NUMBER STR [0]  "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
  1 PIN_FLD_GSM_INFO     SUBSTRUCT [0] allocated 20, used 3
  2 PIN_FLD_DIRECTION    ENUM [0]   0
  2 PIN_FLD_CELL_ID      STR [0]  "002C"
  2 PIN_FLD_LOC_AREA_CODE STR [0]  "644F220021"
  1 10000               SUBSTRUCT [0] allocated 20, used 2
  2 10001               STR [0]  "Present"
  2 PIN_FLD_LOCATION     STR [0]  "644F220021002C"
0 PIN_FLD_MSID         STR [0]  "004085752159"
0 PIN_FLD_REQ_MODE     ENUM [0]   2
```

**Result**: received for operation `PCM_OP_TCF_AAA_AUTHORIZE (4002)`
Chapter 8

0 PIN_FLD_POID POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 188289 0
0 PIN_FLD_EXPIRATION_T TSTAMP [0] (1363821071) Wed Mar 20 23:11:11 2013
0 PIN_FLD_QUANTITY DECIMAL [0] 50
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/telephony 112551 21
0 PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1 /reservation 184385 0
0 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 114087 1027
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
1 PIN_FLD_AMOUNT DECIMAL [0] 0.500
1 PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 7.500
0 PIN_FLD_RESULT ENUM [0] 1
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1447355_0"
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 114343 0

Operation: send PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 1
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1447355_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "644F220021002C"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2 PIN_FLD_DIRECTION ENUM [0] 0
2 PIN_FLD_CELL_ID STR [0] "002c"
2 PIN_FLD_LOC_AREA_CODE STR [0] "644F220021"
1 10000 SUBSTRUCT [0] allocated 20, used 2
2 10001 STR [0] "Present"
2 PIN_FLD_LOCATION STR [0] "644F220021002C"
0 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_REQ_MODE ENUM [0] 2
0 PIN_FLD_RATING_MODE ENUM [0] 1
0 PIN_FLD_QUANTITY DECIMAL [0] 45.000000000000000

Result: received for operation PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

0 PIN_FLD_POID POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 188289 1
0 PIN_FLD_EXPIRATION_T TSTAMP [0] (1363821074) Wed Mar 20 23:11:14 2013
0 PIN_FLD_QUANTITY DECIMAL [0] 50.000000000000000
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/telephony 112551 21
0 PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1 /reservation 184385 0
0 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 114087 1028
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
1 PIN_FLD_AMOUNT DECIMAL [0] 0.950
1 PIN_FLD_AVAILABLERESOURCE_LIMIT DECIMAL [0] 7.050
0 PIN_FLD_RESULT ENUM [0] 1
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1447355_0"

Operation: send PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 2
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1447355_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "644F220021002C"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2 PIN_FLD_DIRECTION ENUM [0] 0
2 PIN_FLD_CELL_ID STR [0] "002c"
2 PIN_FLD_LOC_AREA_CODE STR [0] "644F220021"
1 10000 SUBSTRUCT [0] allocated 20, used 2
2 10001 STR [0] "Present"
2 PIN_FLD_LOCATION STR [0] "644F220021002C"
0 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_REQ_MODE ENUM [0] 2
0 PIN_FLD_RATING_MODE ENUM [0] 1

Result: received for operation PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)
Chapter 8

Usage Scenarios

Here is an example message flow that illustrates the message sequence when a subscriber is not found in the NCC database.

**Subscriber not found in NCC database**

**Subscriber not found in NCC database flow**

![Subscriber not found in NCC database flow diagram]

---

**Subscriber not found in NCC database flow**

![Subscriber not found in NCC database flow diagram]
**Subscriber not found in NCC database scenario**

This scenario describes the sequence of messages that occurs when the subscriber is not found in the NCC database.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | The subscriber makes a call.  
|      | MSC sends an InitialDP operation to the SLC.  
|      | The slee_acs process receives the InitialDP and passes it to the CCS service loader.  
|      | The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. The subscriber is not found and there is no default control plan to run. |
| 2    | The slee_acs process sends a TC_END(ReleaseCall) message to MSC.  
|      | The call is disconnected. |
No NCC subscriber, default control plan

No NCC subscriber, default control plan flow

No NCC Subscriber Default Control Plan

1. InitialDP

2. Subscriber not found
   - PCM_OP_TCF_AAA_AUTHORIZE

3. RequestReportBCSMEvent
   - ApplyCharging(45 seconds)
   - Connect
   - Caller talks for 45 seconds

4. ApplyChargingReport CallActive=true
   - 45 seconds

5. PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (45 seconds)
   - PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE response (60 seconds)

6. ApplyCharging (45 seconds)
   - Caller hangs up after 21 seconds

7. ApplyChargingReport
   - CallActive=false
   - 66 seconds
   - EventReportBCSM(disconnect)

8. PCM_OP_TCF_AAA_STOP_ACCOUNTING (21 seconds)
   - PCM_OP_TCF_AAA_STOP_ACCOUNTING response
No NCC subscriber, default control plan scenario
This scenario describes the sequence of messages that occur when a subscriber is not found in the NCC database and the default control plan is found.

The subscriber makes a voice call on a CAMEL network. The voice call is handled by NCC and charged against BRM. In this sequence, the call is terminated when the caller hangs up.

See Preconditions for Voice Call Scenarios for information about the preconditions for this scenario.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | The subscriber makes a call.  
MSC sends an InitialDP operation to the SLC  
The slee_acs process receives the InitialDP and passes it to the CCS service loader.  
The CCS service loader attempts to look up the subscriber and wallet in the SCP database on the SLC; the subscriber is not found.  
The CCS service loader then checks for a default control plan, which it finds.  
The slee_acs process runs the default control plan.  
The slee_acs process reaches a Universal Attempt Terminate with Billing (UATB) feature node in the control plan and invokes an InitialTimeReservation action on the BCD actions library.  
The BCD actions library creates an event (BcdSleeEvent) to invoke a PCM_OP_TCF_AAA_AUTHORIZE operation and sends the event to the BCD Client that is the least busy.  
The BCD Client finds a free connection on the BRM Connection Manager that has the lowest proportion of its connections currently in use.  
The BCD Client invokes the PCM_OP_TCF_AAA_AUTHORIZE opcode and starts a timer for the configured value for this type of operation. |
| 2      | BRM responds to the operation and authorizes 50 seconds of time to be used.  
The BCD Client receives the BRM output flist from PCM_OP_TCF_AAA_AUTHORIZE and packages it in a BcdSleeEvent and sends it to the slee_acs process. It also marks the BRM connection as available and cancels the operation timer.  
The BCD actions library takes the output flist and translates it into the response to the InitialTimerReservation action. |
| 3      | The slee_acs process sends a RequestReportBCSMEvent operation, a Connect or a Continue operation, and an ApplyCharging operation.  
The ApplyCharging operation allows 45 seconds of call time (50 seconds minus the 5-second buffer previously defined).  
The MSC connects the call to party B.  
Party B answers the call.  
After 45 seconds, the MSC sends ApplyChargingReport(callActive=true, 45 seconds) to the slee_acs process. |
| 4      | The slee_acs invokes an ExtendTimeReservation action on the BCD actions library.  
The BCD actions library constructs an event (BcdSleeEvent) and invokes a PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode and sends the event to the least busy BCD Client.  
The BCD Client finds a free connection on the least busy connection manager.  
The BCD Client sends the PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode and sets the timer to the configured value for this type of operation. |
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
</table>
| 6      | - BRM responds by indicating that the caller can use 50 more seconds.  
        - The BCD Client sends the output flist from the operation to the slee_acs process in an event (BcdSleeEvent). It also marks the BRM connection as available and cancels the operation timer.  
        - The BCD actions library translates the output flist to the response of the ExtendTimeReservation action. |
| 7      | - The slee_acs process sends an ApplyCharging(45 seconds) operation to the MSC.  
        - After 18 more seconds, party A hangs up. |
| 8      | - MSC sends an ApplyChargingReport(callActive=false, 63 seconds) and an EventReportBCSM(Disconnect) to the slee_acs process. |
| 9      | - The slee_acs invokes a ConfirmTimeReservation action on the BCD actions library.  
        - The BCD actions library constructs an event (BcdSleeEvent) and invokes a PCM_OP_TCF_AAA_STOP_ACCOUNTING operation to indicate that 18 seconds of total talk time were used and sends it to the least busy Billing Client.  
        - The BCD Client finds a free connection on the least busy connection manager.  
        - The BCD Client calls the PCM_OP_SEND() function to invoke a PCM_OP_TCF_AAA_STOP_ACCOUNTING operation and sets a timer to the configured value for this type of operation. |
| 10     | - BRM responds by indicating that 63 seconds of talk time has been deducted from the account.  
        - The BCD Client sends the output flist from the operation to the slee_acs process packaged in an event (BcdSleeEvent). It also marks the BRM connection as available and cancels the operation timer.  
        - The BCD actions library translates the output flist into the response of the ConfirmTimeReservation action. It also stores the charge information in a call context for possible later use. |
Called subscriber busy

Called subscriber busy flow
Here is an example message flow that illustrates the message sequence that occurs when a called subscriber is busy.

Called subscriber busy scenario
This scenario describes the message sequence that occurs when the called party's phone is busy.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The subscriber makes a call.</td>
</tr>
<tr>
<td></td>
<td>The subscriber makes a call.</td>
</tr>
<tr>
<td></td>
<td>MSC sends an InitialDP operation to the slee_acs process.</td>
</tr>
<tr>
<td></td>
<td>The slee_acs process receives the InitialDP and passes it to the CCS service loader.</td>
</tr>
<tr>
<td></td>
<td>The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is in the BRM domain. It also determines which control plan to run.</td>
</tr>
<tr>
<td></td>
<td>The slee_acs process runs the control plan.</td>
</tr>
<tr>
<td></td>
<td>The slee_acs process reaches a Universal Attempt Terminate with Billing (UATB) feature node in the control plan and so invokes an InitialTimeReservation action on the BCD actions library.</td>
</tr>
</tbody>
</table>
Step | Action
--- | ---
2 |  - The BCD actions library constructs a BcdSleeEvent object to invoke a PCM_OP_TCF_AAA_AUTHORIZE operation and sends it to the least busy BCD Client.
   - The BCD Client finds a free connection on the least busy Connection Manager.
   - The BCD Client calls the PCM_OP_SEND() function to invoke the PCM_OP_TCF_AAA_AUTHORIZE operation and sets a timer to the configured value for this type of operation.
3 |  - BRM responds to the operation, indicating that 50 seconds of time can be used.
   - The BCD Client receives the operation output flist from the PCM_OP_RECEIVE() function and sends it in a BcdSleeEvent object to the slee_acs process. It also frees the BRM connection for reuse and cancels the operation timer.
   - The BCD actions library translates the output flist into the response of the InitialTimeReservation action.
4 |  - The slee_acs process sends RequestReportBCSMEvent, ApplyCharging(45 seconds), and Connect operations to MSC.
   - MSC attempts to connect the call to the called party but the party is busy.
5 |  - MSC sends ApplyChargingReport(callActive=false), EventReportBCSM(oBusy) to the slee_acs process.
   - The slee_acs process invokes a ConfirmTimeReservation action (0 seconds) on the BCD actions library.
6 |  - The BCD actions library constructs a BcdSleeEvent object to invoke a PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION operation and sends it to the least busy BCD Client.
   - The BCD Client finds a free connection on the least busy Connection Manager. The Billing Client invokes the PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION operation and sets a timer to the configured value for this type of operation.
7 |  - BRM responds to the operation indicating that the reservation has been canceled and no money has been deducted from the account.
   - The BCD Client receives the operation output flist and sends it in a BcdSleeEvent object to the slee_acs process. It also frees the connection for reuse and cancels the operation timer.
   - The BCD actions library translates the output flist into the response of the ConfirmTimeReservation action.
8 |  - The slee_acs process sends a TC_END(ReleaseCall) message to MSC.
   - The call is disconnected.

Messages: called subscriber busy
The following messages include operations sent to BRM and results returned by BRM for a voice call when the called subscriber is busy. The general message format is: nesting level (0; 1, or 2); field; data type; value.

Operation: send PCM_OP_TCF_AAA_AUTHORIZE (4002)
Flags - 0
# number of field entries allocated 20, used 10
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1370272_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
Chapter 8

1  PIN_FLD_GSM_INFO   SUBSTRUCT [0] allocated 20, used 3
2        PIN_FLD_DIRECTION   ENUM [0] 0
2        PIN_FLD_CELL_ID   STR [0] "000c"
2        PIN_FLD_LOC_AREA_CODE   STR [0] "064001000f"
1     10000  SUBSTRUCT [0] allocated 20, used 2
2        10001  STR [0] "Present"
2        PIN_FLD_LOCATION  STR [0] "004085752159"
0  PIN_FLD_MSID   STR [0] "004085752159"
0  PIN_FLD_REQ_MODE ENUM [0] 2

Result: received for operation PCM_OP_TCF_AAA_AUTHORIZE (4002)

# number of field entries allocated 20, used 11
0  PIN_FLD_POID   POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 179130 0
0  PIN_FLD_EXPIRATION_T TSTAMP [0] (1363156055) Wed Mar 13 06:27:35 2013
0  PIN_FLD_QUANTITY  DECIMAL [0] 50
0  PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/telephony 112551 18
0  PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1 /reservation 178106 0
0  PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 114087 772
0  PIN_FLD_BALANCES  ARRAY [978] allocated 20, used 2
1  PIN_FLD_AMOUNT DECIMAL [0] 0.500
1  PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 9.500
0  PIN_FLD_RESULT ENUM [0] 1
0  PIN_FLD_RATING_STATUS ENUM [0] 0
0  PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1370272_0"
0  PIN_FLD_BALANCE_OBJ POID [0] 0.0.0.1 /account 114343 0

Operation: send PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION (4004)

Flags = 0
# number of field entries allocated 20, used 10
0  PIN_FLD_POID   POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0  PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0  PIN_FLD_SESSION_ID  INT [0] 1
0  PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1370272_0"
0  PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0  PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0  PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0  PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1  PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2        PIN_FLD_DIRECTION   ENUM [0] 0
2        PIN_FLD_CELL_ID   STR [0] "000c"
2        PIN_FLD_LOC_AREA_CODE   STR [0] "064001000f"
1     10000  SUBSTRUCT [0] allocated 20, used 3
2        10001  STR [0] "Present"
2        PIN_FLD_LOCATION  STR [0] "004085752159"
2        10007  STR [0] "55587390000"
0  PIN_FLD_MSID   STR [0] "004085752159"
0  PIN_FLD_REQ_MODE ENUM [0] 2

Result: received for operation PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION (4004)

# number of field entries allocated 20, used 2
0  PIN_FLD_POID   POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 179130 1
0  PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1370272_0"
With ASB voice call costs and cumulative balance

With ASB voice call costs and cumulative balance flow
Here is an example message flow that illustrates the message sequence that occurs for a successful voice call with ASB, Voice Call Costs, and Cumulative Balances.
With ASB voice call costs and cumulative balance scenario
This scenario describes the sequence of messages that occurs for the following conditions:

- The Account State Branch feature node is used to test whether the account is active - if so, the call proceeds
After the call, the Voice Call Cost feature node plays the cost of the session and the Cumulative Balances feature node plays the balances of an account on BRM.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | The subscriber makes a call. MSC sends an InitialDP operation to the slee_acs process.  
- The slee_acs process receives the InitialDP and passes it to the CCS service loader.  
- The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is in the BRM domain. It also determines which control plan to run.  
- The slee_acs process runs the control plan.  
- The slee_acs process reaches an Account State Branch feature node in the control plan and so invokes a WalletInfo action on the BCD actions library.  
- The BCD actions library constructs a BcdSleeEvent object for a PCM_OP_TCF_AAA_QUERY_BALANCE operation and sends it to the least busy BCD Client.  
- The BCD Client finds a free connection on the least busy Connection Manager. |
| 2    | The BCD Client invokes the PCM_OP_TCF_AAA_QUERY_BALANCE and sets a timer to the configured value for this type of operation. |
| 3    | BRM responds to the operation.  
- The BCD Client receives the operation output flist and sends it in a BcdSleeEvent to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
- The BCD actions library translates the output flist into the response of the WalletInfo action.  
- The slee_acs process takes the active branch of the Account State Branch feature node. |
| 4    | PCM_OP_READ_FLDS (Retrieve Account Details)  
- PCM_OP_READ_FLDS response |
| 5    | The slee_acs process reaches a Universal Attempt Terminate with Billing (UATB) feature node in the control plan and invokes an InitialTimeReservation action on the BCD actions library.  
- The BCD actions library constructs a BcdSleeEvent object for an invoke of a PCM_OP_TCF_AAA_AUTHORIZE operation and sends it to the least busy BCD Client.  
- The BCD Billing Client invokes the PCM_OP_TCF_AAA_AUTHORIZE operation and sets a timer to the configured value for this type of operation. |
| 7    | BRM responds to the operation by indicating that 50 seconds of time can be used.  
- The BCD Client receives the operation output flist and sends it in a BcdSleeEvent object to the slee_acs process. It also marks the connection as free for reuse and cancels the operation timer.  
- The BCD actions library translates the output flist into the response of the InitialTimeReservation action. |
| 8    | The slee_acs process sends RequestReportBCSMEvent, ApplyCharging(45 seconds), and Connect operations to the MSC.  
- The MSC connects the called party.  
- The called party answers. |
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 9    | After 45 seconds the MSC sends an `ApplyChargingReport(callActive=true, 45 seconds)` to the `slee_acs` process.  
|      | The `slee_acs` process invokes and `ExtendTimeReservation` action on the BCD actions library.  
|      | The BCD actions library creates a BcdSleeEvent for an invoke of a `PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE` operation and sends it to the least busy BCD Client.  
|      | The BCD Client finds a free connection on the least busy Connection Manager.  
| 10   | The BCD Client invokes the `PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE` operation and sets a timer to the configured value for this type of operation.  
| 11   | BRM responds to the operation by indicating that 50 more seconds of time can be used.  
|      | The BCD Client receives the operation output `flist` and sends it in a BcdSleeEvent object to the `slee_acs` process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
|      | The BCD actions library translates the output `flist` into the response of the `ExtendTimeReservation` action.  
| 12   | The `slee_acs` process sends an `ApplyCharging(45 seconds)` operation to the MSC.  
|      | After 18 seconds, the called party hangs up.  
| 13   | MSC sends `EventReportBCSM(oDisconnect)`, `ApplyChargingReport(callActive=false, 63 seconds)` to the `slee_acs` process.  
|      | The `slee_acs` process invokes a `ConfirmTimeReservation` action on the BCD actions library.  
| 14   | The BCD actions library creates a BcdSleeEvent object for an invoke of a `PCM_OP_TCF_AAA_STOP_ACCOUNTING` operation.  
|      | The BCD Client invokes the `PCM_OP_TCF_AAA_STOP_ACCOUNTING` operation and sets a timer to the configured value for this type of operation.  
| 15   | BRM responds to the operation by indicating that the cost of 63 seconds has been prematurely deducted from the account.  
|      | The BCD Client receives the operation output `flist` and sends it in a BcdSleeEvent object to the `slee_acs` process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
|      | The BCD actions library translates the output `flist` into the response of the `ConfirmTimeReservation` action. It also stores the charge information in the call context for possible use later.  
|      | The `slee_acs` process proceeds with the control plan and reaches a Voice Call Cost feature node.  
|      | The Voice Call Cost feature node invokes the `GetChargeDetails` action on the BCD actions library.  
|      | The BCD actions library returns the charge information that it stored earlier in the call context.  
| 16   | The `slee_acs` process sends `ConnectToResource`, `PlayAnnouncement` to the MSC to play the call cost information.  

120  NCC BRM Charging Driver Technical Guide
### Step 17
- The MSC plays the call cost announcement and then sends SpecializedResourceReport to the slee_acs process to indicate that the announcement is complete.
- The slee_acs process takes the success branch of the Voice Call Cost feature node.
- The slee_acs process receives a Cumulative Balances feature node.
- The Cumulative Balances feature node invokes a GetWallet action on the FOX actions library. This does a lookup in the SCP database on the SLC and returns. (The FOX actions library is used for the subscriber domain and GetWallet is a subscriber domain action.)
- The Cumulative Balances feature node invokes a WalletInfo action on the BCD actions library.
- The BCD actions library creates a BcdSleeEvent object for an invoke of a PCM_OP_TCF_AAA_QUERY_BALANCE operation and sends it to the least busy BCD Client.

### Step 18
- The BCD Client invokes the PCM_OP_TCF_AAA_QUERY_BALANCE operation and sets a timer to the configured value for this type of operation.

### Step 19
- BRM responds to the operation.
- The BCD Client receives the operation output flist and sends it in a BcdSleeEvent object to the slee_acs process. It also marks the BRM connection as free and cancels the operation timer.
- The BCD actions library translates the output flist into the response of the WalletInfo action.

### Step 20
- The Cumulative Balances feature node plays the balance information from the WalletInfo response by sending a PlayAnnouncement action to the MSC.

### Step 21
- The MSC plays the account balances announcement and then sends a SpecializedResourceReport message to the slee_acs process to indicate that the announcement is complete.
- The Cumulative Balances feature node takes the success branch exit and the slee_acs process reaches an end node.

### Step 22
- The slee_acs process sends a (DisconnectForwardConnection, ReleaseCall) action to the MSC and releases the call connection.
- The caller is disconnected.

---

**Messages: with ASB voice call costs and cumulative balance**
The following messages include operations sent to BRM and results returned by BRM for a voice call with ASB voice call costs and cumulative balance. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation: Send PCM_OP_TCF_AAA_QUERY_BALANCE (4104)**

```plaintext
Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_INFO SUBSTRUCT [0] allocated 20, used 3
2 PIN_FLD_DIRECTION ENUM [0] 0
2 PIN_FLD_CELL_ID STR [0] "000c"
2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
1 10000 SUBSTRUCT [0] allocated 20, used 2
2 10001 STR [0] "Present"
2 PIN_FLD_LOCATION STR [0] "004085752159"
0 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_FLAGS INT [0] 4
```
Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

0 PIN_FLD_POID          POID [0] 0.0.0.1 /balance_group 114087 1059
0 PIN_FLD_ACCOUNT_OBJ   POID [0] 0.0.0.1 /account 114100 0
0 PIN_FLD_BILLINFO_OBJ  POID [0] 0.0.0.1 /billinfo 111015 0
0 PIN_FLD_EFFECTIVE_T   TSTAMP [0] (1362359921) Mon Mar 04 01:18:41 2013
0 PIN_FLD_BALANCES      ARRAY [978] allocated 20, used 10
  1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  1 PIN_FLD_NEXT_BAL      DECIMAL [0] 0
  1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
  1 PIN_FLD_CURRENT_BAL   DECIMAL [0] -25
  1 PIN_FLD_SUB_BALANCES  ARRAY [0] allocated 20, used 12
    2 PIN_FLD_CONTRIBUTOR_STR STR [0] ""
    2 PIN_FLD_VALID_TO      TSTAMP [0] (0) <null>
    2 PIN_FLD_VALID_FROM     TSTAMP [0] (1362297600) Sun Mar 03 08:00:00 2013
    2 PIN_FLD_VALID_FROMDETAILS INT [0] 0
    2 PIN_FLD_CURRENT_BAL   DECIMAL [0] -25
    2 PIN_FLD_NEXT_BAL      DECIMAL [0] 0
    2 PIN_FLD_DELAYED_BAL    DECIMAL [0] 0
    2 PIN_FLD_ROLLOVER_DATA INT [0] 0
    2 PIN_FLD_GRANTOR_OBJ   POID [0] 0.0.0.1 /purchased_product 112231 0
    2 PIN_FLD_FLAGS          INT [0] 2
    1 PIN_FLD_CURRENT_TOTAL DECIMAL [0] -25
    1 PIN_FLD_CREDIT_FLOOR  DECIMAL [0] NULL
    1 PIN_FLD_CREDIT_LIMIT  DECIMAL [0] 0
    1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
    1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_REALTIME_CNTR INT [0] 4
0 PIN_FLD_BALANCES      ARRAY [100001] allocated 20, used 10
  1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  1 PIN_FLD_NEXT_BAL      DECIMAL [0] 0
  1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
  1 PIN_FLD_CURRENT_BAL   DECIMAL [0] 6
  1 PIN_FLD_SUB_BALANCES  ARRAY [4] allocated 20, used 12
    2 PIN_FLD_CONTRIBUTOR_STR STR [0] ""
    2 PIN_FLD_VALID_TO      TSTAMP [0] (0) <null>
    2 PIN_FLD_VALID_FROM     TSTAMP [0] (1362528348) Wed Mar 06 00:05:48 2013
    2 PIN_FLD_VALID_FROMDETAILS INT [0] 0
    2 PIN_FLD_CURRENT_BAL   DECIMAL [0] 6
    2 PIN_FLD_NEXT_BAL      DECIMAL [0] 0
    2 PIN_FLD_DELAYED_BAL    DECIMAL [0] 0
    2 PIN_FLD_ROLLOVER_DATA INT [0] 0
    2 PIN_FLD_GRANTOR_OBJ   POID [0] 0.0.0.1 /purchased_product 112231 0
    2 PIN_FLD_FLAGS          INT [0] 2
    1 PIN_FLD_CURRENT_TOTAL DECIMAL [0] 6
    1 PIN_FLD_CREDIT_FLOOR  DECIMAL [0] NULL
    1 PIN_FLD_CREDIT_LIMIT  DECIMAL [0] 0
    1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
    1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_BALANCES      ARRAY [1000076] allocated 20, used 10
  1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  1 PIN_FLD_NEXT_BAL      DECIMAL [0] 0
  1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
  1 PIN_FLD_CURRENT_BAL   DECIMAL [0] -125.37
  1 PIN_FLD_SUB_BALANCES  ARRAY [2] allocated 20, used 12
    2 PIN_FLD_CONTRIBUTOR_STR STR [0] ""
    2 PIN_FLD_VALID_TO      TSTAMP [0] (1363939853) Fri Mar 22 08:10:53 2013
    2 PIN_FLD_VALID_FROM     TSTAMP [0] (1362297600) Sun Mar 03 08:00:00 2013
    2 PIN_FLD_VALID_FROMDETAILS INT [0] 0
    2 PIN_FLD_CURRENT_BAL   DECIMAL [0] -125.37
    2 PIN_FLD_NEXT_BAL      DECIMAL [0] 0
    2 PIN_FLD_DELAYED_BAL    DECIMAL [0] 0
    2 PIN_FLD_ROLLOVER_DATA INT [0] 0
    2 PIN_FLD_GRANTOR_OBJ   POID [0] 0.0.0.1 /purchased_product 112231 0
    2 PIN_FLD_FLAGS          INT [0] 2
Chapter 8, Usage Scenarios

1. PIN_FLD_CURRENT_TOTAL DECIMAL [0] -125.37
2. PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
3. PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
4. PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
5. PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""

Operation: Send PCM_OP_READ_FLDS (4)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /account 114343 0
0 PIN_FLD_CURRENCY INT [0] 0
0 PIN_FLD_STATUS ENUM [0] 0
0 PIN_FLD_CLOSE_WHEN_T TSTAMP [0] (0) <null>
0 PIN_FLD_MOD_T TSTAMP [0] (0) <null>
0 PIN_FLD_LAST_STATUS_T TSTAMP [0] (0) <null>

Result: received for operation PCM_OP_READ_FLDS (4)

0 PIN_FLD_POID POID [0] 0.0.0.1 /account 114343 46
0 PIN_FLD_CURRENCY INT [0] 978
0 PIN_FLD_STATUS ENUM [0] 10100
0 PIN_FLD_CLOSE_WHEN_T TSTAMP [0] (0) <null>
0 PIN_FLD_MOD_T TSTAMP [0] (1363075261) Tue Mar 12 08:01:01 2013
0 PIN_FLD_LAST_STATUS_T TSTAMP [0] (1363072273) Tue Mar 12 07:11:13 2013

Operation: Send PCM_OP_TCF_AAA_AUTHORIZE (4002)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmcClient-tbalcomb-2013-3-12-6_session_1456361_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2 PIN_FLD_DIRECTION ENUM [0] 0
2 PIN_FLD_CELL_ID STR [0] "000c"
2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
1 10000 SUBSTRUCT [0] allocated 20, used 2
2 10001 STR [0] "Present"
2 PIN_FLD_LOCATION STR [0] "004085752159"
0 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_REQ_MODE ENUM [0] 2

Result: received for operation PCM_OP_TCF_AAA_AUTHORIZE (4002)

0 PIN_FLD_POID POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 189820 0
0 PIN_FLD_EXPIRATION_T TSTAMP [0] (1363897662) Thu Mar 21 20:27:42 2013
0 PIN_FLD_QUANTITY DECIMAL [0] 50
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/telephony 112551 21
0 PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1 /reservation 10987 1059
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
1 PIN_FLD_AMOUNT DECIMAL [0] 0.500
1 PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 24.500
0 PIN_FLD_RESULT ENUM [0] 1
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmcClient-tbalcomb-2013-3-12-6_session_1456361_0"
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 114343 0

Operation: Send PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 1
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmcClient-tbalcomb-2013-3-12-6_session_1456361_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
Result: received for operation PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZ (4026)

Operation: to send PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

Flags: 0

Result: received for operation PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZ (4026)

Operation: send PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

Flags: 0
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 3
  1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
  2 PIN_FLD_DIRECTION ENUM [0] 0
  2 PIN_FLD_CELL_ID STR [0] "000c"
  2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
1 10000 SUBSTRUCT [0] allocated 20, used 2
  2 10001 STR [0] "Present"
  2 PIN_FLD_LOCATION STR [0] "004085752159"
0 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_REQ_MODE ENUM [0] 2
0 PIN_FLD_RATING_MODE ENUM [0] 1
0 PIN_FLD_QUANTITY DECIMAL [0] 45.000000000000000

Result: received for operation PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

0 PIN_FLD_POID POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 189820 3
0 PIN_FLD_EXPIRATION_T TSTAMP [0] (1363897669) Thu Mar 21 20:27:49 2013
0 PIN_FLD_QUANTITY DECIMAL [0] 50.000000000000000
0 PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1 /reservation 191868 2
0 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 114087 1062
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
  1 PIN_FLD_AMOUNT DECIMAL [0] 1.850
  1 PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 23.150
0 PIN_FLD_RESULT ENUM [0] 1
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1456361_0"

Operation: Send PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SET INT [0] 4
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1456361_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 3
  1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
  2 PIN_FLD_DIRECTION ENUM [0] 0
  2 PIN_FLD_CELL_ID STR [0] "000c"
  2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
1 10000 SUBSTRUCT [0] allocated 20, used 3
  2 10001 STR [0] "Present"
  2 PIN_FLD_LOCATION STR [0] "004085752159"
  2 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_REQ_MODE ENUM [0] 2
0 PIN_FLD_RATING_MODE ENUM [0] 1
0 PIN_FLD_QUANTITY DECIMAL [0] 30.000000000000000

Result: received for operation PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

0 PIN_FLD_POID POID [0] 0.0.0.1 /event/session/telco/gsm/ncc 277675064525247937 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1456361_0"
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 114343 0
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/telephony 112551 21
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
  1 PIN_FLD_AMOUNT DECIMAL [0] 1.650
  1 PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 23.350

Operation: Send PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
<table>
<thead>
<tr>
<th>PIN_FLD_EXTENDED_INFO</th>
<th>SUBSTRUCT [0] allocated 20, used 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_DIRECTION</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_CELL_ID</td>
<td>STR [0] &quot;000c&quot;</td>
</tr>
<tr>
<td>PIN_FLD_LOC_AREA_CODE</td>
<td>STR [0] &quot;064001000f&quot;</td>
</tr>
<tr>
<td>PIN_FLD_GSM_INFO</td>
<td>SUBSTRUCT [0] allocated 20, used 3</td>
</tr>
<tr>
<td>PIN_FLD_LOCATION</td>
<td>STR [0] &quot;004085752159&quot;</td>
</tr>
<tr>
<td>PIN_FLD_MSID</td>
<td>STR [0] &quot;004085752159&quot;</td>
</tr>
<tr>
<td>PIN_FLD_FLAGS</td>
<td>INT [0] 4</td>
</tr>
</tbody>
</table>

**Result:** received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)
Operation: send is PCM_OP_READ_FLDS (4)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /account 114343 0
0 PIN_FLD_CURRENCY INT [0] 0
0 PIN_FLD_STATUS ENUM [0] 0
0 PIN_FLD_CLOSE_WHEN_T TSTAMP [0] (0) <null>
0 PIN_FLD_MOD_T TSTAMP [0] (1363075261) Tue Mar 12 08:01:01 2013
0 PIN_FLD_LAST_STATUS_T TSTAMP [0] (0) <null>

Result: received for operation PCM_OP_READ_FLDS (4)

0 PIN_FLD_POID POID [0] 0.0.0.1 /account 114343 46
0 PIN_FLD_CURRENCY INT [0] 978
0 PIN_FLD_STATUS ENUM [0] 10100
0 PIN_FLD_CLOSE_WHEN_T TSTAMP [0] (0) <null>
0 PIN_FLD_MOD_T TSTAMP [0] (1363075261) Tue Mar 12 08:01:01 2013
0 PIN_FLD_LAST_STATUS_T TSTAMP [0] (1363072273) Tue Mar 12 07:11:13 2013
Chapter 8

**SMS call info and SMS account balances**

**SMS call info and SMS account balances flow**
Here is an example message flow that illustrates the message sequence that occurs for a voice call with the SMS Call Info and SMS Account Balance feature nodes.
Chapter 8, Usage Scenarios

Voice call, caller hang-up, post-call call info and balance info SMS

1. InitialDP
2. PCM_OP_TCF_ANG_AUTHORIZ
   PCM_OP_TCF_ANG_AUTHORIZ
   response (60 seconds)
3. PCM_OP_TCF_ANG_AUTHORIZ
4. CallActive = true
   45 seconds
5. PCM_OP_TCF_ANG_UPDATE_AND_REAUTHORIZ
   45 seconds
   response (50 seconds)
6. PCM_OP_TCF_ANG_UPDATE_AND_REAUTHORIZ
   response (50 seconds)
7. ApplyCharging (45 seconds)
8. Caller hangs up after 18 seconds
9. ApplyChargingReport
   CallActive = false
   63 seconds
   EventReport BCSM (disconnect)
10. PCM_OP_TCF_ANG_STOP_ACCOUNTING
    response
11. GenericMessage
12. PCM_OP_TCF_ANG_QUERY_BALANCE
    (Retrieve Balance Details)
13. PCM_OP_TCF_ANG_QUERY_BALANCE
    response
14. PCM_OP_TCF_ANG_QUERY_BALANCE
    response
15. PCM_OP_READ_FILE
    (Retrieve Account Details)
16. PCM_OP_READ_FILE
    response
17. GenericMessage
18. MTForwardSM
19. MTForwardSM
result
### SMS call info and SMS account balances scenario

This scenario describes the sequence of messages that occurs after a voice call when the SMS Call Info feature node is used to send the cost of the session to the caller and the SMS Account Balances feature node is used to send the balances of a BRM account to the caller.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | The subscriber, Party A, makes a voice call.  
The MSC sends an InitialDP operation to the SLC. |
| 2    | The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.  
The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. It also determines the control plan to run.  
The slee_acs process runs the control plan.  
The slee_acs process reaches a Universal Attempt Terminate with Billing (UATB) feature node in the control plan and invokes an InitialTimeReservation action on the BCD actions library.  
The BCD actions library creates an event (BcdSleeEvent) to invoke a PCM_OP_TCF_AAA_AUTHORIZE operation and sends the event to the BCD Client that is the least busy.  
The BCD Client finds a free connection on the least busy Connection Manager.  
The BCD Client invokes the PCM_OP_TCF_AAA_AUTHORIZE opcode and starts a timer for the configured value for this type of operation. |
| 3    | BRM responds to the operation and authorizes 50 seconds of time to be used.  
The BCD Client receives the BRM output flist fromPCM_OP_TCF_AAA_AUTHORIZE and packages it in a BcdSleeEvent and sends it to slee_acs. It also marks the BRM connection as available and cancels the operation timer.  
The BCD actions library takes the output flist and translates it into the response to the InitialTimerReservation action. |
| 4    | The slee_acs process sends a RequestReportBCSMEvent operation, a Connect or a Continue operation, and an ApplyCharging operation. The ApplyCharging operation allows 45 seconds of call time (50 seconds minus the 5 second buffer previously defined).  
The MSC connects the call to party B.  
Party B answers the call. |
| 5    | After 45 seconds, the MSC sends ApplyChargingReport(callActive=true, 45 seconds) to slee_acs. |
| 6    | The slee_acs invokes an ExtendTimeReservation action on the BCD actions library.  
The BCD actions library constructs an event (BcdSleeEvent) and invokes a PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode and sends the event to the least busy BCD Client.  
The BCD Client finds a free connection on the least busy connection manager.  
The BCD Client sends the PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE opcode and sets the timer to the configured value for this type of operation. |
| 7    | BRM responds by indicating that the caller can use 50 more seconds.  
The BCD Client sends the output flist from the operation to slee_acs in an event (BcdSleeEvent). It also marks the BRM connection as available and cancels the operation timer.  
The BCD actions library translates the output flist to the response of the ExtendTimeReservation action. |
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 8    | The slee_acs sends an ApplyCharging(45 seconds) operation to the MSC.  
|      | After 18 more seconds, party A hangs up.  
| 9    | MSC sends an ApplyChargingReport(callActive=false, 63 seconds) and an EventReportBCSM(Disconnect) to slee_acs.  
| 10   | The slee_acs invokes a ConfirmTimeReservation action on the BCD actions library.  
|      | The BCD actions library constructs an event (BcdSleeEvent) and invokes a PCM_OP_TCF_AAA_STOP_ACCOUNTING operation to indicate that 63 seconds of total talk time were used and sends it to the least busy Billing Client.  
|      | The BCD Client finds a free connection on the least busy connection manager.  
|      | The BCD Client calls the PCM_OP_SEND() function to invoke a PCM_OP_TCF_AAA_STOP_ACCOUNTING operation and sets a timer to the configured value for this type of operation.  
| 11   | BRM responds by indicating that 63 seconds of talk time has been deducted from the account.  
|      | The BCD Client sends the output flist from the operation and to slee_acs packaged in an event (BcdSleeEvent). It also marks the BRM connection as available and cancels the operation timer.  
|      | The BCD actions library translates the output flist into the response of the ConfirmTimeReservation action. It also stores the charge information in a call context for possible later use.  
| 12   | The slee_acs process reaches an SMS Call Info feature node.  
|      | The SMS Call Info feature node invokes the GetChargeDetails action on the BCD actions library  
|      | The BCD actions library returns the charge information stored previously.  
|      | The slee_acs process constructs an MMX GenericMessage containing the charge information and sends it to xmsTrigger.  
| 13   | MSC sends an SMS containing the charge information to the caller.  
|      | The slee_acs process takes the success branch of the SMS Call Info feature node.  
|      | The slee_acs process reaches an SMS Account Balances Balances feature node.  
|      | The SMS Account Balances Balances feature node invokes a GetWallet action on the FOX actions library. This does a lookup in the SCP database on the SLC and returns. (The FOX actions library is used because FOX is used for the subscriber domain and GetWallet is a subscriber domain action.)  
|      | The SMS Account Balances Balances feature node invokes a WalletInfo action on the BCD actions library.  
|      | The BCD actions library constructs a BcdSleeEvent for the invoke of a PCM_OP_TCF_AAA_QUERY_BALANCE operation and sends it to the least busy BCD Client.  
|      | The BCD Client invokes the PCM_OP_TCF_AAA_QUERY_BALANCE operation and sets a timer to the configured value for this type of operation.  
| 14   | BRM responds to the operation.  
|      | The BCD Client receives the output flist and sends it in a BcdSleeEvent object to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
|      | The BCD actions library translates the output flist into the response of the WalletInfo action.  
| 15   | PCM_OP_READ_FLDS(Retrieve Account Details)  
| 16   | PCM_OP_READ_FLDS response
### Step 17
- The SMS Account Balances Balances feature node constructs an MMX GenericMessage object containing the balances information and sends it to xmsTrigger.

### Step 18
- xmsTrigger sends an MTForwardSM operation to the MSC, containing balances information.

### Step 19
- The MSC sends an SMS containing the balances information to the caller.
- The slee_acs process takes the success branch of the SMS Account Balances feature node and reaches an end node.

### Messages: SMS call info and SMS account balances

The following messages include operations sent to BRM and results returned by BRM for a voice call that is followed by the SMS Call Info feature node sending the cost of the call and SMS Account Balances feature node sending the account balance. The general message format is: nesting level (0; 1, or 2); field; data type; value.

#### Operation: send PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_POID</td>
<td>POID</td>
<td>0.0.0.1/service/telco/gsm/telephony</td>
</tr>
<tr>
<td>PIN_FLD_PROGRAM_NAME</td>
<td>STR</td>
<td>&quot;NCC_BCD_Client&quot;</td>
</tr>
<tr>
<td>PIN_FLD_OBJ_TYPE</td>
<td>STR</td>
<td>&quot;gsm/ncc&quot;</td>
</tr>
<tr>
<td>PIN_FLD_EXTENDED_INFO</td>
<td>SUBSTRUCT</td>
<td>allocated 20, used 2</td>
</tr>
<tr>
<td>PIN_FLD_GSM_INFO</td>
<td>SUBSTRUCT</td>
<td>allocated 20, used 3</td>
</tr>
<tr>
<td>PIN_FLD_DIRECTION</td>
<td>ENUM</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CELL_ID</td>
<td>STR</td>
<td>&quot;000c&quot;</td>
</tr>
<tr>
<td>PIN_FLD_LOC_AREA_CODE</td>
<td>STR</td>
<td>&quot;064001000f&quot;</td>
</tr>
<tr>
<td>PIN_FLD_MSID</td>
<td>STR</td>
<td>&quot;004085752159&quot;</td>
</tr>
<tr>
<td>PIN_FLD_FLAGS</td>
<td>INT</td>
<td>4</td>
</tr>
</tbody>
</table>

#### Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_POID</td>
<td>POID</td>
<td>0.0.0.1/balance_group 114087 1073</td>
</tr>
<tr>
<td>PIN_FLD_ACCOUNT_OBJ</td>
<td>POID</td>
<td>0.0.0.1/account 114343 0</td>
</tr>
<tr>
<td>PIN_FLD_BILLINFO_OBJ</td>
<td>POID</td>
<td>0.0.0.1/billinfo 111015 0</td>
</tr>
<tr>
<td>PIN_FLD_EFFECTIVE_T</td>
<td>TSTAMP</td>
<td>(1362297600) Sun Mar 03 08:00:00 2013</td>
</tr>
<tr>
<td>PIN_FLD_BALANCES</td>
<td>ARRAY</td>
<td>1978 allocated 20, used 12</td>
</tr>
<tr>
<td>PIN_FLD_RESERVED_AMOUNT</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_next_BAL</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CURRENT_BAL</td>
<td>DECIMAL</td>
<td>-3</td>
</tr>
<tr>
<td>PIN_FLD_SUB_BALANCES</td>
<td>ARRAY</td>
<td>0 allocated 20, used 12</td>
</tr>
<tr>
<td>PIN_FLD_CONTRIBUTOR_STR</td>
<td>STR</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_VALID_TO DETAILS</td>
<td>INT</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_VALID_FROM DETAILS</td>
<td>INT</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_VALID_FROM</td>
<td>TSTAMP</td>
<td>(1362297600) Sun Mar 03 08:00:00 2013</td>
</tr>
<tr>
<td>PIN_FLD_ROLLOVER_DATA</td>
<td>INT</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_GRANTOR_OBJ</td>
<td>POID</td>
<td>0.0.0.1/purchased_product 112231 0</td>
</tr>
<tr>
<td>PIN_FLD_STATUS</td>
<td>ENUM</td>
<td>1</td>
</tr>
<tr>
<td>PIN_FLD_FLAGS</td>
<td>INT</td>
<td>2</td>
</tr>
<tr>
<td>PIN_FLD_CURRENT_TOTAL</td>
<td>DECIMAL</td>
<td>-3</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_FLOOR</td>
<td>DECIMAL</td>
<td>NULL</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_LIMIT</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_THRESHOLDS</td>
<td>INT</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_THRESHOLDS_FIXED</td>
<td>STR</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_REALTIME_CNTR</td>
<td>INT</td>
<td>4</td>
</tr>
<tr>
<td>PIN_FLD_BALANCES</td>
<td>ARRAY</td>
<td>1000011 allocated 20, used 10</td>
</tr>
<tr>
<td>PIN_FLD_RESERVED_AMOUNT</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_next_BAL</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
</tbody>
</table>
PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
PIN_FLD_CURRENT_BAL DECIMAL [0] 6
PIN_FLD_SUB_BALANCES ARRAY [4] allocated 20, used 12
PIN_FLD_CONTRIBUTOR_STR STR [0] ""
PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
PIN_FLD_VALID_FROM TSTAMP [0] (1362528348) Wed Mar 06 00:05:48 2013
PIN_FLD_CURRENT_BAL DECIMAL [0] 6
PIN_FLD_DELAYED_BAL DECIMAL [0] 0
PIN_FLD_ROLLOVER_DATA INT [0] 0
PIN_FLD_GRANTOR_OBJ POID [0] 0.0.0.0 0 0
PIN_FLD_STATUS ENUM [0] 1
PIN_FLD_FLAGS INT [0] 2
PIN_FLD_CURRENT_TOTAL DECIMAL [0] 6
PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
PIN_FLD_BALANCES ARRAY [1000076] allocated 20, used 10
PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
PIN_FLD_NEXT_BAL DECIMAL [0] 0
PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
PIN_FLD_CURRENT_BAL DECIMAL [0] -125.37
PIN_FLD_SUB_BALANCES ARRAY [2] allocated 20, used 12
PIN_FLD_CONTRIBUTOR_STR STR [0] ""
PIN_FLD_VALID_TO TSTAMP [0] (1363939853) Fri Mar 22 08:10:53 2013
PIN_FLD_VALID_FROM TSTAMP [0] (1362297600) Sun Mar 03 08:00:00 2013
PIN_FLD_CURRENT_BAL DECIMAL [0] -125.37
PIN_FLD_NEXT_BAL DECIMAL [0] 0
PIN_FLD_DELAYED_BAL DECIMAL [0] 0
PIN_FLD_ROLLOVER_DATA INT [0] 0
PIN_FLD_GRANTOR_OBJ POID [0] 0.0.0.0 0 0
PIN_FLD_STATUS ENUM [0] 1
PIN_FLD_FLAGS INT [0] 2
PIN_FLD_CURRENT_TOTAL DECIMAL [0] -125.37
PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""

Operation: send PCM_OP_READ_FLDS (4)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /account 114343 0
0 PIN_FLD_CURRENCY INT [0] 0
0 PIN_FLD_STATUS ENUM [0] 0
0 PIN_FLD_CLOSE_WHEN_T TSTAMP [0] (0) <null>
0 PIN_FLD_MOD_T TSTAMP [0] (0) <null>
0 PIN_FLD_LAST_STATUS_T TSTAMP [0] (0) <null>

Result: received for operation PCM_OP_READ_FLDS (4)

0 PIN_FLD_POID POID [0] 0.0.0.1 /account 114343 46
0 PIN_FLD_CURRENCY INT [0] 978
0 PIN_FLD_STATUS ENUM [0] 10100
0 PIN_FLD_CLOSE_WHEN_T TSTAMP [0] (0) <null>
0 PIN_FLD_MOD_T TSTAMP [0] (1363075261) Tue Mar 12 08:01:01 2013
0 PIN_FLD_LAST_STATUS_T TSTAMP [0] (1363072273) Tue Mar 12 07:11:13 2013

Operation: Send PCM_OP_TCF_AAA_AUTHORIZE (4002)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1458363_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
Result: received for operation PCM_OP_TCF_AAA_AUTHORIZE (4002)

Operation: Send PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

Flags = 0

Result: received for operation PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

Operation: send PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

Flags = 0
Chapter 8: Usage Scenarios

Result: received for operation PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE (4026)

Operation: Send PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

Result: received for operation PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

Operation: Send PCM_OP_TCF_AAA_QUERY_BALANCE (4104)
Chapter 8

Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

0 PIN_FLD_POID            POID [0] 0.0.0.1 /balance_group 114087 1078
0 PIN_FLD_ACCOUNT_OBJ    POID [0] 0.0.0.1 /account 114343 0
0 PIN_FLD_BILINFO_OBJ POID [0] 0.0.0.1 /bilinfo 111015 0
0 PIN_FLD_EFFECTIVE_T  TSTAMP [0] (1362359921) Mon Mar 04 01:18:41 2013
0 PIN_FLD_BILINFO_OBJ   POID [0] 0.0.0.1 /bilinfo_group 114087
0 PIN_FLD_BILGROUP_OBJ   POID [0] 0.0.0.1 /bilgroup 114731 0
0 PIN_FLD_BILSTATUS_OBJ POID [0] 0.0.0.1 /bilstatus 114087 0
0 PIN_FLD_BALANCES      ARRAY [978] allocated 20, used 10
  1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  1 PIN_FLD_NEXT_BAL     DECIMAL [0] 0
  1 PIN_FLD_CURRENT_BAL  DECIMAL [0] -2.31
  1 PIN_FLD_SUB_BILANCES ARRAY [0] allocated 20, used 12
    2 PIN_FLD_CONTRIBUTOR_STR STR [0] "Present"
    2 PIN_FLD_VALID_TO   TSTAMP [0] (0) <null>
    2 PIN_FLD_VALID_FROM TSTAMP [0] (1362297600) Sun Mar 03 08:00:00 2013
    2 PIN_FLD_CURRENT_BAL DECIMAL [0] -2.31
    2 PIN_FLD_NEXT_BAL DECIMAL [0] 0
    2 PIN_FLD_DELAYED_BAL DECIMAL [0] 0
    2 PIN_FLD_ROLLOVER_DATA INT [0] 0
    2 PIN_FLD_GRANTOR_OBJ POID [0] 0.0.0.1 /purchased_product 112231 0
    2 PIN_FLD_STATUS         ENUM [0] 1
    2 PIN_FLD_FLAGS           INT [0] 2
  1 PIN_FLD_CURRENT_TOTAL DECIMAL [0] 6
  1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
  1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
  1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
  1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] 
0 PIN_FLD_REALTIME_CNTR INT [0] 4
0 PIN_FLD_BILGROUP_OBJ   POID [0] 0.0.0.1 /bilgroup 114731 0
0 PIN_FLD_BILSTATUS_OBJ POID [0] 0.0.0.1 /bilstatus 114087 0
0 PIN_FLD_BILINFO_OBJ   POID [0] 0.0.0.1 /bilinfo 111015 0
0 PIN_FLD_BILINFO_OBJ   POID [0] 0.0.0.1 /bilinfo_group 114087
0 PIN_FLD_BILGROUP_OBJ   POID [0] 0.0.0.1 /bilgroup 114731 0
0 PIN_FLD_BILSTATUS_OBJ POID [0] 0.0.0.1 /bilstatus 114087 0
0 PIN_FLD_BALANCES      ARRAY [1000011] allocated 20, used 10
  1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  1 PIN_FLD_NEXT_BAL     DECIMAL [0] 0
  1 PIN_FLD_CURRENT_BAL  DECIMAL [0] 6
  1 PIN_FLD_SUB_BILANCES ARRAY [4] allocated 20, used 12
    2 PIN_FLD_CONTRIBUTOR_STR STR [0] "Present"
    2 PIN_FLD_VALID_TO   TSTAMP [0] (0) <null>
    2 PIN_FLD_VALID_FROM TSTAMP [0] (1362528348) Wed Mar 06 00:05:48 2013
    2 PIN_FLD_CURRENT_BAL DECIMAL [0] 6
    2 PIN_FLD_NEXT_BAL DECIMAL [0] 0
    2 PIN_FLD_DELAYED_BAL DECIMAL [0] 0
    2 PIN_FLD_ROLLOVER_DATA INT [0] 0
    2 PIN_FLD_GRANTOR_OBJ POID [0] 0.0.0.0 0 0
    2 PIN_FLD_STATUS         ENUM [0] 1
    2 PIN_FLD_FLAGS           INT [0] 2
  1 PIN_FLD_CURRENT_TOTAL DECIMAL [0] 6
  1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
  1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
  1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
  1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] 
0 PIN_FLD_BILGROUP_OBJ   POID [0] 0.0.0.1 /bilgroup 114731 0
0 PIN_FLD_BILSTATUS_OBJ POID [0] 0.0.0.1 /bilstatus 114087 0
0 PIN_FLD_BILINFO_OBJ   POID [0] 0.0.0.1 /bilinfo 111015 0
0 PIN_FLD_BALANCES      ARRAY [1000076] allocated 20, used 10
  1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  1 PIN_FLD_NEXT_BAL     DECIMAL [0] 0
  1 PIN_FLD_CURRENT_BAL  DECIMAL [0] -125.37
  1 PIN_FLD_SUB_BILANCES ARRAY [2] allocated 20, used 12
PIN_FLD_CONTRIBUTOR_STR    STR [0] ""
PIN_FLD_VALID_TO     TSTAMP [0] (1363939853) Fri Mar 22 08:10:53 2013
PIN_FLD_VALID_FROM   TSTAMP [0] (1362297600) Sun Mar 03 08:00:00 2013
PIN_FLD_VALID_TO_DETAILS    INT [0] 0
PIN_FLD_VALID_FROM_DETAILS    INT [0] 0
PIN_FLD_CURRENT_BAL  DECIMAL [0] -125.37
PIN_FLD_NEXT_BAL     DECIMAL [0] 0
PIN_FLD_DELAYED_BAL  DECIMAL [0] 0
PIN_FLD_ROLLOVER_DATA INT [0] 0
PIN_FLD_GRANTOR_OBJ    POID [0] 0.0.0.0  0 0
PIN_FLD_STATUS         ENUM [0] 1
PIN_FLD_FLAGS           INT [0] 2
PIN_FLD_CURRENT_TOTAL DECIMAL [0] -125.37
PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
PIN_FLD_CREDIT_THRESHOLDS    INT [0] 0
PIN_FLD_CREDIT_THRESHOLDS_FIXED    STR [0] ""

Operation: Send PCM_OP_READ_FLDS (4)

Flags = 0
0 PIN_FLD_POID            POID [0] 0.0.0.1 /account 114343 0
0 PIN_FLD_CURRENCY        INT [0] 0
0 PIN_FLD_STATUS         ENUM [0] 0
0 PIN_FLD_CLOSE_WHEN_T TSTAMP [0] (0) <null>
0 PIN_FLD_MOD_T        TSTAMP [0] (1363075261) Tue Mar 12 08:01:01 2013
0 PIN_FLD_LAST_STATUS_T TSTAMP [0] (0) <null>

Result: received for operation PCM_OP_READ_FLDS (4)

0 PIN_FLD_POID            POID [0] 0.0.0.1 /account 114343 46
0 PIN_FLD_CURRENCY        INT [0] 978
0 PIN_FLD_STATUS         ENUM [0] 10100
0 PIN_FLD_CLOSE_WHEN_T TSTAMP [0] (0) <null>
0 PIN_FLD_MOD_T        TSTAMP [0] (1363075261) Tue Mar 12 08:01:01 2013
0 PIN_FLD_LAST_STATUS_T TSTAMP [0] (1363072273) Tue Mar 12 07:11:13 2013
SMS charging with named event reservation

SMS charging with named event reservation flow
Here is an example message flow that illustrates the message sequence that occurs for SMS charging with a named event reservation.

SMS charging with named event reservation scenario
This scenario describes the sequence of messages that occurs when the subscriber sends a short message (SMS) from a GSM phone and a named event reservation is used and BRM is used for charging.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | • Subscriber sends an SMS from a GSM phone.  
      • MSC sends MOForwardSM to xmsTrigger. |
Step | Action
---|---
2 | xmsTrigger sends InitialDP to the slee_acs process.
   - The slee_acs process receives the InitialDP and passes it to the CCS service loader.
   - The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. It also determines the control plan to run.
   - The slee_acs process runs the control plan.
   - The slee_acs process reaches a Named Event feature node specifying Reserve Event, name of event = SMS.
   - The Named Event feature node invokes the NamedEventReservation action on the BCD actions library.
   - The BCD actions library constructs a FlistSleeEvent containing PCM_OP_TCF_AAA_AUTHORIZE, with quantity set to 1.
   - The BCD Client finds a free connection on the least busy connections manager.
3 | The BCD Client invokes the PCM_OP_TCF_AAA_AUTHORIZE operation and sets a timer to the configured value for this type of operation.
4 | BRM responds, indicating that the SMS has been successfully reserved.
   - The BCD Client receives the output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.
   - The BCD actions library takes the output flist and translates it into the response to the NamedEventReservation action.
   - The control plan reaches an Attempt Delivery to Pending (ADP) feature node.
5 | The ADP feature node sends RequestReportBCSMEvent, Connect operations to xmsTrigger.
6 | xmsTrigger sends MTForwardSM to the MSC serving the called subscriber.
7 | MSC successfully delivers SMS to the called subscriber and sends a success result of MTForwardSM to xmsTrigger.
8 | xmsTrigger sends EventReportBCSM(oDisconnect) to the slee_acs process.
9 | xmsTrigger sends a success result of MOForwardSM to the originating MSC.
   - The slee_acs process reaches a Named Event feature node specifying ConfirmEvent, name of event = SMS.
   - The BillableEvent feature node invokes the ConfirmNamedEventReservation action on the BCD actions library.
   - The BCD actions library constructs an FlistSleeEvent for a PCM_OP_TCF_AAA_ACCOUNTING operation.
10 | The BCD Client invokes the PCM_OP_TCF_AAA_ACCOUNTING operation and sets a timer to the configured value for this type of operation.
11 | BRM responds to the operation, indicating that the SMS has been charged successfully.
   - The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.
   - The BCD actions library translates the output flist into the response to the ConfirmNamedEventReservation action.
   - The slee_acs process reaches an end node and clears the call context.
Messages: SMS charging with named event reservation
The following messages include operations sent to BRM and results returned by BRM for SMS charging with a named event reservation. The general message format is: nesting level (0; 1, or 2); field; data type; value.

Operation: send is PCM_OP_TCF_AAA_AUTHORIZE (4002)

Flags - 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/sms -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_ne_1393296_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2 PIN_FLD_DIRECTION ENUM [0] 0
2 PIN_FLD_CELL_ID STR [0] "000c"
2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
1 10000 SUBSTRUCT [0] allocated 20, used 3
2 10001 STR [0] "Present"
2 PIN_FLD_LOCATION STR [0] "004085752159"
2 10002 INT [0] 1
0 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_REQ_MODE ENUM [0] 8

Result: received for operation PCM_OP_TCF_AAA_AUTHORIZE (4002)

0 PIN_FLD_POID POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 177146 0
0 PIN_FLD_EXPIRATION_T TSTAMP [0] (1363158022) Wed Mar 13 07:00:22 2013
0 PIN_FLD_QUANTITY DECIMAL [0] 1
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/sms 112359 17
0 PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1 /reservation 179194 0
0 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 114087 841
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
1 PIN_FLD_AMOUNT DECIMAL [0] 0.200
1 PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 8.400
0 PIN_FLD_RESULT ENUM [0] 1
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_ne_1393296_0"
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 114343 0

Operation: send is PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

Flags - 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/sms -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_ne_1393296_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2 PIN_FLD_DIRECTION ENUM [0] 0
2 PIN_FLD_CELL_ID STR [0] "000c"
2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
1 10000 SUBSTRUCT [0] allocated 20, used 3
2 10001 STR [0] "Present"
2 PIN_FLD_LOCATION STR [0] "004085752159"
2 10002 INT [0] 1
0 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_REQ_MODE ENUM [0] 8

Result: received for operation PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

0 PIN_FLD_POID POID [0] 0.0.0.1 /activity/session/telco/gsm/ncc 27753427036884986 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_ne_1393296_0"
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 114343 0
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/sms 112359 17
0 PIN_FLD_RATING_STATUS ENUM [0] 0
SMS charging with named event reservation fail

SMS charging with named event reservation fail flow
Here is an example message flow that illustrates the message sequence that occurs for SMS charging with a named event reservation that fails.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | - Subscriber sends an SMS from a GSM phone.  
<pre><code>  | - MSC sends MOForwardSM to xmsTrigger.      |
</code></pre>
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 2    | - xmsTrigger sends InitialDP to the slee_acs process.  
- The slee_acs process receives the InitialDP and passes it to the CCS service loader.  
- The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. It also determines the control plan to run.  
- The slee_acs process runs the control plan.  
- The slee_acs process reaches a Named Event feature node specifying Reserve Event, name of event = SMS.  
- The Named Event feature node invokes the NamedEventReservation action on the BCD actions library.  
- The BCD actions library constructs a FlistSleeEvent containing PCM_OP_TCF_AAA_AUTHORIZE, with quantity set to 1.  
- The BCD Client finds a free connection on the least busy connections manager. |
| 3    | - The BCD Client invokes the PCM_OP_TCF_AAA_AUTHORIZE operation and sets a timer to the configured value for this type of operation. |
| 4    | - BRM responds, indicating that the SMS has been successfully reserved.  
- The BCD Client receives the output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
- The BCD actions library takes the output flist and translates it into the response to the NamedEventReservation action.  
- The control plan reaches an Attempt Delivery to Pending (ADP) feature node. |
| 5    | - The ADP feature node sends RequestReportBCSMEvent, Connect operations to xmsTrigger. |
| 6    | - xmsTrigger sends SendRoutingInfoForSMS to HLR to find the location of the destination handset |
| 7    | - HLR sends SendRoutingInfoForSMS error to xmsTrigger because destination number is invalid. |
| 8    | - xmsTrigger sends EventReportBCSM(RouteSelectFailure) to slee_acs. |
| 9    | - slee_acs sends ReleaseCall to xmsTrigger. |
| 10   | - xmsTrigger sends a MOForwardSM error to the originating MSC.  
- slee_acs takes the Delivery Failure branch of the ADP node and reaches a Named Event feature node.  
- The Named Event feature node is configured to revoke 1 event of type SMS. It invokes a RevokeNamedEvent action on the BCD actions library.  
- BCD actions library constructs an event (FlistSleeEvent) containing PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION. |
| 11   | - The BCD Client invokes the PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION operation and sets a timer to the configured value for this type of operation. |
| 12   | - BRM responds, indicating that the cost of the SMS has been refunded successfully.  
- The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the connection as free for reuse and cancels the operation timer.  
- The BCD actions library takes the output flist and translates it into the response to the RevokeNamedEvent action.  
- The slee_acs process reaches an end node and clears the call context. |
Messages: SMS charging with named event reservation fail
The following messages include operations sent to BRM and results returned by BRM for SMS charging that fails. The operation uses a named event reservation. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation:** send is PCM_OP_TCF_AAA_AUTHORIZE (4002)

<table>
<thead>
<tr>
<th>Flags</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_POID</td>
<td>POID [0] 0.0.0.1 /service/telco/gsm/sms -1 0</td>
</tr>
<tr>
<td>PIN_FLD_PROGRAM_NAME</td>
<td>STR [0] &quot;NCC_BCD_Client&quot;</td>
</tr>
<tr>
<td>PIN_FLD_SESSION_ID</td>
<td>INT [0] 0</td>
</tr>
<tr>
<td>PIN_FLD_AUTHORIZATION_ID</td>
<td>STR [0] &quot;brmClient-tbalcomb-2013-3-12-6_ne_1394297_0&quot;</td>
</tr>
<tr>
<td>PIN_FLD_OBJ_TYPE</td>
<td>STR [0] &quot;gsm/ncc&quot;</td>
</tr>
<tr>
<td>PIN_FLD_CALLING_NUMBER</td>
<td>STR [0] &quot;004085752159&quot;</td>
</tr>
<tr>
<td>PIN_FLD_CALLED_NUMBER</td>
<td>STR [0] &quot;555873900000&quot;</td>
</tr>
<tr>
<td>PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_DIRECTION ENUM [0] 0</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_CELL_ID STR [0] &quot;000c&quot;</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_LOC_AREA_CODE STR [0] &quot;064001000f&quot;</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_MSID STR [0] &quot;004085752159&quot;</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_REQ_MODE ENUM [0] 8</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 114087 846</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2</td>
<td></td>
</tr>
</tbody>
</table>

| PIN_FLD_AMOUNT DECIMAL [0] 1.600 |
| PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 8.400 |
| PIN_FLD_RATING_STATUS ENUM [0] 0 |
| PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_ne_1394297_0" |
| PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 114343 0 |

**Result:** received for operation PCM_OP_TCF_AAA_AUTHORIZE (4002)

<table>
<thead>
<tr>
<th>Flags</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_POID</td>
<td>POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 173317 0</td>
</tr>
<tr>
<td>PIN_FLD_EXPIRATION_T PSTAMP [0] (1363158099) Wed Mar 13 07:01:39 2013</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/sms 112359 17</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1 /reservation 175365 0</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 114087 846</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2</td>
<td></td>
</tr>
</tbody>
</table>

| PIN_FLD_AMOUNT DECIMAL [0] 1.600 |
| PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 8.400 |
| PIN_FLD_RATING_STATUS ENUM [0] 0 |
| PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_ne_1394297_0" |

**Operation:** send PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION (4004)

<table>
<thead>
<tr>
<th>Flags</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_POID</td>
<td>POID [0] 0.0.0.1 /service/telco/gsm/sms -1 0</td>
</tr>
<tr>
<td>PIN_FLD_PROGRAM_NAME</td>
<td>STR [0] &quot;NCC_BCD_Client&quot;</td>
</tr>
<tr>
<td>PIN_FLD_SESSION_ID</td>
<td>INT [0] 1</td>
</tr>
<tr>
<td>PIN_FLD_AUTHORIZATION_ID</td>
<td>STR [0] &quot;brmClient-tbalcomb-2013-3-12-6_ne_1394297_0&quot;</td>
</tr>
<tr>
<td>PIN_FLD_OBJ_TYPE</td>
<td>STR [0] &quot;gsm/ncc&quot;</td>
</tr>
<tr>
<td>PIN_FLD_CALLING_NUMBER</td>
<td>STR [0] &quot;004085752159&quot;</td>
</tr>
<tr>
<td>PIN_FLD_CALLED_NUMBER</td>
<td>STR [0] &quot;555873900000&quot;</td>
</tr>
<tr>
<td>PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_DIRECTION ENUM [0] 0</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_CELL_ID STR [0] &quot;000c&quot;</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_LOC_AREA_CODE STR [0] &quot;064001000f&quot;</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_MSID STR [0] &quot;004085752159&quot;</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_REQ_MODE ENUM [0] 8</td>
<td></td>
</tr>
</tbody>
</table>

**Result:** received for operation PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION (4004)
SMS charging with direct named event - insufficient funds

SMS charging with direct named event - insufficient funds flow
Here is an example message flow that illustrates the message sequence that occurs for SMS charging with a direct named event that where the subscriber has insufficient funds.

SMS charging with direct named event - Insufficient Funds scenario
This scenario describes the sequence of messages that occurs when the subscriber sends a short message (SMS) from a GSM phone and BRM is used for charging but the subscriber has insufficient funds.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | • The subscriber sends an SMS from a GSM phone.  
       • MSC sends MOForwardSM to xmsTrigger. |
| 2    | • xmsTrigger sends InitialDP to slee_acs. |
Chapter 8

Usage Scenarios

3  The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.
   The CCS service loader looks up the subscriber and wallet in the SLC database and determines that the wallet information is on the BRM domain. It also determines the control plan to run.
   The slee_acs process runs the control plan.
   The slee_acs process reaches a Billable Event feature node specifying Direct Event, name of event = SMS.
   The Billable Event feature node invokes the DirectNamedEvent action on the BCD actions library.
   The BCD actions library constructs an FlistSleeEvent containing PCM_OP_TCF_AAA_ACCOUNTING, with quantity set to 1.

4  BCD Billing client finds a free connection on the least busy Connection Manager, which is the one with the lowest proportion of its connections currently in use.
   BCD Billing Client invokes the PCM_OP_TCF_AAA_ACCOUNTING operation and starts a timer for the configured value for this type of operation.
   BRM responds to the operation with PIN_ERR=26 (insufficient funds) indicating that 0.5 SMS have been charged.
   The BCD Billing Client receives the operation output flist and sends it in a BcdSleeEvent to slee_acs. It also marks the BRM connection as free for reuse and cancels the operation timer.

5  The BCD actions library sends PCM_OP_TCF_AAA_REFUND to the BCD Billing Client to reverse the charge for the last PCM_OP_TCF_AAA_ACCOUNTING operation.

6  The BCD Billing client finds a free connection on the least busy Connection Manager, which is the one with the lowest proportion of its connections currently in use.
   The BCD Billing Client invokes the PCM_OP_TCF_AAA_REFUND operation and starts a timer for the configured value for this operation type.
   BRM responds to the operation.
   The BCD actions library constructs a failure response of the DirectNamedEvent action.
   The control plan exits the No Credit branch of the Billable Event feature node and reaches a Disconnect feature node.

7  The Disconnect feature node sends a ReleaseCall operation to xmsTrigger.

8  xmsTrigger sends a failure result of MOForwardSM to the originating MSC.

9  The slee_acs process reaches an end feature node and clears the call context.

Messages: SMS charging with direct named event - insufficient funds
The following messages include operations sent to BRM and results returned by BRM for an SMS message for which there is insufficient funds. The general message format is: nesting level (0; 1, or 2); field; data type; value.

Operation: send is PCM_OP_TCF_AAA_ACCOUNTING (4012)

Flags = 0
0 PIN_FLD_POID           POID [0] 0.0.0.1 /service/telco/gsm/sm -1 0
0 PIN_FLD_PROGRAM_NAME    STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID      INT [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-cleejoh-2013-3-12-4_dne_1274367_0"
0 PIN_FLD_OBJ_TYPE        STR [0] "gsm/ncc"
Chapter 8

0 PIN_FLD_CALLING_NUMBER STR [0] "004085752153"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1  PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2  PIN_FLD_DIRECTION ENUM [0] 0
2  PIN_FLD_LOC_AREA_CODE STR [0] "004010001f"
1 100000 SUBSTRUCT [0] allocated 20, used 3
2 10001 STR [0] "Present"
2  PIN_FLD_LOCATION STR [0] "004085752153"
2 10002 INT [0] 8
0 PIN_FLD_MSID STR [0] "004085752153"
0 PIN_FLD_MODE ENUM [0] 1

Result: received for operation PCM_OP_TCF_AAA_ACCOUNTING (4012)

0 PIN_FLD_POID POID [0] 0.0.0.1 /event/activity/telco/gsm/ncc 27751673485089065 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-cleejoh-2013-3-12-4_dne_1274367_0"
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 109399 0
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/sms 108343 20
0 PIN_FLD_RATING_STATUS ENUM [0] 26
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
1  PIN_FLD_AMOUNT DECIMAL [0] 0.050
0 PIN_FLD_QUANTITY DECIMAL [0] 0.25000000
0 PIN_FLD_RUM_NAME STR [0] "Number Of Events"
0 PIN_FLD_RESULT ENUM [0] 1
0 PIN_FLD_REASON ENUM [0] 3

Operation: send is PCM_OP_TCF_AAA_REFUND (4102)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/sms -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-cleejoh-2013-3-12-4_dne_1274367_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752153"
0 PIN_FLD_CALLED_NUMBER STR [0] "55587390000"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1  PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2  PIN_FLD_DIRECTION ENUM [0] 0
2  PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
1 100000 SUBSTRUCT [0] allocated 20, used 3
2 10001 STR [0] "Present"
2  PIN_FLD_LOCATION STR [0] "004085752153"
2 10002 INT [0] 8
0 PIN_FLD_MSID STR [0] "004085752153"
0 PIN_FLD_REQ_MODE ENUM [0] 8
0 PIN_FLD_ADJUSTMENT_INFO ARRAY [0] allocated 20, used 2
1  PIN_FLD_RESOURCE_ID INT [0] 978
1  PIN_FLD_PERCENT DECIMAL [0] 100.000000000000000

Result: received for operation PCM_OP_TCF_AAA_REFUND (4102)

0 PIN_FLD_POID POID [0] 0.0.0.1 /event/activity/telco/gsm/ncc 27751673485089065 0
0 PIN_FLD_RESULT ENUM [0] 1
0 PIN_FLD_ADJUSTMENT_INFO ARRAY [0] allocated 20, used 6
1  PIN_FLD_RESOURCE_ID INT [0] 978
1  PIN_FLD_AMOUNT DECIMAL [0] 0.05
1  PIN_FLD_AMOUNT_ADJUSTED DECIMAL [0] 0.050
1  PIN_FLD.Amount_TAX_ADJUSTED DECIMAL [0] 0
1  PIN_FLD_AMOUNT_TAXED DECIMAL [0] 0
1  PIN_FLD_DISCOUNT DECIMAL [0] 0
0 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 109783 1178
0 PIN_FLD_ITEM_OBJ POID [0] 0.0.0.1 /item/adjustment 182664 0
0 PIN_FLD_BAL_GROUPS_OBJ POID [0] 0.0.0.1 /account 109399 0
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/sms 108343 20
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-cleejoh-2013-3-12-4_dne_1274367_0"
0 PIN_FLD_EVENT_OBJ POID [0] 0.0.0.1 /event/billing/adjustment/event 277516734850844040 0
Successful USSD recharge

Successful USSD recharge flow
Here is an example message flow that illustrates the message sequence that occurs for a successful USSD recharge.

Successful USSD recharge (BRM Vouchers)

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | The calling subscriber sends a USSD message, containing the service code for a voucher recharge, the voucher number and PIN.  
     | • MSC sends processUnstructuredSS-Request to USSD Gateway. |
| 2    | • USSD Gateway sends InitialDP to the slee_acs process, using the USSD voucher recharge service key as determined by the service code configuration.  
     | • The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information in on the BRM domain. It also determines the control plan to run, using the serviceKey to select the USSD voucher recharge control plan.  
     | • The slee_acs process runs the control plan.  
     | • The slee_acs process reaches a VoucherRecharge feature node. |
Chapter 8

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>The VoucherRecharge feature node sends ConnectToResource, PromptAndCollectUserInformation to USSD Gateway, instructing it to prompt the caller for voucher number and PIN.</td>
</tr>
<tr>
<td></td>
<td>USSD Gateway retrieves the voucher number and PIN from the received processUnstructuredSS_Request, without further interaction with the MSC or subscriber.</td>
</tr>
<tr>
<td>4</td>
<td>MSC sends PromptAndCollectUserInformation result, containing a voucher number and PIN to the slee_acs process.</td>
</tr>
<tr>
<td></td>
<td>The VoucherRecharge feature node invokes the VoucherRedeem action on the BCD actions library.</td>
</tr>
<tr>
<td>5</td>
<td>The BCD actions library creates an event (FlistSleeEvent) that contains a PCM_OP_PYMT_TOPUP operation and sends it to the BCD Client.</td>
</tr>
<tr>
<td></td>
<td>The BCD Client invokes the PCM_OP_PYMT_TOPUP operation and sets a timer to the configured value for this type of operation.</td>
</tr>
<tr>
<td>6</td>
<td>BRM responds to the operation, indicating that the account has been successfully recharged.</td>
</tr>
<tr>
<td></td>
<td>The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.</td>
</tr>
<tr>
<td></td>
<td>The BCD actions library translates the output flist into the response of the VoucherRedeem action.</td>
</tr>
<tr>
<td>7</td>
<td>The Voucher Recharge feature node sends PlayAnnouncement to the USSD Gateway.</td>
</tr>
<tr>
<td></td>
<td>USSD Gateway translates the announcement ID to some text stating that the recharge was successful, based on configuration.</td>
</tr>
<tr>
<td>8</td>
<td>USSD Gateway sends processUnstructuredSS-Request result to the MSC.</td>
</tr>
<tr>
<td></td>
<td>USSD text stating that the recharge was successful is displayed on the caller’s handset.</td>
</tr>
<tr>
<td>9</td>
<td>USSD Gateway sends SpecializedResourceReport to the slee_acs process.</td>
</tr>
<tr>
<td>10</td>
<td>The control plan reaches an end node and ACS sends DisconnectForwardConnection, ReleaseCall to USSD Gateway.</td>
</tr>
</tbody>
</table>

Messages: successful USSD recharge
The following messages include operations sent to BRM and results returned by BRM for a successful recharge using USSD. The general message format is: nesting level (0; 1, or 2); field; data type; value.

Operation: send is PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

Flags - 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
  1   PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
  2   PIN_FLD_DIRECTION ENUM [0] 0
  2   PIN_FLD_CELL_ID STR [0] "000c"
  2   PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
  1  10000   SUBSTRUCT [0] allocated 20, used 2
  2   10001   STR [0] "Present"
  2   PIN_FLD_LOCATION STR [0] "004085752160"
0 PIN_FLD_MSID STR [0] "004085752160"
0 PIN_FLD_FLAGS INT [0] 4

Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

0 PIN_FLD_POID POID [0] 0.0.0.1 /balance_group 127667 6440
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_ACCOUNT_OBJ</td>
<td>POID</td>
<td>0.0.0.1 /account 128819 0</td>
</tr>
<tr>
<td>PIN_FLD_BILLINFO_OBJ</td>
<td>POID</td>
<td>0.0.0.1 /billinfo 129203 0</td>
</tr>
<tr>
<td>PIN_FLD_EFFECTIVE_T</td>
<td>TSTAMP</td>
<td>(1375375354) Sun May 20 23:42:34 2012</td>
</tr>
<tr>
<td>PIN_FLD_SUB_BALANCES</td>
<td>ARRAY</td>
<td>[978] allocated 20, used 10</td>
</tr>
<tr>
<td>PIN_FLD_RESERVED_AMOUNT</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_NEXT_BAL</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CURRENT_BAL</td>
<td>DECIMAL</td>
<td>400</td>
</tr>
<tr>
<td>PIN_FLD_SUB_BALANCES</td>
<td>ARRAY</td>
<td>[0] allocated 20, used 12</td>
</tr>
<tr>
<td>PIN_FLD_CONTRIBUTOR_STR</td>
<td>STR</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_CURRENT_TOTAL</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_FLOOR</td>
<td>DECIMAL</td>
<td>NULL</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_LIMIT</td>
<td>DECIMAL</td>
<td>500</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_THRESHOLDS</td>
<td>INT</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_THRESHOLDS_FIXED</td>
<td>STR</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_REALTIME_CNTR</td>
<td>INT</td>
<td>4</td>
</tr>
<tr>
<td>PIN_FLD_BALANCES</td>
<td>ARRAY</td>
<td>[1000011] allocated 20, used 10</td>
</tr>
<tr>
<td>PIN_FLD_RESERVED_AMOUNT</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_NEXT_BAL</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CURRENT_BAL</td>
<td>DECIMAL</td>
<td>70</td>
</tr>
<tr>
<td>PIN_FLD_SUB_BALANCES</td>
<td>ARRAY</td>
<td>[4] allocated 20, used 12</td>
</tr>
<tr>
<td>PIN_FLD_CURRENT_TOTAL</td>
<td>DECIMAL</td>
<td>70</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_FLOOR</td>
<td>DECIMAL</td>
<td>NULL</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_LIMIT</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_THRESHOLDS</td>
<td>INT</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_THRESHOLDS_FIXED</td>
<td>STR</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_BALANCES</td>
<td>ARRAY</td>
<td>[1000076] allocated 20, used 10</td>
</tr>
<tr>
<td>PIN_FLD_RESERVED_AMOUNT</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_NEXT_BAL</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CONSUMPTION_RULE</td>
<td>ENUM</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CURRENT_BAL</td>
<td>DECIMAL</td>
<td>125</td>
</tr>
<tr>
<td>PIN_FLD_SUB_BALANCES</td>
<td>ARRAY</td>
<td>[2] allocated 20, used 12</td>
</tr>
<tr>
<td>PIN_FLD_CURRENT_TOTAL</td>
<td>DECIMAL</td>
<td>125</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_FLOOR</td>
<td>DECIMAL</td>
<td>NULL</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_LIMIT</td>
<td>DECIMAL</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_THRESHOLDS</td>
<td>INT</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_THRESHOLDS_FIXED</td>
<td>STR</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_BALANCES</td>
<td>ARRAY</td>
<td>[1000076] allocated 20, used 10</td>
</tr>
</tbody>
</table>
**PIN_FLD_CREDIT_THRESHOLDS** INT [0] 0
**PIN_FLD_CREDIT_THRESHOLDS_FIXED** STR [0] ""

**Operation:** send is PCM_OP_PYMT_TOPUP (3726)

Flags = 0

- **PIN_FLD_POID** POID [0] 0.0.0.1 /account 128819 0
- **PIN_FLD_PROGRAM_NAME** STR [0] "NCC_BCD_Client"
- **PIN_FLD_OBJ_TYPE** STR [0] "gsm/ncc"
- **PIN_FLD_INHERITED_INFO** SUBSTRUCT [0] allocated 20, used 1
- **PIN_FLD_VOUCHERS_INFO** ARRAY [0] allocated 20, used 4
  - **PIN_FLD DEVICE ID** STR [0] "1000250212"
  - **PIN_FLD_VOUCHER_PIN** STR [0] "0099"
  - **PIN_FLD_BILLINFO_OBJ** POID [0] 0.0.0.1 /billinfo 129203 0
  - **PIN_FLD_BAL_GRP_OBJ** POID [0] 0.0.0.1 /balance_group 127667 6440

Result: received for operation PCM_OP_PYMT_TOPUP (3726)

- **PIN_FLD_POID** POID [0] 0.0.0.1 /account 128819 0
- **PIN_FLD_RESULTS** ARRAY [0] allocated 63, used 63
  - **PIN_FLD_CREATED_T** TSTAMP [0] (1339095022) Thu Jun 07 18:50:22 2012
  - **PIN_FLD_MOD_T** TSTAMP [0] (1339095022) Thu Jun 07 18:50:22 2012
  - **PIN_FLD_DESCR** STR [0] ""
Chapter 8

1     PIN_FLD_UNRATED_QUANTITY    DECIMAL [0] 0
1     PIN_FLD_USAGE_TYPE        STR [0] ""  
1     PIN_FLD_USERID            POID [0] 0.0.0.1 /servicepcm_client 1 33918
1     PIN_FLD_SUB_BAL_IMPACTS    ARRAY [0] allocated 20, used 3
2         PIN_FLD_BAL_GRP_OBJ    POID [0] 0.0.0.1 /balance_group 127667 0
2     PIN_FLD_RESOURCE_ID        INT [0] 978
2     PIN_FLD_SUB_BALANCES       ARRAY [0] allocated 20, used 8
3     PIN_FLD_AMOUNT            DECIMAL [0] -50
3     PIN_FLD_CONTRIBUTOR_STR   STR [0] ""  
3     PIN_FLD_GRANTOR_OBJ       POID [0] 0.0.0.0 0 0
3     PIN_FLD_ROLLOVER_DATA     INT [0] 0
3     PIN_FLD_VALID_FROM        TSTAMP [0] (0) <null>
3     PIN_FLD_VALID_FROM_DETAILS INT [0] 0
3     PIN_FLD_VALID_TO          TSTAMP [0] (0) <null>
3     PIN_FLD_VALID_TO_DETAILS  INT [0] 0
1     PIN_FLD_BAL_IMPACTS       ARRAY [0] allocated 22, used 22
2         PIN_FLD_ACCOUNT_OBJ    POID [0] 0.0.0.1 /account 128819 317
2     PIN_FLD_AMOUNT            DECIMAL [0] -50
2     PIN_FLD_AMOUNT_DEFERRED   DECIMAL [0] 0
2     PIN_FLD_AMOUNT_ORIG       DECIMAL [0] 0
2     PIN_FLD_BAL_GRP_OBJ       POID [0] 0.0.0.1 /balance_group 127667 6440
2     PIN_FLD_DISCOUNT          DECIMAL [0] 0
2     PIN_FLD_DISCOUNT_INFO     STR [0] ""
2     PIN_FLD_GL_ID             INT [0] 0
2     PIN_FLD_IMPACT_CATEGORY   STR [0] "default"
2     PIN_FLD_IMPACT_TYPE       ENUM [0] 2
2     PIN_FLD_ITEM_OBJ          POID [0] 0.0.0.1 /item/payment 174950 0
2     PIN_FLD_LINEAGE           STR [0] ""
2     PIN_FLD_OFFERING_OBJ      POID [0] 0.0.0.0 0 0
2     PIN_FLD_PERCENT           DECIMAL [0] 1
2     PIN_FLD_PRODUCT_OBJ       POID [0] 0.0.0.1 /product 82729 6
2     PIN_FLD_QUANTITY          DECIMAL [0] 1
2     PIN_FLD_RATE_OBJ          POID [0] 0.0.0.1 /rate 79195 1
2     PIN_FLD_RATE_TAG          STR [0] "Rate 1"
2     PIN_FLD_Resource_ID        INT [0] 978
2     PIN_FLD_Resource_ID_ORIG  INT [0] 0
2     PIN_FLD_RUM_ID            INT [0] 0
2     PIN_FLD_TAX_CODE          STR [0] ""
1     PIN_FLD_EVENT_MISC_DETAILS ARRAY [0] allocated 20, used 2
2     PIN_FLD_REASON_DOMAIN_ID  INT [0] 100
2     PIN_FLD_REASON_ID         INT [0] 0
1     PIN_FLD_PAYMENT          SUBSTRUCT [0] allocated 20, used 13
2     PIN_FLD_ACCOUNT_NO       STR [0] ""
2     PIN_FLD_ACH              INT [0] 0
2     PIN_FLD_AMOUNT            DECIMAL [0] 50
2     PIN_FLD_AMOUNT_ORIGINAL_PAYMENT DECIMAL [0] 0
2     PIN_FLD_BILL_NO          STR [0] ""
2     PIN_FLD_BILL_ID           INT [0] 0
2     PIN_FLD_CHANNEL_ID       INT [0] 0
2     PIN_FLD_COMMAND          ENUM [0] 0
2     PIN_FLD_CURRENCY         INT [0] 978
2     PIN_FLD_MERCHANT         STR [0] ""
2     PIN_FLD_PAY_TYPE         ENUM [0] 10016
2     PIN_FLD_STATUS           ENUM [0] 0
2     PIN_FLD_SUB_TRANS_ID     STR [0] ""  
2     PIN_FLD_TRANS_ID         STR [0] "T1,48,0"
1     PIN_FLD_VOUCHERS_INFO    ARRAY [0] allocated 20, used 5
2     PIN_FLD_CARD_EXPIRATION  INT [0] 0
2     PIN_FLD DEVICE_ID        STR [0] "1000250212"
2     PIN_FLD_VALID_FROM       TSTAMP [0] (0) <null>
2     PIN_FLD_VALID_TO         TSTAMP [0] (0) <null>
2     PIN_FLD_VOUCHER_PIN      STR [0] "0099"
1     PIN_FLD_RESULT          ENUM [0] 1
1     PIN_FLD_SELECT_RESULT   ENUM [0] 0
1     PIN_FLD_SELECT_STATUS   INT [0] 4
1     PIN_FLD_ITEM_NO         STR [0] "F1-72"
0 PIN_FLD_VOUCHERS_INFO     ARRAY [0] allocated 20, used 6
1     PIN_FLD_BILLINFO_OBJ    POID [0] 0.0.0.1 /billinfo 129203 0
1     PIN_FLD_DEVICE_ID       STR [0] "1000250212"
1     PIN_FLD_VOUCHER_PIN     STR [0] "0099"
1     PIN_FLD_EXPIRATION_T     TSTAMP [0] (2126174400) Sun May 17 12:00:00 2037
IVR recharge with invalid PIN

IVR recharge with invalid PIN flow
Here is an example message flow for an IVR recharge with an invalid PIN.

```
IVR recharge with invalid PIN flow (BRM Vouchers)
```

**Step** | **Action**
--- | ---
1 | The subscriber dials the voucher recharge number, which is toll free.
2 | MSC sends InitialDP to the slee_acs process with ServiceKey set to a special value that indicates voucher recharge.
3 | MSC prompts and collects user information.
4 | MSC collects digits.
5 | MSC prompts and collects user information.
6 | MSC plays announcement.
7 | MSC sends DisConnectForwardCall.
8 | MSC disconnects connection.

IVR recharge with invalid PIN scenario
This scenario describes the sequence of messages that occurs when the subscriber sends a USSD message requesting a voucher recharge and the subscriber's PIN is invalid. In this scenario both the account and the voucher are on BRM.

1. The subscriber dials the voucher recharge number, which is toll free.
2. MSC sends InitialDP to the slee_acs process with ServiceKey set to a special value that indicates voucher recharge.
### Step 2
- The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines the control plan to run, using the serviceKey to select the voucher recharge control plan.
- The slee_acs process runs the control plan.
- The Voucher Recharge feature node sends ConnectToResource, PromptAndCollectUserInformation to the MSC, instructing the MSC to prompt the caller for the voucher number and PIN.

### Step 3
- MSC plays the specified announcement to the caller and collects the voucher number and PIN.
- MSC sends PromptAndCollectUserInformation result, containing voucher number and PIN to the slee_acs process.

### Step 4
- The Voucher Recharge feature node invokes the VoucherRedeem action on the BCD actions library.
- The BCD actions library constructs an event (FlistSleeEvent) that contains PCM_OP_PYMT_TOPUP and sends it to the BCD Client.
- The BCD Client invokes the PCM_OP_PYMT_TOPUP operation and sets a timer to the configured value for this type of operation.

### Step 5
- The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.
- The BCD actions library translates the output flist into the error response of the VoucherRedeem action.

### Step 6
- The Voucher Recharge feature node sends PlayAnnouncement to the MSC.
- MSC plays an announcement to the caller, stating that the recharge was unsuccessful.

### Step 7
- MSC sends SpecializedResourceReport to the slee_acs process.

### Step 8
- The control plan reaches and end feature node and ACS sends DisconnectForwardConnection and ReleaseCall actions to the MSC and clears the call connection.
- The caller is disconnected.

## Messages: IVR recharge with invalid PIN

The following messages include operations sent to BRM and results returned by BRM for a recharge with an invalid PIN using IVR. The general message format is: nesting level (0; 1, or 2); field; data type; value.

### Operation: send is PCM.OP.TCF.AAA_QUERY_BALANCE (4104)

<table>
<thead>
<tr>
<th>Flags</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 PIN_FLD_POWD</td>
<td>POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0</td>
</tr>
<tr>
<td>0 PIN_FLD_PROGRAM_NAME</td>
<td>STR [0] &quot;NCC_BCD_Client&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_OBJ_TYPE</td>
<td>STR [0] &quot;gsm/ncc&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_EXTENDED_INFO</td>
<td>SUBSTRUCT [0] allocated 20, used 2</td>
</tr>
<tr>
<td>1 PIN_FLD_GSM INFO</td>
<td>SUBSTRUCT [0] allocated 20, used 3</td>
</tr>
<tr>
<td>2 PIN_FLD_DIRECTION</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>2 PIN_FLD_CELL ID</td>
<td>STR [0] &quot;000c&quot;</td>
</tr>
<tr>
<td>2 PIN_FLD_LOC_AREA_CODE</td>
<td>STR [0] &quot;064001000f&quot;</td>
</tr>
<tr>
<td>1 10000</td>
<td>SUBSTRUCT [0] allocated 20, used 2</td>
</tr>
<tr>
<td>2 10001</td>
<td>STR [0] &quot;Present&quot;</td>
</tr>
<tr>
<td>2 PIN_FLD_LOCATION</td>
<td>STR [0] &quot;004085752160&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_MSID</td>
<td>STR [0] &quot;004085752160&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_FLAGS</td>
<td>INT [0] 4</td>
</tr>
</tbody>
</table>
Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

0 PIN_FLD_POID POID [0] 0.0.0.1 /balance_group 127667 6441
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_Billinfo_OBJ POID [0] 0.0.0.1 /billinfo 129203 0
0 PIN_FLD_EFFECTIVE_T TSTAMP [0] (1337557354) Sun May 20 23:42:34 2012
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 10
1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1 PIN_FLD_CURRENT_BAL DECIMAL [0] 350
1 PIN_FLD_SUB_BALANCES ARRAY [0] allocated 20, used 12
2 PIN_FLD_CONTRIBUTOR_STR STR [0] ""
2 PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
2 PIN_FLD_VALID_TO_DETAILS INT [0] 0
2 PIN_FLD_VALID_FROM TSTAMP [0] (1337497200) Sun May 20 07:00:00 2012
2 PIN_FLD_VALID_FROM_DETAILS INT [0] 0
2 PIN_FLD_CURRENT_BAL DECIMAL [0] 350
2 PIN_FLD_NEXT_BAL DECIMAL [0] 0
2 PIN_FLD_DELAYED_BAL DECIMAL [0] 0
2 PIN_FLD_ROLLOVER_DATA INT [0] 0
2 PIN_FLD_GRANTOR_OBJ POID [0] 0.0.0.1 /purchased_product 130995 0
2 PIN_FLD_STATUS ENUM [0] 1
2 PIN_FLD_FLAGS INT [0] 2
1 PIN_FLD_CURRENT_TOTAL DECIMAL [0] 70
1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 500
1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_REALTIME_CNTR INT [0] 4
0 PIN_FLD_BALANCES ARRAY [1000011] allocated 20, used 10
1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1 PIN_FLD_CURRENT_BAL DECIMAL [0] 70
1 PIN_FLD_SUB_BALANCES ARRAY [4] allocated 20, used 12
2 PIN_FLD_CONTRIBUTOR_STR STR [0] ""
2 PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
2 PIN_FLD_VALID_TO_DETAILS INT [0] 0
2 PIN_FLD_VALID_FROM TSTAMP [0] (1337666360) Tue May 22 05:59:20 2012
2 PIN_FLD_VALID_FROM_DETAILS INT [0] 0
2 PIN_FLD_CURRENT_BAL DECIMAL [0] 70
2 PIN_FLD_NEXT_BAL DECIMAL [0] 0
2 PIN_FLD_DELAYED_BAL DECIMAL [0] 0
2 PIN_FLD_ROLLOVER_DATA INT [0] 0
2 PIN_FLD_GRANTOR_OBJ POID [0] 0.0.0.0 0 193
2 PIN_FLD_STATUS ENUM [0] 1
2 PIN_FLD_FLAGS INT [0] 2
1 PIN_FLD_CURRENT_TOTAL DECIMAL [0] -125
1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_BALANCES ARRAY [1000076] allocated 20, used 10
1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1 PIN_FLD_CURRENT_BAL DECIMAL [0] -125
1 PIN_FLD_SUB_BALANCES ARRAY [2] allocated 20, used 12
2 PIN_FLD_CONTRIBUTOR_STR STR [0] ""
2 PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
2 PIN_FLD_VALID_TO_DETAILS INT [0] 0
2 PIN_FLD_VALID_FROM TSTAMP [0] (1337497200) Sun May 20 07:00:00 2012
2 PIN_FLD_VALID_FROM_DETAILS INT [0] 0
2 PIN_FLD_CURRENT_BAL DECIMAL [0] -125
2 PIN_FLD_NEXT_BAL DECIMAL [0] 0
2 PIN_FLD_DELAYED_BAL DECIMAL [0] 0
2 PIN_FLD_ROLLOVER_DATA INT [0] 0
2 PIN_FLD_GRANTOR_OBJ POID [0] 0.0.0.1 /purchased_product 130995 193
2 PIN_FLD_STATUS ENUM [0] 1
2 PIN_FLD_FLAGS INT [0] 2
1 PIN_FLD_CURRENT_TOTAL DECIMAL [0] -125
Chapter 8

Usage Scenarios

1. PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
2. PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
3. PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
4. PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""

Operation: send is PCM_OP_PYMT_TOPUP (3726)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_INHERITED_INFO SUBSTRUCT [0] allocated 20, used 1
1 PIN_FLD_VOUCHERS_INFO ARRAY [0] allocated 20, used 4
2 PIN_FLD_DEVICE_ID STR [0] "9876543214"
2 PIN_FLD_VOUCHER_PIN STR [0] "3210"
2 PIN_FLD_BILLINFO_OBJ POID [0] 0.0.0.1 /billinfo 129203 0
2 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 127667 6441

Result: received for operation PCM_OP_PYMT_TOPUP (3726)

0 PIN_FLD_POID POID [0] 0.0.0.1 /error_poid 128819 0

Error specified:
location=5
pin_errclass=4
pin_err=3
field=83893688
rec_id=0
reserved=0
facility=0
msg_id=0
version=0
USSD recharge with invalid PIN

USSD recharge with invalid PIN flow
Here is an example message flow for a USSD recharge with invalid PIN.

USSD recharge with invalid PIN flow (BRM Vouchers)

### USSD recharge with invalid PIN scenario
This scenario describes the sequence of messages that occurs when a subscriber sends a USSD message requesting a voucher recharge but the subscriber's PIN is invalid.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | • MSC sends processUnstructuredSS-Request to USSD Gateway  
    | • Calling subscriber has sent a USSD message containing the service code for voucher recharge, the voucher number, and PIN. |
| 2    | • USSD Gateway sends InitialDP to the slee_acs process, using the USSD voucher recharge service key as determined by the service code configuration.  
    | • The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines the control plan to run, using the serviceKey to select the USSD voucher recharge control plan.  
    | • The slee_acs process runs the control plan.  
    | • The slee_acs process reaches a Voucher Recharge feature node. |
Step 3
- The Voucher Recharge feature node sends ConnectToResource, PromptAndCollectUserInformation to USSD Gateway, instructing it to prompt the caller for the voucher number and PIN.
- USSD Gateway retrieves the voucher number and PIN from the received processUnstructuredSS-Request, without further interaction with MSC or the subscriber.

Step 4
- MSC sends a PromptAndCollectUserInformation result, containing the voucher number and PIN, to the slee_acs process.
- The Voucher Recharge feature node invokes the VoucherRedeem action on the BCD actions library.

Step 5
- The BCD actions library constructs an event (FlistSleeEvent) that contains a PCM_OP_PYMT_TOPUP operation and sends it to the BCD Client.
- The BCD Client invokes the PCM_OP_PYMT_TOPUP operation and sets a timer to the configured value for this operation.

Step 6
- BRM responds to the operation, indicating that the PIN was invalid.
- The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.
- The BCD actions library translates the output flist into the error response of the VoucherRedeem action.

Step 7
- The Voucher Recharge feature node sends PlayAnnouncement to USSD Gateway.
- USSD Gateway translates the announcement ID into text stating that the recharge was unsuccessful, based on configuration.

Step 8
- USSD Gateway sends processUnstructuredSS-Request result to the MSC.
- USSD text stating that the recharge was unsuccessful is displayed on the caller's handset.

Step 9
- USSD Gateway sends SpecializedResourceReport to the slee_acs process.

Step 10
- The control plan reaches an end node and ACS sends DisconnectForwardConnection and ReleaseCall actions to USSD Gateway.

Messages: USSD recharge with invalid PIN
The following messages include operations sent to BRM and results returned by BRM for a recharge with an invalid PIN using USSD. The general message format is: nesting level (0; 1, or 2); field; data type; value.

Operation: send is PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

Flags - 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2 PIN_FLD_DIRECTION ENUM [0] 0
2 PIN_FLD_CELL_ID STR [0] "0000"
2 PIN_FLD_LOC_AREA_CODE STR [0] "064010000f"
1 10000 SUBSTRUCT [0] allocated 20, used 2
2 10001 STR [0] "Present"
2 PIN_FLD_LOCATION STR [0] "04068572160"
0 PIN_FLD_MSID STR [0] "04068572160"
0 PIN_FLD_FLAGS INT [0] 4

Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)
0 PIN_FLD_POID POID [0] 0.0.0.1 /balance_group 127667 6441
Chapter 8

```
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_BILLING_OBJ POID [0] 0.0.0.1 /billinfo 129203 0
0 PIN_FLD_EFFECTIVE_T TSTAMP [0] (1337557354) Sun May 20 23:42:34 2012
0 PIN_FLD_BALANCES ARRAY (978) allocated 20, used 10
1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1 PIN_FLD_CURRENT_BAL DECIMAL [0] 350
1 PIN_FLD_SUB_BALANCES ARRAY [0] allocated 20, used 12
2 PIN_FLD_CONTRIBUTOR_STR STRING [0] ""'
2 PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
2 PIN_FLD_VALID_TO_DETAILS INT [0] 0
2 PIN_FLD_VALID_FROM TSTAMP [0] (1337497200) Sun May 20 07:00:00 2012
2 PIN_FLD_VALID_FROM_DETAILS INT [0] 0
2 PIN_FLD_CURRENT_BAL DECIMAL [0] 350
2 PIN_FLD_NEXT_BAL DECIMAL [0] 0
2 PIN_FLD_DELAYED_BAL DECIMAL [0] 0
2 PIN_FLD_ROLLOVER_DATA INT [0] 0
2 PIN_FLD_GRANTOR_OBJ POID [0] 0.0.0.1 /purchased_product 130995 0
2 PIN_FLD_STATUS ENUM [0] 1
2 PIN_FLD_FLAGS INT [0] 2
1 PIN_FLD_CURRENT_TOTAL DECIMAL [0] 350
1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 500
1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STRING [0] ""
0 PIN_FLD_BALANCES ARRAY [1000011] allocated 20, used 10
1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1 PIN_FLD_CURRENT_BAL DECIMAL [0] 70
1 PIN_FLD_SUB_BALANCES ARRAY [4] allocated 20, used 12
2 PIN_FLD_CONTRIBUTOR_STR STRING [0] ""'
2 PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
2 PIN_FLD_VALID_TO_DETAILS INT [0] 0
2 PIN_FLD_VALID_FROM TSTAMP [0] (1337666360) Tue May 22 05:59:20 2012
2 PIN_FLD_VALID_FROM_DETAILS INT [0] 0
2 PIN_FLD_CURRENT_BAL DECIMAL [0] 70
2 PIN_FLD_NEXT_BAL DECIMAL [0] 0
2 PIN_FLD_DELAYED_BAL DECIMAL [0] 0
2 PIN_FLD_ROLLOVER_DATA INT [0] 0
2 PIN_FLD_GRANTOR_OBJ POID [0] 0.0.0.0 /purchased_product 130995 193
2 PIN_FLD_STATUS ENUM [0] 1
2 PIN_FLD_FLAGS INT [0] 2
1 PIN_FLD_CURRENT_TOTAL DECIMAL [0] 70
1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
```

158 NCC BRM Charging Driver Technical Guide
PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""

**Operation:** send is PCM_OP_PYMT_TOPUP (3726)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_INHERITED_INFO SUBSTRUCT [0] allocated 20, used 1
1 PIN_FLD_VOUCHERS_INFO ARRAY [0] allocated 20, used 4
2   PIN_FLD_DEVICE_ID STR [0] "9876543214"
2   PIN_FLD_VOUCHER_PIN STR [0] "3210"
2   PIN_FLD_BILLINFO_OBJ POID [0] 0.0.0.1 /billinfo 129203 0
2   PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 127667 6441

**Result:** received for operation PCM_OP_PYMT_TOPUP (3726)

0 PIN_FLD_POID POID [0] 0.0.0.1 /error_poid 128819 0

Error specified:
  location=5
  pin_errclass=4
  pin_err=3
  field=83893688
  rec_id=0
  reserved=0
  facility=0
  msg_id=0
  version=0
Successful SMS recharge using a VWS voucher

Successful SMS recharge using a VWS voucher flow
Here is an example message flow for a scenario of a successful SMS recharge using a VWS voucher.

Successful SMS voucher recharge (VWS Vouchers)

<table>
<thead>
<tr>
<th>MSC</th>
<th>XMS Trigger</th>
<th>slee_abc/BCD Client</th>
<th>VWS</th>
<th>BRM Connection Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Successful SMS recharge using a VWS voucher scenario
This scenario describes the sequence of messages that occur during a successful SMS recharge using a VWS voucher. In this scenario, the account is on BRM while the voucher is in the NCC VWS.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MSC sends MOForwardSM to xmsTrigger.</td>
</tr>
<tr>
<td></td>
<td>The subscriber has sent an SMS, which contains the voucher and PIN, from a GSM phone to the SMS recharge number.</td>
</tr>
</tbody>
</table>
Chapter 8

### Usage Scenarios

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 2    | xmsTrigger sends InitialDP to the slee_acs process, with the service key set to a special value indicating SMS recharge.  
- The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.  
- The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is in the BRM domain. It also determines the control plan to run, based on the serviceKey.  
- The slee_acs process runs the SMS recharge control plan.  
- The slee_acs process reaches an Extract Content feature node, which extracts the voucher number and PIN from the SMS and places them in temporary storage.  
- The slee_acs process reaches a Voucher Recharge feature node, which extracts the voucher number and PIN from temporary storage and uses them to invoke a VoucherRedeem action on the FOX actions library.
| 3    | FOX actions library sends VR_Req to BeClient.  
- BeClient send VR_Req to WVS.
| 4    | VWS sends VR_Ack to BeClient, indicating that the voucher has been reserved and returning the recharge amounts for each balance type.  
- BeClient send VR_Ack to the slee_acs process.  
- The Voucher Recharge feature node invokes a WalletRecharge action on the BCD actions library.
| 5    | The slee_acs process creates an event (FlistSleeEvent) for a PCM_OP_BILL_DEBIT operation and sends it to the BCD Client  
- The BCD Client invokes the PCM_OP_BILL_DEBIT operation and sets a timer to the configured value for this type of operation.
| 6    | BRM responds to the operation indicating that the account has been successfully recharged.  
- The BCD Client receives the operation output flist and sends it in BcdSleeEvent to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
- The BCD actions library translates the output flist into the response of the WalletRecharge action.  
- The Voucher Recharge feature node invokes the VoucherConfirm action on FOX actions library.
| 7    | FOX actions library sends CVR_Req to BeClient.  
- BeClient sends CVR_Req to VWS.  
- VWS sends CVR_Ack to BeClient, indicating that the voucher has been permanently marked as redeemed.
| 8    | BeClient sends CVR_Ack to the slee_acs process.  
- The slee_acs process takes the success branch of the VoucherRecharge feature node and reaches a Send Short Message Notification feature node, which specifies sending a success text message.  
- The SSMN feature node creates an MMX GenericMessage containing the success message and sends it to xmsTrigger.
| 9    | xmsTrigger sends an MTForwardSM operation to the MSC, containing the success message.  
- MSC sends an SMS containing the success message to the caller.  
- The slee_acs process reaches an Accept feature node which sends ReleaseCall to xmsTrigger.
Chapter 8

Step                        Action

13. xmsTrigger sends an MOForwardSM result to the MSC.
    - The MSC sends a notification of successful delivery of the original SMS to the caller
    - The slee_acs process reaches an end node and clears the call connection.

Messages: successful SMS recharge using a VWS voucher
The following messages include operations sent to BRM and results returned by BRM for a VWS voucher recharge to a BRM account using SMS. The general message format is: nesting level (0; 1, or 2); field; data type; value.

Operation: send is PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
  1   PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
  2   PIN_FLD_DIRECTION ENUM [0] 0
  2   PIN_FLD_CELL_ID STR [0] "000c"
  2   PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
  1  10000 SUBSTRUCT [0] allocated 20, used 2
  2   10001 STR [0] "Present"
  2   PIN_FLD_LOCATION STR [0] "004085752160"
0 PIN_FLD_MSID STR [0] "004085752160"

Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

0 PIN_FLD_POID POID [0] 0.0.0.1 /balance_group 127667 6446
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_BILLINFO_OBJ POID [0] 0.0.0.1 /billinfo 129203 0
0 PIN_FLD_EFFECTIVE_T TSTAMP [0] (1337557354) Sun May 20 23:42:34 2012
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 8
  1  PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  1  PIN_FLD_CURRENT_BAL DECIMAL [0] 400
  1  PIN_FLD_CURRENT_BAL DECIMAL [0] 0
  1  PIN_FLD_CURRENT_BAL DECIMAL [0] 0
  1  PIN_FLD_CURRENT_BAL DECIMAL [0] 70
  1  PIN_FLD_CURRENT_BAL DECIMAL [0] 0
  1  PIN_FLD_CURRENT_BAL DECIMAL [0] 0
  1  PIN_FLD_CURRENT_BAL DECIMAL [0] 0
  1  PIN_FLD_CURRENT_BAL DECIMAL [0] 0
  1  PIN_FLD_CURRENT_BAL DECIMAL [0] 0
  1  PIN_FLD_CURRENT_BAL DECIMAL [0] 0

Operation: send is PCM_OP_BILL_DEBIT (105)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_MSID ARRAY [978] allocated 20, used 1
Result: received for operation PCM_OP_BILL_DEBIT (105)

0 PIN_FLD_POID POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_RESULTS ARRAY [0] allocated 20, used 4
1 PIN_FLD_BAL_IMPACTS ARRAY [0] allocated 20, used 17
2 PIN_FLD_IMPACT_TYPE ENUM [0] 2
2 PIN_FLD_RESOURCE_ID INT [0] 978
2 PIN_FLD_RESOURCE_ID_ORIG INT [0] 0
2 PIN_FLD_TAX_CODE STR [0] ""
2 PIN_FLD_RATE_TAG STR [0] ""
2 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 128819 0
2 PIN_FLD_RATE_OBJ POID [0] 0.0.0.0 0 0
2 PIN_FLD_DISCOUNT DECIMAL [0] 0
2 PIN_FLD_PERCENT DECIMAL [0] 0
2 PIN_FLD_QUANTITY DECIMAL [0] 0
2 PIN_FLD_AMOUNT_DEFERRED DECIMAL [0] 0
2 PIN_FLD_AMOUNT DECIMAL [0] -0.500
2 PIN_FLD_AMOUNT_ORIG DECIMAL [0] NULL pin_decimal_t ptr
2 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 127667 0
2 PIN_FLD_GL_ID INT [0] 0
2 PIN_FLD_ITEM_OBJ POID [0] 0.0.0.1 /item/misc 129715 1466
2 PIN_FLD_LINEAGE STR [0] NULL str ptr
1 PIN_FLD_SUB_BAL_IMPACTS ARRAY [0] allocated 20, used 3
2 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 127667 6447
2 PIN_FLD_RESOURCE_ID INT [0] 978
2 PIN_FLD_SUB_BALANCES ARRAY [0] allocated 20, used 3
3 PIN_FLD_AMOUNT DECIMAL [0] -0.500
3 PIN_FLD_VALID_FROM TSTAMP [0] (0) <null>
3 PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
1 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 128819 0
1 PIN_FLD_POID POID [0] 0.0.0.1 /event/billing/debit 272643699316523851 0
Successful USSD recharge using a VWS voucher

Successful USSD recharge using a VWS voucher flow
Here is an example message flow for a scenario of a successful USSD recharge using a VWS voucher.

Successful USSD recharge (VWS Vouchers)

Successful USSD recharge using a VWS voucher scenario
This scenario describes the sequence of messages that occurs for successful USSD recharge using a VWS voucher. In this scenario, the account is on BRM while the voucher is in the NCC VWS.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | • MSC sends processUnstructuredSS-Request to USSD Gateway  
      • The calling subscriber has sent a USSD message that contains the service code for a voucher recharge, the voucher number and PIN. |
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 2 | • The USSD Gateway sends InitialDP to the slee_acs process, using the USSD voucher recharge service key as determined by the service code configuration.  
  • The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. It also determines the control plan to run, using the serviceKey to select the USSD voucher recharge control plan.  
  • The slee_acs process runs the control plan.  
  • The slee_acs process reaches a Voucher Recharge feature node. |
| 3 | • The Voucher Recharge feature node sends ConnectToResource, PromptAndCollectUserInformation to USSD Gateway, instructing it to prompt the caller for the voucher number and PIN.  
  • USSD Gateway retrieves the voucher number and PIN from the received processUnstructuredSS-Request. |
| 4 | • USSD Gateway sends the PromptAndCollectUserInformation result, containing the voucher number and PIN, to the slee_acs process.  
  • The Voucher Recharge feature node invokes the VoucherRedeem action on the FOX actions library. |
| 5 | • The FOX actions library sends VR_Req to BeClient  
  • BeClient sends VR_Req to VWS |
| 6 | • VWS sends VR_Ack to BeClient, indicating that the voucher has been reserved, and returning the recharge amounts for each balance type.  
  • BeClient sends VR_Ack to the slee_acs process.  
  • The Voucher Recharge feature node invokes the WalletRecharge action on the BCD actions library. |
| 7 | • The slee_acs process creates an FlistSleeEvent for a PCM_OP_BILL_DEBIT operation and sends it to the BCD Client.  
  • The BCD actions library creates an event (FlistSleeEvent) that contains a PCM_OP_BILL_DEBIT operation and sends it to the BCD Client  
  • The BCD Client invokes the PCM_OP_BILL_DEBIT operation and sets a timer to the configured value for this type of operation. |
| 8 | • BRM responds to the operation indicating that the account has been successfully recharged.  
  • The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
  • The BCD actions library translates the output flist into the response of the WalletRecharge action.  
  • The Voucher Recharge feature node invokes the Voucher Confirm action on the FOX actions library. |
| 9 | • The FOX actions library sends CVR_Req to BeClient  
  • BeClient sends CVR_Req to VWS. |
| 10 | • VWS sends CVR_Ack to BeClient, indicating that the voucher has been permanently marked as redeemed.  
  • BeClient sends CVR_Ack to the slee_acs process. |
| 11 | • The Voucher Recharge feature node sends PlayAnnouncement to USSD Gateway.  
  • USSD Gateway translates the announcement into text that states that the recharge was successful, based on configuration. |
Chapter 8

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 12   | • USSD Gateway sends processUnstructuredSS-Request result to the MSC.  
      • USSD text, which states that the recharge was successful, is displayed on the caller’s handset. |
| 13   | • USSD Gateway sends SpecializedResourceReport to the slee_acs process. |
| 14   | • The control plan reaches an end node and ACS sends DisconnectForwardConnection and ReleaseCall actions to USSD Gateway. |

Messages: successful USSD recharge using a VWS voucher
The following messages include operations sent to BRM and results returned by BRM for a VWS voucher recharge to a BRM account using USSD. The general message format is: nesting level (0; 1, or 2); field; data type; value.

Operation: send is PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

<table>
<thead>
<tr>
<th>Flags</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 PIN_FLD_POID</td>
<td>POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0</td>
</tr>
<tr>
<td>0 PIN_FLD_PROGRAM_NAME</td>
<td>STR [0] &quot;NCC_BCD_Client&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_OBJ_TYPE</td>
<td>STR [0] &quot;gsm/ncc&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2</td>
<td></td>
</tr>
<tr>
<td>1 PIN_FLD_GSM_INFO</td>
<td>SUBSTRUCT [0] allocated 20, used 3</td>
</tr>
<tr>
<td>2 PIN_FLD_DIRECTION</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>2 PIN_FLD_CELL_ID</td>
<td>STR [0] &quot;000c&quot;</td>
</tr>
<tr>
<td>2 PIN_FLD_LOC_AREA_CODE</td>
<td>STR [0] &quot;064001000f&quot;</td>
</tr>
<tr>
<td>1 10000</td>
<td>SUBSTRUCT [0] allocated 20, used 2</td>
</tr>
<tr>
<td>2 10001</td>
<td>STR [0] &quot;Present&quot;</td>
</tr>
<tr>
<td>2 PIN_FLD_LOCATION</td>
<td>STR [0] &quot;004085752160&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_MSID</td>
<td>STR [0] &quot;004085752160&quot;</td>
</tr>
</tbody>
</table>

Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

<table>
<thead>
<tr>
<th>Flags</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 PIN_FLD_POID</td>
<td>POID [0] 0.0.0.1 /balance_group 127667 6446</td>
</tr>
<tr>
<td>0 PIN_FLD_ACCOUNT_OBJ</td>
<td>POID [0] 0.0.0.1 /account 128819 0</td>
</tr>
<tr>
<td>0 PIN_FLD_BILLINFO_OBJ</td>
<td>POID [0] 0.0.0.1 /billinfo 129203 0</td>
</tr>
<tr>
<td>0 PIN_FLD_EFFECTIVE_T</td>
<td>TSTAMP [0] (1337557354) Sun May 20 23:42:34 2012</td>
</tr>
<tr>
<td>0 PIN_FLD_BALANCES</td>
<td>ARRAY [978] allocated 20, used 8</td>
</tr>
<tr>
<td>1 PIN_FLD_RESERVED_AMOUNT</td>
<td>DECIMAL [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_NEXT_BAL</td>
<td>DECIMAL [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_CONSUMPTION_RULE</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_CURRENT_BAL</td>
<td>DECIMAL [0] 400</td>
</tr>
<tr>
<td>1 PIN_FLD_CREDIT_FLOOR</td>
<td>DECIMAL [0] NULL</td>
</tr>
<tr>
<td>1 PIN_FLD_CREDIT_LIMIT</td>
<td>DECIMAL [0] 500</td>
</tr>
<tr>
<td>1 PIN_FLD_CREDIT_THRESHOLDS</td>
<td>INT [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_CREDIT_THRESHOLDS_FIXED</td>
<td>STR [0] &quot;&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_REALTIME_CNTR</td>
<td>INT [0] 4</td>
</tr>
<tr>
<td>0 PIN_FLD_BALANCES</td>
<td>ARRAY [1000011] allocated 20, used 8</td>
</tr>
<tr>
<td>1 PIN_FLD_RESERVED_AMOUNT</td>
<td>DECIMAL [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_NEXT_BAL</td>
<td>DECIMAL [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_CONSUMPTION_RULE</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_CURRENT_BAL</td>
<td>DECIMAL [0] 70</td>
</tr>
<tr>
<td>1 PIN_FLD_CREDIT_FLOOR</td>
<td>DECIMAL [0] NULL</td>
</tr>
<tr>
<td>1 PIN_FLD_CREDIT_LIMIT</td>
<td>DECIMAL [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_CREDIT_THRESHOLDS</td>
<td>INT [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_CREDIT_THRESHOLDS_FIXED</td>
<td>STR [0] &quot;&quot;</td>
</tr>
<tr>
<td>0 PIN_FLD_BALANCES</td>
<td>ARRAY [1000076] allocated 20, used 8</td>
</tr>
<tr>
<td>1 PIN_FLD_RESERVED_AMOUNT</td>
<td>DECIMAL [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_NEXT_BAL</td>
<td>DECIMAL [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_CONSUMPTION_RULE</td>
<td>ENUM [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_CURRENT_BAL</td>
<td>DECIMAL [0] -125</td>
</tr>
<tr>
<td>1 PIN_FLD_CREDIT_FLOOR</td>
<td>DECIMAL [0] NULL</td>
</tr>
<tr>
<td>1 PIN_FLD_CREDIT_LIMIT</td>
<td>DECIMAL [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_CREDIT_THRESHOLDS</td>
<td>INT [0] 0</td>
</tr>
<tr>
<td>1 PIN_FLD_CREDIT_THRESHOLDS_FIXED</td>
<td>STR [0] &quot;&quot;</td>
</tr>
</tbody>
</table>

Operation: send is PCM_OP_BILL_DEBIT (105)

| Flags | 0 |
Chapter 8

Result: received for operation PCM_OP_BILL_DEBIT (105)

```
0  PIN_FLD_POID          POID [0] 0.0.0.1 /account 128819 0
0  PIN_FLD_PROGRAM_NAME  STR [0] "NCC_BCD_Client"
0  PIN_FLD_OBJ_TYPE      STR [0] "gsm/ncc"
1  PIN_FLD_DEBIT         ARRAY [978] allocated 20, used 1
    PIN_FLD_BAL_OPERAND  DECIMAL [0] -0.500000000000000

Result: received for operation PCM_OP_BILL_DEBIT (105)

0  PIN_FLD_POID          POID [0] 0.0.0.1 /account 128819 0
0  PIN_FLD_RESULTS       ARRAY [0] allocated 20, used 4
1   PIN_FLD_BAL_IMPACTS  ARRAY [0] allocated 20, used 17
2    PIN_FLD_IMPACT_TYPE  ENUM [0] 2
2    PIN_FLD_RESOURCE_ID  INT [0] 978
2    PIN_FLD_TAX_CODE     STR [0] ""
2    PIN_FLD_RATE_TAG      STR [0] ""
2    PIN_FLD_ACCOUNT_OBJ   POID [0] 0.0.0.1 /account 128819 0
2    PIN_FLD_RATE_OBJ      POID [0] 0.0.0.0 0 0
2    PIN_FLD_DISCOUNT      DECIMAL [0] 0
2    PIN_FLD_PERCENT       DECIMAL [0] 0
2    PIN_FLD_QUANTITY      DECIMAL [0] 0
2    PIN_FLD_AMOUNT_DEFERRED DECIMAL [0] 0
2    PIN_FLD_AMOUNT_ORIG   DECIMAL [0] NULL pin_decimal_t ptr
2    PIN_FLD_BAL_GRP_OBJ   POID [0] 0.0.0.1 /balance_group 127667 0
2    PIN_FLD_GL_ID         INT [0] 0
2    PIN_FLD_ITEM_OBJ      POID [0] 0.0.0.1 /item/misc 129715 1466
2    PIN_FLD_LINEAGE       STR [0] NULL str ptr
1   PIN_FLD_SUB_BAL_IMPACTS ARRAY [0] allocated 20, used 3
2    PIN_FLD_BAL_GRP_OBJ   POID [0] 0.0.0.1 /balance_group 127667 6447
2    PIN_FLDRESOURCE_ID    INT [0] 978
2   PIN_FLD_SUB_BALANCES  ARRAY [0] allocated 20, used 3
3    PIN_FLD_AMOUNT       DECIMAL [0] -0.500
3    PIN_FLD_VALID_FROM   TSTAMP [0] (0) <null>
3    PIN_FLD_VALID_TO     TSTAMP [0] (0) <null>
1   PIN_FLD_ACCOUNT_OBJ   POID [0] 0.0.0.1 /account 128819 0
1   PIN_FLD_POID          POID [0] 0.0.0.1 /event/billing/debit 272643699316523851 0
```
Error on IVR Recharge using a VWS voucher with invalid PIN

Here is an example message flow for an error on an IVR Recharge using a VWS voucher with an invalid PIN.

**Error on IVR voucher recharge with invalid PIN flow (VWS Vouchers)**

1. **InitialDP**
   - MSC
   - sles_acs/BCD Client
   - VWS
   - BRM Connection Manager
   - The subscriber calls the toll free number for a voucher recharge
   - MSC sends InitialDP to the sles_acs process with serviceKey set to a special value indicating voucher recharge.
   - xmsTrigger sends InitialDP to the sles_acs process.

2. **collectedDigits**
   - The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain and VWS is used for the voucher. It also uses the serviceKey to select the voucher recharge control plan.
   - The sles_acs process runs the control plan.
   - The sles_acs process reaches a Voucher Recharge feature node.
   - The Voucher Recharge feature node sends ConnectToResource, PromptAndCollectUserInformation to the MSC, instructing the MSC to prompt the caller for the voucher number and PIN.

3. **VR_Req**
   - VR_Req
   - VR_NAck
   - PCM_OP_ACT_ACTIVITY Bad PIN Event
   - PCM_OP_ACT_ACTIVITY response
   - PlayAnnouncement
   - DisconnectedForwardConnection
   - ReleaseCall

Error on IVR recharge using a VWS voucher with invalid PIN scenario

This scenario describes the sequence of messages that occurs for an error when the subscriber calls the toll free number to request a voucher recharge but the subscriber's PIN is invalid. In this scenario, the account is on BRM while the voucher is in the NCC VWS.

**Step** | **Action**
--- | ---
1 | • The subscriber calls the toll free number for a voucher recharge  
   • MSC sends InitialDP to the sles_acs process with serviceKey set to a special value indicating voucher recharge.  
   • xmsTrigger sends InitialDP to the sles_acs process.
2 | • The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain and VWS is used for the voucher. It also uses the serviceKey to select the voucher recharge control plan.  
   • The sles_acs process runs the control plan.  
   • The sles_acs process reaches a Voucher Recharge feature node.  
   • The Voucher Recharge feature node sends ConnectToResource, PromptAndCollectUserInformation to the MSC, instructing the MSC to prompt the caller for the voucher number and PIN.
### Step 3
- MSC plays the specified announcement to the caller and collects the voucher number and PIN.
- MSC sends PromptAndCollectUserInformation result, containing voucher number and PIN to the slee_acs process.

### Step 4
- The Voucher Recharge feature node invokes the VoucherRedeem action on the FOX actions library.
- FOX actions library sends VR_Req to BeClient.
- BeClient sends VR_Req to VWS.

### Step 5
- VWS sends VR_Nack to BeClient, indicating that the voucher PIN is incorrect.
- The Voucher Recharge feature node invokes the VoucherBadPIN action on the BCD actions library.
- The VoucherBadPIN action does nothing and returns success.
- The Voucher Recharge feature node invokes the VoucherCreateEDR action on the BCD actions library to create a record for the bad PIN attempt.

### Step 6
- The VoucherCreateEDR action creates an event (FlistSleeEvent) that contains a PCM_OP_ACT_ACTIVITY operation for the special voucher bad PIN event and sends it to BCD Client.
- The BCD Client invokes the PCM_OP_ACT_ACTIVITY operation and sets a timer to the configured value for this type of operation.

### Step 7
- BRM creates a record for the bad PIN event and responds to the operation by indicating that this has been done.
- The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.
- The BCD actions library translates the output flist into the response of the VoucherCreateEDR action.

**Note:** Recording bad PIN attempts would allow BRM to freeze accounts against which a large number of bad PIN attempts have been made. Some customization would be required, however, for BRM to freeze an account.

### Step 8
- The Voucher Recharge feature node sends PlayAnnouncement to the MSC.
- MSC plays an announcement to the caller, stating that the recharge was unsuccessful.

### Step 9
- The announcement finishes and MSC sends SpecializedResourceReport to the slee_acs process.

### Step 10
- The control plan reaches an end node and ACS sends DisconnectForwardConnection and ReleaseCall actions to the MSC and clears the call context.
- The caller is disconnected.

**Messages:** error on IVR recharge using a VWS voucher with an invalid PIN
The following messages include operations sent to BRM and results returned by BRM for an IVR recharge using a VWS voucher with an invalid PIN. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation:** send is PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

```plaintext
Flags = 0
0 PIN_FLD_POID      POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE   STR [0] "gsm/ncc"
```
Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

0 PIN_FLD_POID POID [0] 0.0.0.1/balance_group 127667 6530
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1/account 128819 0
0 PIN_FLD_BILLINFO_OBJ POID [0] 0.0.0.1/billinfo 129203 0
0 PIN_FLD_EFFECTIVE_T TSTAMP [0] (1337557354) Sun May 20 23:42:34 2012
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 8
1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1 PIN_FLD_CURRENT_BAL DECIMAL [0] 491.13
1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 500
1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_REALTIME_CNTR INT [0] 4
0 PIN_FLD_BALANCES ARRAY [1000011] allocated 20, used 8
1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1 PIN_FLD_CURRENT_BAL DECIMAL [0] 70
1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_BALANCES ARRAY [1000076] allocated 20, used 8
1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1 PIN_FLD_CURRENT_BAL DECIMAL [0] -87.5
1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""

Operation: send is PCM_OP_ACT_ACTIVITY (151)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1/account 128819 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "/voucher"
0 PIN_FLD_INHERITED_INFO SUBSTRUCT [0] allocated 20, used 1
1 10000 SUBSTRUCT [0] allocated 20, used 2
2 10007 STR [0] "9876543214"
2 10008 STR [0] "3210"
0 PIN_FLD_BAL_IMPACTS ARRAY [1000011] allocated 20, used 8
1 PIN_FLD_AMOUNT DECIMAL [0] 1
1 PIN_FLD_RESOURCE_ID INT [0] 1000011
1 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1/account 128819 0
1 PIN_FLD_IMPACT_TYPE ENUM [0] 2

Result: received for operation PCM_OP_ACT_ACTIVITY (151)

0 PIN_FLD_POID POID [0] 0.0.0.1/account 128819 0
0 PIN_FLD_RESULTS ARRAY [0] allocated 20, used 4
1 PIN_FLD_BAL_IMPACTS ARRAY [1000011] allocated 20, used 8
2 PIN_FLD_AMOUNT DECIMAL [0] 1
2 PIN_FLD_RESOURCE_ID INT [0] 1000011
2 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1/account 128819 0
2 PIN_FLD_IMPACT_TYPE ENUM [0] 2
2 PIN_FLD_GL_ID INT [0] 0
2 PIN_FLD_ITEM_OBJ POID [0] 0.0.0.1/item/misc 175577 0
Error on IVR recharge account topup failed

Here is an example message flow for an IVR Recharge in which an account topup failed.

![Diagram of message flow](image-url)
Error on IVR recharge account topup failed scenario
This scenario describes the sequence of messages that occurs when a subscriber calls a toll-free number to request a voucher recharge but the action to recharge the subscriber's wallet on BRM fails. In this scenario, the account is on BRM while the voucher is in the NCC VWS.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | - The subscriber calls the toll free number for a voucher recharge.  
      - MSC sends InitialDP to the slee_acs process with serviceKey set to a special value indicating voucher recharge. |
| 2    | - The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain and VWS is used for the voucher. It also uses the serviceKey to select the voucher recharge control plan.  
      - The slee_acs process runs the control plan.  
      - The slee_acs process reaches a Voucher Recharge feature node.  
      - The Voucher Recharge feature node sends ConnectToResource, PromptAndCollectUserInformation to the MSC, instructing the MSC to prompt the caller for the voucher number and PIN. |
| 3    | - MSC plays the specified announcement to the caller and collects the voucher number and PIN.  
      - MSC sends PromptAndCollectUserInformation result, containing voucher number and PIN to the slee_acs process. |
| 4    | - The Voucher Recharge feature node invokes the VoucherRedeem action on the FOX actions library.  
      - FOX actions library sends VR_Req to BeClient.  
      - BeClient sends VR_Req to VWS. |
| 5    | - VWS sends VR_Ack to BeClient, indicating that the voucher has been reserved, and returning the recharge amounts for each balance type.  
      - BeClient sends VR_Ack to the slee_acs process. |
| 6    | - The Voucher Recharge feature node invokes the WalletRecharge action on the BCD actions library.  
      - The BCD actions library constructs an event (BcdSleeEvent) containing PCM_OP_BILL_DEBIT and sends it to the BCD Client.  
      - The BCD Client invokes the PCM_OP_BILL_DEBIT operation and sets a timer to the configured value for this type of operation. |
| 7    | - BRM responds to the operation by indicating that the account has not been recharged.  
      - The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
      - The BCD actions library translates the output flist into the error response of the WalletRecharge action.  
      - The Voucher Recharge feature node invokes the VoucherRevoke action on the FOX actions library. |
| 8    | - The FOX actions library sends RVR_Req to BeClient.  
      - BeClient sends RVR_Req to VWS. |
| 9    | - VWS sends RVR_Ack to BeClient, indicating that the voucher reservation has been revoked.  
      - BeClient sends CVR_Ack to the slee_acs process. |
### Step 10
- The Voucher Recharge feature node sends PlayAnnouncement to the MSC.
- MSC plays an announcement to the caller, stating that the recharge was not successful.

### Step 11
- The announcement finishes and MSC sends SpecializedResourceReport to the :)acs process.

### Step 12
- The control plan reaches an end node and ACS sends DisconnectForwardConnection and ReleaseCall actions to the MSC and clears the call connection.
- The caller is disconnected.

**Messages: Error on IVR recharge account topup failed**
The following messages include operations sent to BRM and results returned by BRM for an IVR recharge when the account topup failed. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation:** send is PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

**Flags = 0**
- 0 PIN_FLD_POID       POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
- 0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
- 0 PIN_FLD_OBJ_TYPE    STR [0] "gsm/ncc"
- 0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
  - 1 PIN_FLD_GSM_INFO   SUBSTRUCT [0] allocated 20, used 3
    - 2 PIN_FLD_DIRECTION ENUM [0] 0
    - 2 PIN_FLD_CELL_ID   STR [0] "000c"
  - 2 PIN_FLD_Location   STR [0] "004085752160"
- 0 PIN_FLD_MSID       STR [0] "004085752160"

**Result:** received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

- 0 PIN_FLD_POID       POID [0] 0.0.0.1 /balance_group 127667 6446
- 0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 128819 0
- 0 PIN_FLD_BILLINFO_OBJ POID [0] 0.0.0.1 /billinfo 129203 0
- 0 PIN_FLD_EFFECTIVE_T TSTAMP [0] (1337557354) Sun May 20 23:42:34 2012
- 0 PIN_FLD_BALANCES    ARRAY [978] allocated 20, used 8
  - 1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  - 1 PIN_FLD_NEXT_BAL     DECIMAL [0] 0
  - 1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
  - 1 PIN_FLD_CURRENT_BAL  DECIMAL [0] 400
  - 1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
  - 1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 500
  - 1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
  - 1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
- 0 PIN_FLD_REALTIME_CNTR INT [0] 4
- 0 PIN_FLD_BALANCES    ARRAY [1000011] allocated 20, used 8
  - 1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  - 1 PIN_FLD_NEXT_BAL     DECIMAL [0] 0
  - 1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
  - 1 PIN_FLD_CURRENT_BAL  DECIMAL [0] 70
  - 1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
  - 1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
  - 1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
  - 1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
- 0 PIN_FLD_BALANCES    ARRAY [1000076] allocated 20, used 8
  - 1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  - 1 PIN_FLD_NEXT_BAL     DECIMAL [0] 0
  - 1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
  - 1 PIN_FLD_CURRENT_BAL  DECIMAL [0] -125
  - 1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
  - 1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
Chapter 8

Operation: send is PCM_OP_BILL_DEBIT (105)

Result: received for operation PCM_OP_BILL_DEBIT(105)

Error specified:

Error on SMS recharge with invalid PIN

Here is an example message flow for an error on an SMS recharge with an invalid PIN.
### Error on SMS recharge with invalid PIN scenario

This scenario describes the sequence of messages that occurs when a subscriber's attempt to recharge a voucher fails due to an invalid PIN and MSC sends a short text message to inform the caller. In this scenario, the account is on BRM while the voucher is in the NCC VWS.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1. | MSC sends MOForwardSM to xmsTrigger.  
| | • The subscriber has sent an SMS, which contains the voucher and PIN, from a GSM phone to the SMS recharge number. |
| 2. | xmsTrigger sends InitialDP to the slee_acs process, with the service key set to a special value indicating SMS recharge.  
| | • The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.  
| | • The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is in the BRM domain. It also determines the control plan to run, based on the serviceKey.  
| | • The slee_acs process runs the SMS recharge control plan.  
| | • The slee_acs process reaches an Extract Content feature node, which extracts the voucher number and PIN from the SMS and places them in temporary storage.  
| | • The slee_acs process reaches a Voucher Recharge feature node, which extracts the voucher number and PIN from temporary storage and uses them to invoke a VoucherRedeem action on the FOX actions library. |
| 3. | FOX actions library sends VR_Req to BeClient.  
| | • BeClient sends VR_Req to VWS. |
| 4. | VWS sends VR_Nack to BeClient, indicating that the voucher PIN is incorrect.  
| | • The Voucher Recharge feature node invokes a VoucherBadPIN action on the BCD actions library.  
| | • The VoucherBadPIN action does nothing and returns success.  
| | • The Voucher Recharge feature node invokes the VoucherCreateEDR action on the BCD actions library, to create a record for the bad PIN attempt. |
| 5. | VoucherCreateEDR action constructs an event (FlistSleeEvent) that contains a PCM_OP_ACT_ACTIVITY operation for the special voucher bad PIN event and sends it to BCD Client.  
| | • The BCD Client invokes the PCM_OP_ACT_ACTIVITY operation and sets a timer to the configured value for this type of operation. |
| 6. | BRM creates a record for the bad PIN event and responds to the operation by indicating that this has been done.  
| | • The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
| | • The BCD actions library translates the output flist into the response of the VoucherCreateEDR action.  
| | • The slee_acs process takes the Voucher Invalid branch of the Voucher Recharge feature node and reaches a Send Short Message Notification feature node, which specifies sending a failure text message.  
| | • The SSMN feature node creates an MMX GenericMessage containing the failure message and sends it to xmsTrigger. |
| 7. | xmsTrigger sends an MTForwardSM operation to the MSC, containing the failure message. |
| 8. | The MSC sends an SMS containing the failure message to the caller.
### Step 10
- The slee_acs process reaches an Accept feature node, which sends ReleaseCall to xmsTrigger.

### Step 11
- xmsTrigger sends a MOForwardSM result to the MSC.
- The MSC sends a notification of successful delivery of the original SMS to the caller.
- The slee_acs process reaches an end node and clears the call context.

**Messages: error on SMS recharge with invalid PIN**

The following messages include operations sent to BRM and results returned by BRM when an error occurs for a voucher recharge with an invalid PIN using SMS. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation:** send is PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

```plaintext
Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_EXTENDED_INFO
1  PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2     PIN_FLD_DIRECTION ENUM [0] 0
2     PIN_FLD_CELL_ID STR [0] "000c"
2     PIN_FLD_LOC_AREA_CODE STR [0] "06401000f"
1 10000 _SUBSTRUCT [0] allocated 20, used 2
2 10001  STR [0] "Present"
2  PIN_FLD_LOCATION STR [0] "004085752160"
0 PIN_FLD_MSID STR [0] "004085752160"
```

**Result:** received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

```plaintext
0 PIN_FLD_POID POID [0] 0.0.0.1 /balance_group 127667 6530
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_BILLINFO_OBJ POID [0] 0.0.0.1 /billinfo 129203 0
0 PIN_FLD_EFFECTIVE_T TSTAMP [0] (1337557354) Sun May 20 23:42:34 2012
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 8
1  PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1  PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1  PIN_FLD_CURRENT_BAL DECIMAL [0] 491.13
1  PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1  PIN_FLD_CREDIT_LIMIT DECIMAL [0] 500
1  PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1  PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_REALTIME_CNTR INT [0] 4
0 PIN_FLD_BALANCES ARRAY [100011] allocated 20, used 8
1  PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1  PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1  PIN_FLD_CURRENT_BAL DECIMAL [0] 70
1  PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1  PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1  PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1  PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_BALANCES ARRAY [1000076] allocated 20, used 8
1  PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1  PIN_FLD_NEXT_BAL DECIMAL [0] 0
1  PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1  PIN_FLD_CURRENT_BAL DECIMAL [0] -87.5
1  PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1  PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1  PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1  PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
```

**Operation:** send is PCM_OP_ACT_ACTIVITY (151)

```plaintext
Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
```
Chapter 8

Usage Scenarios

Result: received for operation PCM_OP_ACT_ACTIVITY (151)

0 PIN_FLD_POID POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_RESULTS ARRAY [0] allocated 20, used 4
1 PIN_FLD_BAL_IMPACTS ARRAY (1000011) allocated 20, used 8
  2 PIN_FLD_AMOUNT DECIMAL [0] 1
  2 PIN_FLDRESOURCE_ID INT [0] 1000011
  2 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 128819 0
  2 PIN_FLD_IMPACT_TYPE ENUM [0] 2
  2 PIN_FLD_GL_ID INT [0] 0
  2 PIN_FLD_ITEM_OBJ POID [0] 0.0.0.1 /item/misc 175517 0
  2 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 127667 6530
  2 PIN_FLD_LINEAGE STR [0] NULL str ptr
1 PIN_FLD_SUB_BAL_IMPACTS ARRAY (1000011) allocated 20, used 3
  2 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 127667 6531
  2 PIN_FLD_RESOURCE_ID INT [0] 1000011
  2 PIN_FLD_SUB_BALANCES ARRAY [4] allocated 20, used 3
  3 PIN_FLD_AMOUNT DECIMAL [0] 1
  3 PIN_FLD_VALID_FROM TSTAMP [0] (0) <null>
  3 PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
1 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 128819 0
1 PIN_FLD_POID POID [0] 0.0.0.1 /event/activity/voucher 272643699316532185 0
Error on SMS recharge with account topup failed

Error on SMS recharge with account topup failed flow
Here is an example flow for an error on an SMS recharge when an account topup failed.

Error on SMS recharge with account topup failed scenario
This scenario describes the sequence of messages that occurs when the subscriber sends an SMS to recharge a voucher but the action to recharge the subscriber's wallet on BRM fails. In this scenario, the account is on BRM while the voucher is in the NCC VWS.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | • MSC sends MOForwardSM to xmsTrigger.  
          • The subscriber has sent an SMS, which contains the voucher and PIN, from a GSM phone to the SMS recharge number. |
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 2    | xmsTrigger sends InitialDP to the slee_acs process, with the service key set to a special value indicating SMS recharge.  
- The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.  
- The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is in the BRM domain. It also determines the control plan to run, based on the serviceKey.  
- The slee_acs process runs the SMS recharge control plan.  
- The slee_acs process reaches an Extract Content feature node, which extracts the voucher number and PIN from the SMS and places them in temporary storage.  
- The slee_acs process reaches a Voucher Recharge feature node, which extracts the voucher number and PIN from temporary storage and uses them to invoke a VoucherRedeem action on the FOX actions library. |
| 3    | FOX actions library sends VR_Req to BeClient.  
- BeClient send VR_Req to VWS. |
| 4    | VWS sends VR_Ack to BeClient, indicating that the voucher has been reserved and returning the recharge amounts for each balance type.  
- BeClient send VR_Ack to the slee_acs process.  
- The Voucher Recharge feature node invokes a WalletRecharge action on the BCD actions library. |
| 5    | The slee_acs process creates an event (FlistSleeEvent) for a PCM_OPBILL_DEBIT operation and sends it to the BCD Client  
- The BCD Client invokes the PCM_OPBILL_DEBIT operation and sets a timer to the configured value for this type of operation. |
| 6    | BRM responds to the operation indicating that the account has not been successfully recharged.  
- The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
- The BCD actions library translates output flist into the error response of the WalletRecharge action.  
- The Voucher Recharge feature node invokes the VoucherRevoke action on the FOX actions library. |
| 7    | FOX actions library sends RVR_Req to BeClient.  
- BeClient sends RVR_Req to VWS. |
| 8    | VWS sends RVR_Ack to BeClient, indicating that the voucher reservation has been revoked.  
- BeClient sends RVR_Ack to the slee_acs process.  
- The slee_acs process takes the unsupported branch of the Voucher Recharge feature node and reaches a Send Short Message Notification feature node, which specifies sending an error text message. |
| 9    | The SSMN feature node constructs an MMX GenericMessage containing the failure message and sends it to xmsTrigger. |
| 10   | xmsTrigger sends an MTForwardSM operation to the MSC, containing the failure message. |
| 11   | The MSC sends an SMS containing the failure message to the caller. |
| 12   | The slee_acs process reaches an Accept feature node, which sends ReleaseCall to xmsTrigger. |
Chapter 8

Step | Action
--- | ---
13 | 
- xmsTrigger sends a MOForwardSM result to the MSC.
- The MSC sends a notification of the successful delivery of the original SMS to the caller.
- The slee_acs process reaches an end node and clears the call connection.

**Messages: error on SMS recharge with account topup failed**
The following messages include operations sent to BRM and results returned by BRM for an error on an SMS recharge when the account topup failed. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation:** send is PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

```
Flags = 0
0 PIN_FLD_POID          POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME  STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE      STR [0] "gsm/ncc"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1  PIN_FLD_GSM_INFO     SUBSTRUCT [0] allocated 20, used 3
2   PIN_FLD_DIRECTION   ENUM [0] 0
2   PIN_FLD_CELL_ID     STR [0] "0000c"
2   PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
1  10000                SUBSTRUCT [0] allocated 20, used 2
2   10001               STR [0] "Present"
2   PIN_FLD_LOCATION    STR [0] "004085752160"
0 PIN_FLD_MSID          STR [0] "004085752160"
```

**Result:** received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

```
0 PIN_FLD_POID          POID [0] 0.0.0.1 /balance_group 127667 6446
0 PIN_FLD_ACCOUNT_OBJ   POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLDBILLINFO_OBJ   POID [0] 0.0.0.1 /billinfo 129203 0
0 PIN_FLD_EFFECTIVE_T   TSTAMP [0] (1337557354) Sun May 20 23:42:34 2012
0 PIN_FLD_BALANCES      ARRAY [978] allocated 20, used 8
1   PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1   PIN_FLD_NEXT_BAL      DECIMAL [0] 0
1   PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1   PIN_FLD_CURRENT_BAL   DECIMAL [0] 400
1   PIN_FLD_CREDIT_FLOOR  DECIMAL [0] NULL
1   PIN_FLD_CREDIT_LIMIT  DECIMAL [0] 500
1   PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1   PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_REALTIME_CNTR  INT [0] 4
0 PIN_FLD_BALANCES      ARRAY [1000011] allocated 20, used 8
1   PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1   PIN_FLD_NEXT_BAL      DECIMAL [0] 0
1   PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1   PIN_FLD_CURRENT_BAL   DECIMAL [0] 70
1   PIN_FLD_CREDIT_FLOOR  DECIMAL [0] NULL
1   PIN_FLD_CREDIT_LIMIT  DECIMAL [0] 0
1   PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1   PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_BALANCES      ARRAY [1000076] allocated 20, used 8
1   PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1   PIN_FLD_NEXT_BAL      DECIMAL [0] 0
1   PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1   PIN_FLD_CURRENT_BAL   DECIMAL [0] -125
1   PIN_FLD_CREDIT_FLOOR  DECIMAL [0] NULL
1   PIN_FLD_CREDIT_LIMIT  DECIMAL [0] 0
1   PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1   PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
```

**Operation:** send is PCM_OP_BILL_DEBIT (105)
Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_DEBIT ARRAY [978] allocated 20, used 1
1 PIN_FLD_BAL_OPERAND DECIMAL [0] -0.500000000000000

Result: received for operation PCM_OP_BILL_DEBIT(105)

0 PIN_FLD_POID POID [0] 0.0.0.1 /error_poid 128819 0

Error specified:
location=5
pin_errclass=4
pin_err=3
field=83893688
rec_id=0
reserved=0
facility=0
msg_id=0
version=0
Error on USSD recharge with invalid PIN

Error on USSD recharge with invalid PIN flow
Here is an example flow for an error on a USSD recharge with an invalid PIN.

Error on USSD recharge with invalid PIN scenario
This scenario describes the sequence of messages that occurs when the subscriber sends a USSD message to recharge a voucher but the subscriber's pin is invalid and a USSD message is displayed on the caller's handset. In this scenario, the account is on BRM while the voucher is in the NCC VWS.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | • MSC sends processUnstructuredSS-Request to USSD Gateway  
      • The calling subscriber has sent a USSD message that contains the service code for a voucher recharge, the voucher number and PIN. |
| 2    | • The USSD Gateway sends InitialDP to the slee_acs process, using the USSD voucher recharge service key as determined by the service code configuration.  
      • The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. It also determines the control plan to run, using the serviceKey to select the USSD voucher recharge control plan.  
      • The slee_acs process runs the control plan.  
      • The slee_acs process reaches a Voucher Recharge feature node. |
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 3    | • The Voucher Recharge feature node sends ConnectToResource, PromptAndCollectUserInformation to USSD Gateway, instructing it to prompt the caller for the voucher number and PIN.  
  • USSD Gateway retrieves the voucher number and PIN from the received processUnstructuredSS-Request. |
| 4    | • USSD Gateway sends the PromptAndCollectUserInformation result, containing the voucher number and PIN, to the slee_acs process.  
  • The Voucher Recharge feature node invokes the VoucherRedeem action on the FOX actions library.  
  • The FOX actions library sends VR_Req to BeClient  
  • BeClient sends VR_Req to VWS. |
| 5    | • VWS sends VR_Nack to BeClient, indicating that the voucher PIN is incorrect.  
  • The Voucher Recharge feature node invokes a VoucherBadPIN action on the BCD actions library.  
  • The VoucherBadPIN action does nothing and returns success.  
  • The Voucher Recharge feature node invokes the VoucherCreateEDR action on the BCD actions library to create a record for the bad PIN attempt. |
| 6    | • VoucherCreateEDR action constructs a FlistSleeEvent containing a PCM_OP_ACT_ACTIVITY operation for the special voucher bad PIN event and sends it to BCD Client.  
  • The BCD Client invokes the PCM_OP_ACT_ACTIVITY operation and sets a timer to the configured value for this type of operation. |
| 7    | • BRM creates a record for the bad PIN event and responds to the operation by indicating that this has been done.  
  • The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
  • The BCD actions library translates the output flist into the response of the VoucherCreateEDR action. |
| 8    | • The Voucher Recharge feature node sends PlayAnnouncement to USSD Gateway.  
  • USSD Gateway translates the announcement ID into text stating that the recharge was not successful, based on the configuration. |
| 9    | • USSD Gateway sends the processUnstructuredSS-Request result to the MSC.  
  • USSD text stating that the recharge was not successful is displayed on the caller's handset.  
  • USSD Gateway sends a SpecializedResourceReport to the slee_acs process.  
  • The control plan reaches an end node and ACS sends DisconnectForwardConnection and ReleaseCall actions to the USSD Gateway and clears the call connection. |
| 10   | • USSD Gateway translates the announcement ID into text stating that the recharge was not successful, based on the configuration. |
| 11   | • USSD Gateway sends the processUnstructuredSS-Request result to the MSC.  
  • USSD text stating that the recharge was not successful is displayed on the caller's handset.  
  • USSD Gateway sends a SpecializedResourceReport to the slee_acs process.  
  • The control plan reaches an end node and ACS sends DisconnectForwardConnection and ReleaseCall actions to the USSD Gateway and clears the call connection. |

**Messages: error on USSD recharge with invalid PIN**

The following messages include operations sent to BRM and results returned by BRM for an error on a USSD recharge with an invalid PIN. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation:** send is 

```
PCM_OP_TCF_AAA_QUERYBALANCE (4104)
```

**Flags = 0**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 PIN_FLD_POID</td>
<td>POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0</td>
</tr>
<tr>
<td>0 PIN_FLD_PROGRAM_NAME</td>
<td>STR [0] &quot;NCC_BCD_Client&quot;</td>
</tr>
</tbody>
</table>
Chapter 8

0 PIN_FLD_OBJ_TYPE        STR (0) "gsm/ncc"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1  PIN_FLD_GSM_INFO  SUBSTRUCT [0] allocated 20, used 3
2   PIN_FLD_DIRECTION   ENUM [0] 0
2   PIN_FLD_CELL_ID     STR [0] "000c"
2   PIN_FLD_LOC_AREA_CODE STR [0] "06401000f"
1 10000  SUBSTRUCT [0] allocated 20, used 2
2  10001  STR [0] "Present"
2   PIN_FLD_LOCATION    STR [0] "004085752160"
0 PIN_FLD_MSID            STR [0] "004085752160"

Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

0 PIN_FLD_POID           POID [0] 0.0.0.1 /balance_group 127667 6530
0 PIN_FLD_ACCOUNT_OBJ  POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_BILLINFO_OBJ  POID [0] 0.0.0.1 /billinfo 129203 0
0 PIN_FLD_EFFECTIVE_T  TSTAMP [0] (1337557354) Sun May 20 23:42:34 2012
0 PIN_FLD_BALANCES      ARRAY [978] allocated 20, used 8
1   PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1   PIN_FLD_CURRENT_BAL   DECIMAL [0] 491.13
1   PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1   PIN_FLD_CREDIT_LIMIT DECIMAL [0] 500
1   PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1   PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_BALANCES      ARRAY [1000011] allocated 20, used 8
1   PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1   PIN_FLD_CURRENT_BAL   DECIMAL [0] -87.5
1   PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1   PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1   PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1   PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""

Operation: send is PCM_OP_ACT_ACTIVITY (151)

Flags = 0
0 PIN_FLD_POID           POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_PROGRAM_NAME    STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE        STR [0] "/voucher"
0 PIN_FLD_INHERITED_INFO SUBSTRUCT [0] allocated 20, used 1
1 10000  SUBSTRUCT [0] allocated 20, used 2
2  10007  STR [0] "9876543214"
2  10008  STR [0] "3210"
0 PIN_FLD_BAL_IMPACTS   ARRAY [1000011] allocated 20, used 8
1   PIN_FLD_AMOUNT       DECIMAL [0] 1
1   PIN_FLD_RESOURCE_ID   INT [0] 1000011
1   PIN_FLD_ACCOUNT_OBJ  POID [0] 0.0.0.1 /account 128819 0
1   PIN_FLD_IMPACT_TYPE   ENUM [0] 2

Result: received for operation PCM_OP_ACT_ACTIVITY (151)

0 PIN_FLD_POID           POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_RESULTS       ARRAY [0] allocated 20, used 4
1   PIN_FLD_BAL_IMPACTS   ARRAY [1000011] allocated 20, used 8
2   PIN_FLD_AMOUNT       DECIMAL [0] 1
2   PIN_FLD_RESOURCE_ID   INT [0] 1000011
2   PIN_FLD_ACCOUNT_OBJ  POID [0] 0.0.0.1 /account 128819 0
2   PIN_FLD_IMPACT_TYPE   ENUM [0] 2
2   PIN_FLD_GL_ID         INT [0] 0

184  NCC BRM Charging Driver Technical Guide
Error on USSD recharge with account topup failed

Here is an example message flow for an error on a USSD Recharge with an account topup that failed.

![Diagram of USSD recharge message flow](image-url)

**Origin MSC**

1. **processUnstructured SSRequest**

**USSD Gateway**

2. **InitialDP**
   - **ConnectToResource**
   - **PromptAndCollectUserInformation**
   - **collectedDigits**

**slee_acs/ BCD Client**

3. **VR Req**
   - **VR Ack**

**VVS**

4. **PCM_OP_BILL_DEBIT**
   - **PCM_OP_BILL_DEBIT error**

**BRM Connection Manager**

5. **RVR Req**
   - **RVR Ack**

6. **PlayAnnouncement**

7. **specializedResourceReport**

8. **DisconnectForwardConnection**

9. **ReleaseCall**
Error on USSD recharge with account topup failed scenario
This scenario describes the sequence of messages that occurs when the subscriber sends a USSD message requesting a voucher recharge but the wallet recharge action fails on BRM and a USSD text message is displayed on the caller’s display. In this scenario, the account in on BRM while the voucher is in the NCC VWS.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1.0  | MSC sends processUnstructuredSS-Request to USSD Gateway  
      | The calling subscriber has sent a USSD message that contains the service code for a voucher recharge, the voucher number and PIN. |
| 2.0  | The USSD Gateway sends InitialDP to the slee_acs process, using the USSD voucher recharge service key as determined by the service code configuration.  
      | The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. It also determines the control plan to run, using the serviceKey to select the USSD voucher recharge control plan.  
      | The slee_acs process runs the control plan.  
      | The slee_acs process reaches a Voucher Recharge feature node. |
| 3.0  | The Voucher Recharge feature node sends ConnectToResource, PromptAndCollectUserInformation to USSD Gateway, instructing it to prompt the caller for the voucher number and PIN.  
      | USSD Gateway retrieves the voucher number and PIN from the received processUnstructuredSS-Request.  
      | USSD Gateway sends the PromptAndCollectUserInformation result, containing the voucher number and PIN, to the slee_acs process.  
      | The Voucher Recharge feature node invokes the VoucherRedeem action on the FOX actions library. |
| 4.0  | The FOX actions library sends VR_Req to BeClient.  
      | BeClient sends VR_Req to VWS. |
| 5.0  | VWS sends VR_Ack to BeClient, indicating that the voucher has been reserved, and returning the recharge amounts for each balance type.  
      | BeClient sends VR_Ack to the slee_acs process.  
      | The Voucher Recharge feature node invokes the WalletRecharge action on the BCD actions library. |
| 6.0  | The slee_acs process creates a FlistSleeEvent for a PCM_OP_BILL_DEBIT operation and sends it to the BCD Client.  
      | The BCD actions library creates an event (FlistSleeEvent) that contains a PCM_OP_BILL_DEBIT operation and sends it to the Client.  
      | The BCD Client invokes the PCM_OP_BILL_DEBIT operation and sets a timer to the configured value for this type of operation.  
      | BRM responds to the operation by indicating that the account has not been successfully recharged.  
      | The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
      | The BCD actions library translates the output flist into the response of the WalletRecharge action.  
      | The Voucher Recharge feature node invokes the VoucherRevoke action on the FOX actions library. |
| 7.0  | FOX actions library sends RVR_Req to BeClient.  
      | BeClient sends RVR_Req to VWS. |
Chapter 8

Usage Scenarios

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 10   | - VWS sends RVR_Ack to BeClient, indicating that the voucher reservation has been revoked.  
      - BeClient sends RVR_Ack to the slee_acs process. |
| 11   | - Voucher Recharge node sends PlayAnnouncement to USSD Gateway.  
      - USSD Gateway translates the announcement ID into some text that states that the recharge was not successful, based on the configuration. |
| 12   | - USSD Gateway sends processUnstructuredSS-Request result to the MSC.  
      - USSD text stating that the recharge was not successful is displayed on the caller's handset. |
| 13   | - USSD Gateway sends SpecializedResourceReport to the _slee_acs process. |
| 14   | - The control plan reaches an end node and ACS sends DisconnectForwardConnection and ReleaseCall actions to the USSD Gateway and clears the call connection. |

Messages: error on USSD recharge with account topup failed

The following messages include operations sent to BRM and results returned by BRM for an error on a USSD recharge with a failed account topup. The general message format is: nesting level (0; 1, or 2); field; data type; value.

Operation: send is PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

Flags = 0  
0 FIN_FLDBPOID POID [0] 0.0.0.1 /service/telco/gsm/telephony = 1 0  
0 FIN_FLDPограм_NAME STR [0] "NCC_BCD_Client"  
0 FIN_FLD Somehow_TYPE STR [0] "gsm/ncc"  
0 FIN_FLDEXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2  
1 FIN_FLDSOMETIME_INFO SUBSTRUCT [0] allocated 20, used 3  
2 FIN_FLD_DIRECTIONS ENUM [0] 0  
2 FIN_FLD_CELL_ID STR [0] "000c"  
2 FIN_FLD_LOC_AREA_CODE STR [0] "064001000f"  
1 10000 SUBSTRUCT [0] allocated 20, used 2  
2 10001 STR [0] "Present"  
2 FIN_FLDLOCATION STR [0] "004085752160"  
0 FIN_FLDMSID STR [0] "004085752160"

Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

0 FIN_FLDBPOID POID [0] 0.0.0.1 /balance_group 127667 6446  
0 FIN_FLDACCOUNT_OBJ POID [0] 0.0.0.1 /account 128819 0  
0 FIN_FLDBILLINFO_OBJ POID [0] 0.0.0.1 /billinfo 129203 0  
0 FIN_FLDEDIFFERENT T TSTAMP [0] (1337557354) Sun May 20 23:42:34 2012  
0 FIN_FLDBALANCES ARRAY [978] allocated 20, used 8  
1 FIN_FLDSRESERVED_AMOUNT_DECIMAL [0] 0  
1 FIN_FLD_NEXT_BAL DECIMAL [0] 0  
1 FIN_FLD_CONSUMPTION_RULE ENUM [0] 0  
1 FIN_FLD_CURRENT_BAL_DECIMAL [0] 400  
1 FIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL  
1 FIN_FLD_CREDIT_LIMIT DECIMAL [0] 500  
1 FIN_FLD_CREDIT_THRESHOLDS INT [0] 0  
1 FIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 FIN_FLD_REALTIME_CNTR INT [0] 4  
0 FIN_FLD_BALANCES ARRAY [1000011] allocated 20, used 8  
1 FIN_FLD_RESERVED_AMOUNT_DECIMAL [0] 0  
1 FIN_FLDNEXT_BAL DECIMAL [0] 0  
1 FIN_FLD_CONSUMPTION_RULE ENUM [0] 0  
1 FIN_FLD_CURRENT_BAL DECIMAL [0] 70  
1 FIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL  
1 FIN_FLD_CREDIT_LIMIT DECIMAL [0] 0  
1 FIN_FLD_CREDIT_THRESHOLDS INT [0] 0  
1 FIN_FLD_CREDIT_THRESHOLDS FIXED STR [0] ""
0 FIN_FLD_BALANCES ARRAY [1000076] allocated 20, used 8
Chapter 8

```
1  PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1  PIN_FLD_NEXT_BAL DECIMAL [0] 0
1  PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1  PIN_FLD_CURRENT_BAL DECIMAL [0] -125
1  PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1  PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1  PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1  PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""

Operation: send is PCM_OP_BILL_DEBIT (105)

Flags = 0
0  PIN_FLD_POID  POID [0] 0.0.0.1 /account 128819 0
0  PIN_FLD_PROGRAM_NAME  STR [0] "NCC_BCD_Client"
0  PIN_FLD_OBJ_TYPE  STR [0] "gsm/ncc"
0  PIN_FLD_DEBIT  ARRAY [978] allocated 20, used 1
1  PIN_FLD_BAL_OPERAND  DECIMAL [0] -0.500000000000000

Result: received for operation PCM_OP_BILL_DEBIT(105)

0  PIN_FLD_POID  POID [0] 0.0.0.1 /error_poid 128819 0

Error specified:
location=5
pin_errclass=4
pin_err=3
field=83893688
rec_id=0
reserved=0
facility=0
msg_id=0
version=0

Messages: top up with invalid voucher number or PIN
The following messages include operations sent to BRM and results returned by BRM for a top up request using a bad voucher number or PIN. The general message format is: nesting level (0; 1, or 2); field; data type; value.

Result: Received for operation PCM_OP_PYMT_TOPUP (3726)

Error specified:
location=5
pin_errclass=4
pin_err=3
field=83893688
rec_id=0
reserved=0
facility=0
msg_id=0
version=0

The error code of 3 is defined like this:
#define PIN_ERR_NOT_FOUND                    3

Operation: Send PCM_OP_READ_FLDS (4)

Flags = 0
0  PIN_FLD_POID  POID [0] 0.0.0.1 /account 135267 0
0  PIN_FLD_CURRENCY  INT [0] 0
0  PIN_FLD_STATUS  ENUM [0] 0
0  PIN_FLD_CLOSE_WHEN_T TSTAMP [0] (0) <null>
0  PIN_FLD_MOD_T  TSTAMP [0] (0) <null>
0  PIN_FLD_LAST_STATUS_T TSTAMP [0] (0) <null>

Result: Received for operation PCM_OP_READ_FLDS (4)

0  PIN_FLD_POID  POID [0] 0.0.0.1 /account 135267 8
0  PIN_FLD_CURRENCY  INT [0] 978
0  PIN_FLD_STATUS  ENUM [0] 10100
0  PIN_FLD_CLOSE_WHEN_T TSTAMP [0] (0) <null>
0  PIN_FLD_LAST_STATUS_T TSTAMP [0] (1331675046) Tue Mar 13 21:44:06 2012
```
Post call charging of voice call

Post call charging of voice call flow
Here is an example message flow for post call charging of a voice call.

Post call charging of voice call scenario
This scenario describes the sequence of messages that occurs when the subscriber is charged after hanging up.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | • The MSC sends an InitialDP operation to the SLC.  
     | • The subscriber has made a call attempt.   
     | • The slee_acs process on the SLC receives the InitialDP and passes it to the CCS service loader.  
     | • The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. It also determines the control plan to run.  
     | • The slee_acs process runs the post call charging control plan.  
     | • The slee_acs process reaches an Attempt Terminate to Pending with Duration (ATPD) feature node with the maximum call duration set to one day. |
| 2    | • The slee_acs process sends RequestReportBCSMEvent, ApplyCharging(1 day) and Connect operations to the MSC.  
     | • MSC connects the call to the called party.  
     | • The called party answers.  
     | • After 30 minutes, the calling party hangs up.  |
| 3    | • MSC sends ApplyChargingReport(callActive=false,30 minutes), EventReportBCSM(disconnect) to the slee_acs process.  
     | • The control plan reaches a Direct Unit Charge (DUCR) feature node.  
     | • The slee_acs process invokes a DirectTimeCharge action on the BCD actions library. |
Step | Action
---|---
4. | The BCD actions library creates an event (BcdSleeEvent) to invoke a PCM_OP_TCF_AAA_ACCOUNTING operation, indicating that 30 minutes of total talk time have been used, and sends it to the least busy BCD Client.
   - The BCD Client finds a free connection on the least busy Connection Manager.
   - The BCD Client invokes the PCM_OP_TCF_AAA_ACCOUNTING operation and sets a timer to the configured value for this type of operation.
5. | BRM responds to the operation indicating that the cost of 30 minutes of talk time has been permanently deducted from the account.
   - The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.
   - The BCD actions library translates the output flist into the response of the DirectTime action. It also stores the charge information in the call context for possible later use.
   - The slee_acs process proceeds with the control plan. If there are no more feature nodes, processing for this call ends.

Messages: post call charging of voice call
The following messages include operations sent to BRM and results returned by BRM for a voice call with post call charging. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation:** send is PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

Flags = 0
0  PIN_FLD_POID         POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0  PIN_FLD_PROGRAM_NAME  STR [0] "NCC_BCD_Client"
0  PIN_FLD_SESSION_ID    INT [0] 0
0  PIN_FLD_AUTHORIZATION_ID  STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1381284_0"
0  PIN_FLD_OBJ_TYPE      STR [0] "gsm/ncc"
0  PIN_FLD_CALLING_NUMBER  STR [0] "004085752159"
0  PIN_FLD_CALLED_NUMBER  STR [0] "55587390000"
0  PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1     PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2          PIN_FLD_DIRECTION ENUM [0] 0
2          PIN_FLD_CELL_ID  STR [0] "0000c"
2          PIN_FLD_LOC_AREA_CODE  STR [0] "064001000f"
1     10000 SUBSTRUCT [0] allocated 20, used 2
2          PIN_FLD_LOCATION  STR [0] "004085752159"
0  PIN_FLD_MSID         STR [0] "004085752159"
0  PIN_FLD_REQ_MODE     ENUM [0] 2
0  PIN_FLD_QUANTITY     DECIMAL [0] 15.000000000000000

**Result:** received for operation PCM_OP_TCF_AAA_STOP_ACCOUNTING (4007)

0  PIN_FLD_POID         POID [0] 0.0.0.1 /event/session/telco/gsm/ncc 277516734850840473 0
0  PIN_FLD_AUTHORIZATION_ID  STR [0] "brmClient-tbalcomb-2013-3-12-6_session_1381284_0"
0  PIN_FLD_ACCOUNT_OBJ  POID [0] 0.0.0.1 /account 114343 0
0  PIN_FLD_SERVICE_OBJ  POID [0] 0.0.0.1 /service/telco/gsm/telephony 112551 18
0  PIN_FLD_RATING_STATUS ENUM [0] 0
0  PIN_FLD_BALANCES     ARRAY [978] allocated 20, used 2
1     PIN_FLD_AMOUNT DECIMAL [0] 0.150
1     PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 9.850
Successful service subscription

Successful service subscription flow
Here is an example message flow for a successful service subscription flow.

Successful service subscription scenario
This scenario describes the sequence of messages that occurs when the caller calls the toll free number to subscribe to a service.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MSC sends InitialDP to the slee_acs process, with serviceKey set to a special value indicating “subscribe to service”.</td>
</tr>
<tr>
<td></td>
<td>The subscriber has dialed the “subscribe to service” toll free number.</td>
</tr>
<tr>
<td></td>
<td>The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. It also determines the control plan to run, using the serviceKey to select the “subscribe to service” control plan.</td>
</tr>
<tr>
<td></td>
<td>The slee_acs process runs the control plan.</td>
</tr>
<tr>
<td></td>
<td>The slee_acs process reaches a Named Event feature node, specifying named event rates for a non-reservable event.</td>
</tr>
<tr>
<td>Step</td>
<td>Action</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
</tr>
</tbody>
</table>
| 2    | - The BCD actions library constructs a FlistSleeEvent containing a PCM_OP_ACT_ACTIVITY operation and sends it to BCD Client.  
   - The BCD actions library constructs an event (FlistSleeEvent) that contains the PCM_OP_ACT_ACTIVITY operation and sends it to the BCD Client.  
   - The BCD Client invokes the PCM_OP_ACT_ACTIVITY operation and sets a timer to the configured value for this type of operation.  
| 3    | - BRM responds to the operation, giving the cost of the deal.  
   - The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
   - The BCD actions library translates the output flist into the response of the NamedEventRates action.  
   - The Named Event feature node stores the cost in call context, takes the success exit and reaches a Play Variable Part Announcement node.  
| 4    | - The slee_acs process sends ConnectToResource, PlayAnnouncement to the MSC, with a variable part indicating the cost of the deal.  
   - MSC plays the announcement to the caller.  
| 5    | - The announcement finishes and MSC sends a SpecializedResourceReport message to the slee_acs process.  
   - The SDR feature node takes an exit to aNamed Event feature node that specifies a direct named event for the same type of non-reservable event.  
| 6    | - The SDR feature node takes an exit to aNamed Event feature node that specifies a direct named event for the same type of non-reservable event.  
   - The BCD actions library constructs an event (FlistSleeEvent) that contains a PCM_OP_SUBSCRIPTION_PURCHASE_DEAL operation and sends it to BCD Client.  
   - Sometimes the BCD actions library will send PCM_OP_CUST_POL_GET_DEALS before sending PCM_OP_SUBSCRIPTION_PURCHASE_DEAL if it has no knowledge of the requested deal or if the cached information about the deal is too old.  
   - The BCD Client invokes the PCM_OP_SUBSCRIPTION_PURCHASE_DEAL operation and sets a timer to the configured value for this type of operation.  
| 7    | - BRM responds to the operation indicating that the deal has been purchased.  
   - The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
   - The BCD actions library translates the output flist into the response of the DirectNamedEvent action.  
   - The Named Event feature node takes the success exit and reaches a Play Announcement feature node for a success announcement.  
| 8    | - A Play Announcement feature node sends PlayAnnouncement to the MSC.  
   - MSC plays an announcement to the caller, stating that the transaction was successful.  
| 9    | - The announcement finishes and MSC sends a SpecializedResourceReport message to the slee_acs process.  

192  NCC BRM Charging Driver Technical Guide
Step | Action
--- | ---
12 | The control plan reaches an end node and ACS sends DisconnectForwardConnection and ReleaseCall actions to MSC and clears the call context.
| The caller is disconnected.

**Messages: successful service subscription**
The following messages include operations sent to BRM and results returned by BRM for a successful service subscription. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation: send is PCM_OP_TCF_AAA_QUERY_BALANCE (4104)**

Flags = 0

- 0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
- 0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
- 0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
- 0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
  - 1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
    - 2 PIN_FLD_DIRECTION ENUM [0] 0
    - 2 PIN_FLD_CELL_ID STR [0] "000c"
    - 2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
  - 10000 SUBSTRUCT [0] allocated 20, used 2
    - 10001 STR [0] "Present"
  - 2 PIN_FLD_LOCATION STR [0] "004085752160"
- 0 PIN_FLD_MSID STR [0] "004085752160"

**Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)**

- 0 PIN_FLD_POID POID [0] 0.0.0.1 /balance_group 127667 6798
- 0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.0 /account 128819 0
- 0 PIN_FLD_BILLINFO_OBJ POID [0] 0.0.0.1 /billinfo 129203 0
- 0 PIN_FLD_EFFECTIVE_T TSTAMP [0] (1337557354) Sun May 20 23:42:34 2012
- 0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 8
  - 1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  - 1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
  - 1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
  - 1 PIN_FLD_CURRENT_BAL DECIMAL [0] 10
  - 1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
  - 1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
  - 1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
  - 1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
- 0 PIN_FLD_REALTIME_CNTR INT [0] 4
- 0 PIN_FLD_BALANCES ARRAY [1000011] allocated 20, used 8
  - 1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  - 1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
  - 1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
  - 1 PIN_FLD_CURRENT_BAL DECIMAL [0] 77
  - 1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
  - 1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
  - 1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
  - 1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
- 0 PIN_FLD_BALANCES ARRAY [1000076] allocated 20, used 8
  - 1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  - 1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
  - 1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
  - 1 PIN_FLD_CURRENT_BAL DECIMAL [0] 125.37
  - 1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
  - 1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
  - 1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
  - 1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""

**Operation: send is PCM_OP_ACT_ACTIVITY (151)**

Flags = 128

- 0 PIN_FLD_POID POID [0] 0.0.0.1 /account 128819 0
- 0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
- 0 PIN_FLD_OBJ_TYPE STR [0] "/gsm/ncc"
**Result:** received for operation PCM_OP_ACT_ACTIVITY (151)

| PIN_FLD_POID | POID [0] | 0.0.0.1 | /account 128819 0 |
| PIN_FLD_RESULTS | ARRAY [0] | allocated 27, used 27 |
| PIN_FLD_POID | POID [0] | 0.0.0.1 | /event/activity/gsm/ncc -1 0 |
| PIN_FLD_SESSION_OBJ | STR [0] | "Activity Tracking Event Log" |
| PIN_FLD_SESSION_OBJ | POID [0] | 0.0.0.1 | /servicepcm_client 1, 34764 |
| PIN_FLD_SESSION_OBJ | POID [0] | 0.0.0.1 | /session 27643699316532386 0 |
| PIN_FLD_ACCOUNT_OBJ | POID [0] | 0.0.0.1 | /account 128819 0 |
| PIN_FLD_PROGRAM_NAME | STR [0] | "NCC_BCD_Client" |
| PIN_FLD_SESSION_T | TSTAMP [0] | (1339108525) | Thu Jun 07 22:35:25 2012 |
| PIN_FLD_END_T | TSTAMP [0] | (1339108525) | Thu Jun 07 22:35:25 2012 |
| PIN_FLD_FLAGS | INT [0] | 128 |
| PIN_FLD_SYS_DESCR | STR [0] | "Activity: /gsm/ncc" |
| PIN_FLD_BILLINFO_OBJ | POID [0] | 0.0.0.1 | /billinfo 129203 0 |
| PIN_FLD_RUM_NAME | STR [0] | "Occurrence" |
| PIN_FLD_UNIT | Enum [0] | 0 |
| PIN_FLD_TOD_MODE | Enum [0] | 0 |
| PIN_FLD_NET_QUANTITY | Decimal [0] | 1 |
| PIN_FLD_MIN_QUANTITY | Decimal [0] | 1 |
| PIN_FLD_INCR_QUANTITY | Decimal [0] | 1 |
| PIN_FLD_MIN_UNIT | Enum [0] | 0 |
| PIN_FLD_INCR_UNIT | Enum [0] | 0 |
| PIN_FLD_RATING_STATUS | Enum [0] | 0 |
| PIN_FLD_SUB_BAL_IMPACTS | ARRAY [0] | allocated 20, used 18 |
| PIN_FLD_BAL_IMPACTS | ARRAY [0] | allocated 20, used 18 |
| PIN_FLD_BAL_GRP_OBJ | POID [0] | 0.0.0.1 | /balance_group 127667, 6798 |
| PIN_FLD_BAL_GRP_OBJ | POID [0] | 0.0.0.1 | /billinfo 129203 0 |
| PIN_FLD_TAX_CODE | STR [0] | "US/Pacific" |
| PIN_FLD_TAX_CODE | STR [0] | "US/Pacific" |
| PIN_FLD_RATE_OBJ | POID [0] | 0.0.0.1 | /rate 79259 1 |
| PIN_FLD_RATE_OBJ | POID [0] | 0.0.0.1 | /billinfo 129203 0 |
| PIN_FLD_RATE_TAG | STR [0] | "Rate 1" |
| PIN_FLD_RATE_TAG | STR [0] | "Rate 1" |
| PIN_FLD_IMPACT_TYPE | Enum [0] | 1 |
| PIN_FLD_IMPACT CATEGORY | STR [0] | "default" |
| PIN_FLD_IMPACT CATEGORY | STR [0] | "default" |
| PIN_FLD_OFFERING_OBJ | POID [0] | 0.0.0.1 | /purchased_product 129779 176 |
| PIN_FLD_LINEAGE | STR [0] | NULL str ptr |
| PIN_FLD_GL_ID | INT [0] | 0 |
| PIN_FLD_RESOURCE_ID | INT [0] | 978 |
| PIN_FLD_RESOURCE_ID | INT [0] | 978 |
| PIN_FLD_AMOUNT | DECIMAL [0] | 0.200 |
| PIN_FLD_AMOUNT | DECIMAL [0] | 0.200 |
| PIN_FLD_BALANCES | ARRAY [0] | allocated 20, used 3 |
| PIN_FLD_BALANCES | ARRAY [0] | allocated 20, used 3 |
| PIN_FLD_TOTAL | ARRAY [978] | allocated 20, used 1 |
| PIN_FLD_TOTAL | ARRAY [978] | allocated 20, used 1 |

**Operation:** send is PCM_OP_CUST_POL_GET_DEALS (278)

Flags = 0

| PIN_FLD_POID | POID [0] | 0.0.0.1 | /account 128819 0 |
| PIN_FLD_RESULTS | ARRAY [10] | allocated 20, used 14 |
| PIN_FLD_MOD_T | TSTAMP [0] | (1327455952) | Wed Jan 25 01:45:52 2012 |
| PIN_FLD_READ_ACCESS | STR [0] | "G" |
| PIN_FLD_WRITE_ACCESS | STR [0] | "G" |
```
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_ACCOUNT_OBJ</td>
<td>POID [0]</td>
<td>0.0.0.1 /account 1 0</td>
</tr>
<tr>
<td>PIN_FLD_CODE</td>
<td>STR [0]</td>
<td>&quot;$50 Euro Topup Deal&quot;</td>
</tr>
<tr>
<td>PIN_FLD_DESCR</td>
<td>STR [0]</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>PIN_FLD_END_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_FLAGS</td>
<td>INT [0]</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_NAME</td>
<td>STR [0]</td>
<td>&quot;$50 Euro Topup Deal&quot;</td>
</tr>
<tr>
<td>PIN_FLD_PERMITTED</td>
<td>STR [0]</td>
<td>&quot;/account&quot;</td>
</tr>
<tr>
<td>PIN_FLD_START_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_PRODUCTS</td>
<td>ARRAY [0]</td>
<td>allocated 20, used 19</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_DISCOUNT</td>
<td>DECIMAL [0]</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_END_DETAILS</td>
<td>INT [0]</td>
<td>2</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_END_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_START_DETAILS</td>
<td>INT [0]</td>
<td>1</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_START_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_FILE_NAME</td>
<td>STR [0]</td>
<td>&quot;50 Euro Topup Deal&quot;</td>
</tr>
<tr>
<td>PIN_FLD_PERMITTED</td>
<td>STR [0]</td>
<td>&quot;/account&quot;</td>
</tr>
<tr>
<td>PIN_FLD_START_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_PRODUCTS</td>
<td>ARRAY [0]</td>
<td>allocated 20, used 19</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_DISCOUNT</td>
<td>DECIMAL [0]</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_END_DETAILS</td>
<td>INT [0]</td>
<td>2</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_END_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_START_DETAILS</td>
<td>INT [0]</td>
<td>1</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_START_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_QUANTITY</td>
<td>DECIMAL [0]</td>
<td>1</td>
</tr>
<tr>
<td>PIN_FLD_STATUS</td>
<td>ENUM [0]</td>
<td>1</td>
</tr>
<tr>
<td>PIN_FLD_STATUS_FLAGS</td>
<td>INT [0]</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_USAGE_DISCOUNT</td>
<td>DECIMAL [0]</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_USAGE_END_DETAILS</td>
<td>INT [0]</td>
<td>2</td>
</tr>
<tr>
<td>PIN_FLD_USAGE_END_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_USAGE_START_DETAILS</td>
<td>INT [0]</td>
<td>1</td>
</tr>
<tr>
<td>PIN_FLD_USAGE_START_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_POID</td>
<td>POID [0]</td>
<td>0.0.0.1 /deal 88008 4</td>
</tr>
<tr>
<td>PIN_FLD_MOD_T</td>
<td>TSTAMP [0]</td>
<td>(13277455952) Wed Jan 25 01:45:52 2012</td>
</tr>
<tr>
<td>PIN_FLD_READ_ACCESS</td>
<td>STR [0]</td>
<td>&quot;B&quot;</td>
</tr>
<tr>
<td>PIN_FLD_WRITE_ACCESS</td>
<td>STR [0]</td>
<td>&quot;S&quot;</td>
</tr>
<tr>
<td>PIN_FLD_ACCOUNT_OBJ</td>
<td>POID [0]</td>
<td>0.0.0.1 /account 1 0</td>
</tr>
<tr>
<td>PIN_FLD_DESCR</td>
<td>STR [0]</td>
<td>&quot;DirectEventDeal&quot;</td>
</tr>
<tr>
<td>PIN_FLD_END_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_FLAGS</td>
<td>INT [0]</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_NAME</td>
<td>STR [0]</td>
<td>&quot;DirectEventDeal&quot;</td>
</tr>
<tr>
<td>PIN_FLD_PERMITTED</td>
<td>STR [0]</td>
<td>&quot;/account&quot;</td>
</tr>
<tr>
<td>PIN_FLD_START_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_PRODUCTS</td>
<td>ARRAY [0]</td>
<td>allocated 20, used 19</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_DISCOUNT</td>
<td>DECIMAL [0]</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_END_DETAILS</td>
<td>INT [0]</td>
<td>2</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_END_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_START_DETAILS</td>
<td>INT [0]</td>
<td>1</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_START_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_FILE_NAME</td>
<td>STR [0]</td>
<td>&quot;50 Euro Topup Deal&quot;</td>
</tr>
<tr>
<td>PIN_FLD_PERMITTED</td>
<td>STR [0]</td>
<td>&quot;/account&quot;</td>
</tr>
<tr>
<td>PIN_FLD_START_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_PRODUCTS</td>
<td>ARRAY [0]</td>
<td>allocated 20, used 19</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_DISCOUNT</td>
<td>DECIMAL [0]</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_END_DETAILS</td>
<td>INT [0]</td>
<td>2</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_END_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_START_DETAILS</td>
<td>INT [0]</td>
<td>1</td>
</tr>
<tr>
<td>PIN_FLD_CYCLE_START_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_QUANTITY</td>
<td>DECIMAL [0]</td>
<td>1</td>
</tr>
<tr>
<td>PIN_FLD_STATUS</td>
<td>ENUM [0]</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_STATUS_FLAGS</td>
<td>INT [0]</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_USAGE_DISCOUNT</td>
<td>DECIMAL [0]</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_USAGE_END_DETAILS</td>
<td>INT [0]</td>
<td>2</td>
</tr>
<tr>
<td>PIN_FLD_USAGE_END_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_USAGE_START_DETAILS</td>
<td>INT [0]</td>
<td>1</td>
</tr>
<tr>
<td>PIN_FLD_USAGE_START_T</td>
<td>TSTAMP [0]</td>
<td>(0) &lt;null&gt;</td>
</tr>
<tr>
<td>PIN_FLD_POID</td>
<td>POID [0]</td>
<td>0.0.0.1 /deal 88008 4</td>
</tr>
</tbody>
</table>

Operation: send is PCM_OP_SUBSCRIPTION_PURCHASE_DEAL (108)
Flags = 0
```
Result: received for operation PCM_OP_SUBSCRIPTION_PURCHASE_DEAL (108)

0  PIN_FLD_POID       POID [0] 0.0.0.1 /account 128819 0
0  PIN_FLD_RESULTS   ARRAY [0] allocated 20, used 9
 1  PIN_FLD_BAL_IMPACTS ARRAY [0] allocated 20, used 18
     2  PIN_FLD_ACCOUNT_OBJ   POID [0] 0.0.0.1 /account 128819 338
     2  PIN_FLD_PRODUCT_OBJ   POID [0] 0.0.0.1 /product 84328 7
     2  PIN_FLD_TAX_CODE      STR [0] ""
     2  PIN_FLD_RATE_OBJ      POID [0] 0.0.0.1 /rate 79451 1
     2  PIN_FLD_RATE_TAG      STR [0] "Rate 1"
     2  PIN_FLD_IMPACT_CATEGORY STR [0] "default"
     2  PIN_FLD_OFFERING_OBJ  POID [0] NULL poid pointer
     2  PIN_FLD_LINEAGE        STR [0] NULL str ptr
     2  PIN_FLD_GL_ID         INT [0] 0
     2  PIN_FLD_QUANTITY      DECIMAL [0] 1.00000000
     2  PIN_FLD_IMPACT_TYPE   ENUM [0] 1
     2  PIN_FLD_DISCOUNT      DECIMAL [0] 0
     2  PIN_FLD_PERCENT       DECIMAL [0] 1
     2  PIN_FLD_AMOUNT        DECIMAL [0] 0.300
     2  PIN_FLD_RESOURCE_ID   INT [0] 978
     2  PIN_FLD_AMOUNT_DEFERRED DECIMAL [0] 0
     2  PIN_FLD_BAL_GRP_OBJ   POID [0] 0.0.0.1 /balance Group 127667 6798
     2  PIN_FLD_ITEM_OBJ      POID [0] 0.0.0.1 /item/misc 129715 0
 1  PIN_FLD_SUB_BAL_IMPACTS ARRAY [0] allocated 20, used 3
     2  PIN_FLD_BAL_GRP_OBJ   POID [0] 0.0.0.1 /balance Group 127667 6799
     2  PIN_FLD_RESOURCE_ID   INT [0] 978
     2  PIN_FLD_SUB_BALANCES  ARRAY [0] allocated 20, used 3
        3  PIN_FLD_AMOUNT      DECIMAL [0] 0.300
        3  PIN_FLD_VALID_FROM   TSTAMP [0] (0) <null>
        3  PIN_FLD_VALID_TO     TSTAMP [0] (0) <null>
 1  PIN_FLD_UNRATED QUANTITY DECIMAL [0] 0
 1  PIN_FLD_SERVICE_OBJ   POID [0] 0.0.0.0 0 0
 1  PIN_FLD_ACCOUNT_OBJ   POID [0] 0.0.0.1 /account 128819 0
 1  PIN_FLD_RATING_STATUS ENUM [0] 0
 1  PIN_FLD_NET_QUANTITY DECIMAL [0] 1
 1  PIN_FLD_RUM_NAME      STR [0] "Occurrence"
 1  PIN_FLD_POID         POID [0] 0.0.0.1 /event/billing/product/fee/purchase 27264369316538418 0
 0  PIN_FLD_RESULTS   ARRAY [1] allocated 20, used 3
     1  PIN_FLD_ACCOUNT_OBJ   POID [0] 0.0.0.0 0 0
     1  PIN_FLD_RATE_OBJ      POID [0] 0.0.0.1 /account 128819 0
     1  PIN_FLD_POID         POID [0] 0.0.0.1 /event/billing/product/action/purchase 17592186221106 0
 0  PIN_FLD_PRODUCTS  ARRAY [0] allocated 20, used 2
     1  PIN_FLD_PRODUCT_OBJ   POID [0] 0.0.0.1 /product 84328 7
     1  PIN_FLD_PACKAGE_ID   INT [0] 260
 0  PIN_FLD_RESULTS   ARRAY [2] allocated 20, used 2
     1  PIN_FLD_ACCOUNT_OBJ   POID [0] 0.0.0.1 /account 128819 0
     1  PIN_FLD_POID         POID [0] 0.0.0.1 /event/billing/deal/purchase 17592186223154 0
Service subscription - not allowed to purchase deal

Service subscription - not allowed to purchase deal flow
Here is an example message flow for the subscriber not being allowed to purchase a deal.

Service subscription - not allowed to purchase deal scenario
This scenario describes the sequence of messages that occurs when the subscriber is not allowed to purchase a deal.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | - MSC sends InitialDP to the slee_acs process, with serviceKey set to a special value indicating “subscribe to service”.
      | - The subscriber has dialed the "subscribe to service" toll free number.
      | - The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines that the wallet information is on the BRM domain. It also determines the control plan to run, using the serviceKey to select the "subscribe to service" control plan.
      | - The slee_acs process runs the control plan.
      | - The slee_acs process reaches a Selection Dependent Routing feature node. |
| 2    | - The Selection Dependent Routing feature node sends ConnectToResource, PromptAndCollectUserInformation to the MSC, instructing the MSC to prompt for a single digit from a menu.
      | - MSC plays the specified announcement to the caller and collects the menu choice. |
| 3    | - MSC sends PromptAndCollectUserInformation result, containing the menu choice digit, to the slee_acs process.
<pre><code>  | - The SDR feature node takes an exit to a Named Event feature node and specifies a direct named event for a non-reservable event. |
</code></pre>
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 4    | - The BCD actions library constructs an event (FlistSleeEvent) that contains a PCM_OP_SUBSCRIPTION_PURCHASE_DEAL operation and sends it to the BCD Client.  
- The BCD Client invokes the PCM_OP_SUBSCRIPTION_PURCHASE_DEAL operation and sets a timer to the configured value for this type of operation. |
| 5    | - BRM responds to the operation with an error.  
- The BCD Client receives the operation error output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.  
- The BCD actions library translates the error output flist into the error response of the DirectNamedEvent action.  
- The Named Event feature node takes an error exit and reaches a Play Announcement node for an error announcement. |
| 6    | - A Play Announcement feature node sends PlayAnnouncement to the MSC.  
- MSC plays an announcement to the caller, stating that the transaction was not successful. |
| 7    | - The announcement finishes and MSC sends SpecializedResourceReport to the slee_acs process. |
| 8    | - The control plan reaches an end node and ACS sends DisconnectForwardConnection and ReleaseCall actions to MSC and clears the call context.  
- The caller is disconnected. |

**Messages: service subscription - not allowed to purchase deal**

The following messages include operations sent to BRM and results returned by BRM for service subscription when the subscriber is not allowed to purchase a deal. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation: send is PCM_OP_TCF_AAA_QUERY_BALANCE (4104)**

```
Flags = 0
0  PIN_FLD_POID       POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0  PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0  PIN_FLD_OBJ_TYPE    STR [0] "gsm/ncc"
0  PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
  1  PIN_FLD_GSM_INFO   SUBSTRUCT [0] allocated 20, used 3
  2  PIN_FLD_DIRECTION  ENUM [0] 0
  2  PIN_FLD_CELL_ID    STR [0] "000c"
  2  PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
  1  10000               SUBSTRUCT [0] allocated 20, used 3
  2  10001               STR [0] "Present"
  2  PIN_FLD_LOCATION   STR [0] "004085752160"
  2  10002               INT [0] 1
0  PIN_FLD_MSID        STR [0] "004085752160"
0  PIN_FLD_FLAGS       INT [0] 4
```

**Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)**

```
0  PIN_FLD_POID       POID [0] 0.0.0.1 /balance_group 127667 6801
0  PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 128819 0
0  PIN_FLD_BILLINFO_OBJ POID [0] 0.0.0.1 /billinfo 129203 0
0  PIN_FLD_EFFECTIVE_T TSTAMP [0] (1337557354) Sun May 20 23:42:34 2012
0  PIN_FLD_BALANCES   ARRAY [978] allocated 20, used 10
  1  PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
  1  PIN_FLD_NEXT_BAL    DECIMAL [0] 0
  1  PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
  1  PIN_FLD_CURRENT_BAL  DECIMAL [0] 0
  1  PIN_FLD_SUB_BALANCES ARRAY [0] allocated 20, used 12
  2  PIN_FLD_CONTRIBUTOR_STR STR [0] ""
  2  PIN_FLD_VALID_TO    TSTAMP [0] (0) <null>
```
PIN_FLD_VALID_TO DETAILS INT [0] 0
2 PIN_FLD_VALID_FROM TSTAMP [0] (1337497200) Sun May 20 07:00:00 2012
2 PIN_FLD_VALID_FROM DETAILS INT [0] 0
2 PIN_FLD_NEXT_BAL DECIMAL [0] 0
2 PIN_FLD_DELAYED_BAL DECIMAL [0] 0
2 PIN_FLD_ROLLOVER_DATA INT [0] 0
2 PIN_FLD_GRANTOR_OBJ POID [0] 0.0.0.1 /purchased_product 130995 0
2 PIN_FLD_STATUS ENUM [0] 1
2 PIN_FLD_FLAGS INT [0] 2
1 PIN_FLD_CURRENT_TOTAL DECIMAL [0] 0
1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_REALTIME_CNTR INT [0] 4
0 PIN_FLD_BALANCES ARRAY [1000011] allocated 20, used 10
1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1 PIN_FLD_CURRENT_BAL DECIMAL [0] -125.37
2 PIN_FLD_SUB_BALANCES ARRAY [4] allocated 20, used 12
2 PIN_FLD_CONTRIBUTOR_STR STR [0] ""
2 PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
2 PIN_FLD_VALID_TO_DETAILS INT [0] 0
2 PIN_FLD_VALID_FROM TSTAMP [0] (1337497200) Sun May 20 07:00:00 2012
2 PIN_FLD_VALID_FROM_DETAILS INT [0] 0
2 PIN_FLD_CURRENT_BAL DECIMAL [0] -125.37
2 PIN_FLD_NEXT_BAL DECIMAL [0] 0
2 PIN_FLD_DELAYED_BAL DECIMAL [0] 0
2 PIN_FLD_ROLLOVER_DATA INT [0] 0
2 PIN_FLD_GRANTOR_OBJ POID [0] 0.0.0.0 0 0
2 PIN_FLD_STATUS ENUM [0] 1
2 PIN_FLD_FLAGS INT [0] 2
1 PIN_FLD_CURRENT_TOTAL DECIMAL [0] -125.37
1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_BALANCES ARRAY [1000076] allocated 20, used 10
1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1 PIN_FLD_CURRENT_BAL DECIMAL [0] -125.37
2 PIN_FLD_SUB_BALANCES ARRAY [2] allocated 20, used 12
2 PIN_FLD_CONTRIBUTOR_STR STR [0] ""
2 PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
2 PIN_FLD_VALID_TO_DETAILS INT [0] 0
2 PIN_FLD_VALID_FROM TSTAMP [0] (1337497200) Sun May 20 07:00:00 2012
2 PIN_FLD_VALID_FROM_DETAILS INT [0] 0
2 PIN_FLD_CURRENT_BAL DECIMAL [0] -125.37
2 PIN_FLD_NEXT_BAL DECIMAL [0] 0
2 PIN_FLD_DELAYED_BAL DECIMAL [0] 0
2 PIN_FLD_ROLLOVER_DATA INT [0] 0
2 PIN_FLD_GRANTOR_OBJ POID [0] 0.0.0.1 /purchased_product 130995 205
2 PIN_FLD_STATUS ENUM [0] 1
2 PIN_FLD_FLAGS INT [0] 2
1 PIN_FLD_CURRENT_TOTAL DECIMAL [0] -125.37
1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""

Operation: send is PCM_OP_SUBSCRIPTION_PURCHASE_DEAL (108)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /account 128819 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_DEAL_INFO SUBSTRUCT [0] allocated 20, used 1
1 PIN_FLD_DEAL_OBJ POID [0] 0.0.0.1 /deal 88008 4
Chapter 8

Result: received for operation PCM_OP_SUBSCRIPTION_PURCHASE_DEAL (108)

0 PIN_FLD_POID POID [0] 0.0.0.1 /account 128819 0

Error specified:
location=5
pin_errclass=1
pin_err=81
field=0
rec_id=0
reserved=0
facility=0
msg_id=0
version=0
Successful credit transfer using IVR

Successful credit transfer using IVR flow
Here is an example message flow for a successful credit transfer using IVR.
**Successful credit transfer using IVR scenario**

This scenario describes the sequence of messages that occurs when the subscriber completes a successful credit transfer using IVR.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | - Subscriber has called the special “gift 100 free minutes” number.  
      - MSC sends InitialDP to the slee_acs process, with serviceKey set to a special value indicating “gift 100 free minutes”.  
      - xmsTrigger sends InitialDP to the slee_acs process.  
      - The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines the control plan to run, using the serviceKey to select the “gift 100 free minutes” control plan.  
      - The slee_acs process runs the control plan.  
      - The slee_acs process reaches a Collect Digits To Buffer node. |
| 2    | - The Collect Digits To Buffer feature node sends ConnectToResource, PromptAndCollectUserInformation to the MSC, instructing the MSC to prompt the caller for his PIN.  
      - MSC plays the specified announcement to the caller and collects the PIN. |
| 3    | - MSC sends the PromptAndCollectUserInformation result, which contains the PIN, to the slee_acs process.  
      - The slee_acs process stores the PIN and then reaches another Collect Digits To Buffer feature node. |
| 4    | - The Collect Digits To Buffer feature node sends PromptAndCollectUserInformation to the MSC, instructing the MSC to prompt the caller for the recipient's number.  
      - MSC plays the specified announcement to the caller and collects the recipient's number. |
| 5    | - MSC sends PromptAndCollectUserInformation result, containing the recipient's number to the slee_acs process.  
      - The slee_acs process stores the recipient's number and reaches a Credit Wallet Transfer feature node.  
      - The Credit Wallet Transfer node performs the following actions, all of which are implemented on the (FOX) subscriber domain using local database queries:  
        - AlternateWalletDetails  
        - AlternateSubscriberDetails  
      - The Credit Wallet Transfer feature node invokes the (wallet domain) GetCreditTransferDetails action on the BCD actions library. In spite of being in the BCD actions library, this action also does a local database look-up in exactly the same way as the equivalent FOX action.  
      - The Credit Wallet Transfer feature node invokes the (FOX) subscriber domain CompareSubscriberPINDetails action, which compares the entered PIN with the PIN stored in the subscriber’s profile.  
      - The Credit Wallet Transfer feature node invokes the NamedEventReservation action on BCD actions library.  
      - BCD actions library creates an (FlistSleeEvent) that contains a PCM_OP_TCF_AAA_AUTHORIZE operation, with quantity set to 1 and units set to time (the default). SVC_TYPE is set to Gift100Mins (The name of the named event).  
      - The BCD Client finds a free connection on the least busy Connection Manager. |
| 6    | - The BCD Client invokes the PCM_OP_TCF_AAA_AUTHORIZE operation and sets a timer to the configured value for this type of operation. |
### Step 7
- BRM responds to the operation by indicating that the charge for this service has been reserved against the caller’s account.
- The BCD Client receives the operation output flist via and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.
- The BCD actions library translates the output flist into the response of the NamedEventReservation action.
- The Credit Wallet Transfer feature node invokes the VoucherTypeRedeem action on the FOX actions library.

### Step 8
- The FOX actions library sends VI_Req to BeClient.
- BeClient sends VI_Req to VWS.

### Step 9
- VWS sends VI_Ack to BeClient, indicating that the voucher type exists, and returning the recharge amounts for each balance type.
- BeClient sends VI_Ack to the slee_acs process.
- The Credit Wallet Transfer feature node invokes a VoucherTypeWalletRecharge action on the BCD actions library.

### Step 10
- The BCD actions library constructs an event (FlistSleeEvent) that contains a PCM_OP_QUERY_BALANCE operation and sends it to the BCD Client.
- The BCD Client invokes the PCM_OP_QUERY_BALANCE operation and sets a timer for the configured value for this type of operation.

### Step 11
- BRM responds to the operation indicating that the account exists and gives the account object ID.
- BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.

### Step 12
- The BCD actions library constructs an event (FlistSleeEvent) that contains a PCM_OP_BILL_DEBIT operation and sends it to the BCD Client.
- The BCD Client invokes the PCM_OP_BILL_DEBIT operation and sets a timer to the configured value for this type of operation.

### Step 13
- BRM responds to the operation indicating that the account has been successfully recharged.
- The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.
- The BCD actions library translates the output flist and into the response of the VoucherTypeWalletRecharge action.
- The Credit Wallet Transfer feature node invokes the ConfirmNamedEventReservation action on BCD actions library.
- The BCD actions library constructs an event (FlistSleeEvent) that contains a PCM_OP_TFC_AAA_ACCOUNTING operation.

### Step 14
- The BCD Client invokes the PCM_OP_TFC_AAA_ACCOUNTING operation and sets a timer to the configured value for this type of operation.

### Step 15
- BRM responds to the operation indicating that the charge for this service has been made successfully against the caller’s account.
- The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer.
- The BCD actions library translates the output flist into the response of the ConfirmNamedEventReservation action.
Chapter 8, Usage Scenarios

### Step 16
- The Credit Wallet Transfer feature node constructs an MMX GenericMessage containing a "you have been gifted 100 minutes" message for the recipient and sends it to xmsTrigger.

### Step 17
- xmsTrigger sends an MTForwardSM operation to the MSC, containing the "you have been gifted 100 minutes" message.

### Step 18
- The MSC sends an SMS containing the "you have been gifted 100 minutes" message to the recipient.
- The slee_acs process takes the success branch of the Credit Wallet Transfer feature node.

### Step 19
- The slee_acs process reaches a Play Announcement feature node, which sends PlayAnnouncement to the MSC.
- MSC plays an announcement to the caller, which states that the 100 minutes have been gifted.

### Step 20
- The Announcement finishes and MSC sends a SpecializedResourceReport message to slee_acs process.

### Step 21
- The control plan reaches an end node and ACS sends DisconnectForwardConnection and ReleaseCall actions to MSC and clears the call context.
- The caller is disconnected.

#### Messages: successful credit transfer using IVR scenario
The following messages include operations sent to BRM and results returned by BRM for a successful credit transfer using IVR. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation:** send is PCM_OP_TCF_AAA_AUTHORIZE (4002)

```
Flags = 0
0 PIN_FLD_POID        POID [0] 0.0.0.1 /service/telco/gsm/sms -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID   INT [0] 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmclient-tbalcomb-2013-3-12-6_ne_1461366_0"
0 PIN_FLD_TYPE         STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "08001234567"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
  1 PIN_FLD_GSM_INFO     SUBSTRUCT [0] allocated 20, used 3
    2 PIN_FLD_DIRECTION   ENUM [0] 0
    2 PIN_FLD_CELL_ID     STR [0] "000c"
    2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
  1 10000                SUBSTRUCT [0] allocated 20, used 3
    2 10001               STR [0] "Present"
    2 PIN_FLD_LOCATION    STR [0] "004085752159"
    2 10002               INT [0] 1
0 PIN_FLD_MSID           STR [0] "004085752159"
0 PIN_FLD_REQ_MODE       ENUM [0] 8
```

**Result:** received for operation PCM_OP_TCF_AAA_AUTHORIZE (4002)

```
0 PIN_FLD_POID        POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 189822 0
0 PIN_FLD_EXPIRATION_T TSTAMP [0] (1363910537) Fri Mar 22 00:02:17 2013
0 PIN_FLD_QUANTITY     DECIMAL [0] 1
0 PIN_FLD_SERVICE_OBJ  POID [0] 0.0.0.1 /service/telco/gsm/sms 112359 20
0 PIN_FLD_RESERVATION_OBJ POID [0] 0.0.0.1/reservation 191870 0
0 PIN_FLD_BAL_GRP_OBJ  POID [0] 0.0.0.1/balance_group 114087 1096
0 PIN_FLD_BALANCES     ARRAY [978] allocated 20, used 2
  1 PIN_FLD_AMOUNT      DECIMAL [0] 0.200
  1 PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 4.800
0 PIN_FLD_RESULT       ENUM [0] 1
0 PIN_FLD_RATING_STATUS ENUM [0] 0
```
Chapter 8

0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_ne_1461366_0"
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 114343 0

Operation: send PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/telephony -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "08001234567"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
  2 PIN_FLD_DIRECTION ENUM [0] 0
  2 PIN_FLD_CELL_ID STR [0] "000c"
  2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
1 10000 SUBSTRUCT [0] allocated 20, used 3
  2 10001 STR [0] "Present"
  2 10002 INT [0] 1
0 PIN_FLD_MSID STR [0] "004085752152"

Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

0 PIN_FLD_POID POID [0] 0.0.0.1 /balance_group 181459 8
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 182099 0
0 PIN_FLD_BILLINFO_OBJ POID [0] 0.0.0.1 /billinfo 184147 0
0 PIN_FLD_EFFECTIVE_T TSTAMP [0] (1363815368) Wed Mar 20 21:36:08 2013
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 8
1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1 PIN_FLD_CURRENT_BAL DECIMAL [0] 0
1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_REALTIME_CNTR INT [0] 2
0 PIN_FLD_BALANCES ARRAY [1000076] allocated 20, used 8
1 PIN_FLD_RESERVED_AMOUNT DECIMAL [0] 0
1 PIN_FLD_NEXT_BAL DECIMAL [0] 0
1 PIN_FLD_CONSUMPTION_RULE ENUM [0] 0
1 PIN_FLD_CURRENT_BAL DECIMAL [0] 0
1 PIN_FLD_CREDIT_FLOOR DECIMAL [0] NULL
1 PIN_FLD_CREDIT_LIMIT DECIMAL [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS INT [0] 0
1 PIN_FLD_CREDIT_THRESHOLDS_FIXED STR [0] ""
0 PIN_FLD_BAL_IMPACTS ARRAY [1000076] allocated 20, used 17
2 PIN_FLD_IMPACT_TYPE ENUM [0] 2
2 PIN_FLD_RESOURCE_ID INT [0] 1000076
2 PIN_FLD_RESOURCE_ID_ORIG INT [0] 0
2 PIN_FLD_RATE_CODE STR [0] ""
2 PIN_FLD_RATE_TAG STR [0] ""
2 PIN_FLD_DISCOUNT DECIMAL [0] 0
2 PIN_FLD_PERCENT DECIMAL [0] 0

Operation: send PCM_OP_BILL_DEBIT (105)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /account 182099 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "08001234567"
0 PIN_FLD_DEBIT ARRAY [1000076] allocated 20, used 1
1 PIN_FLD_BAL_OPERAND DECIMAL [0] -1.00000000000000000000000

Result: received for operation PCM_OP_BILL_DEBIT (105)

0 PIN_FLD_POID POID [0] 0.0.0.1 /account 182099 0
0 PIN_FLD_RESULTS ARRAY [0] allocated 20, used 4
1 PIN_FLD_BAL_IMPACTS ARRAY [0] allocated 20, used 17
2 PIN_FLD_IMPACT_TYPE ENUM [0] 2
2 PIN_FLD_RESOURCE_ID INT [0] 1000076
2 PIN_FLD_RESOURCE_ID_ORIG INT [0] 0
2 PIN_FLD_RATE_CODE STR [0] ""
2 PIN_FLD_RATE_TAG STR [0] ""
2 PIN_FLD_DISCOUNT DECIMAL [0] 0
2 PIN_FLD_PERCENT DECIMAL [0] 0
Chapter 8

2 PIN_FLD_QUANTITY DECIMAL [0] 0
2 PIN_FLD_AMOUNT_DEFERRED DECIMAL [0] 0
2 PIN_FLD_AMOUNT DECIMAL [0] -1.00
2 PIN_FLD_AMOUNT_ORIG DECIMAL [0] NULL pin_decimal_t ptr
2 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /Balance_group 181459 0
2 PIN_FLD_GL_ID INT [0] 0
2 PIN_FLD_ITEM_OBJ POID [0] 0.0.0.1 /item/sponsor 182259 0
2 PIN_FLD_LINEAGE STR [0] NULL str ptr
1 PIN_FLD_SUB_BAL_IMPACTS ARRAY [0] allocated 20, used 3
2 PIN_FLD_BAL_GRP_OBJ POID [0] 0.0.0.1 /balance_group 181459 9
2 PIN_FLD_RESOURCE_ID INT [0] 1000076
2 PIN_FLD_SUB_BALANCES ARRAY [2] allocated 20, used 3
3 PIN_FLD_AMOUNT DECIMAL [0] -1.00
3 PIN_FLD_VALID_FROM TSTAMP [0] (0) <null>
3 PIN_FLD_VALID_TO TSTAMP [0] (0) <null>
1 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 182099 0
1 PIN_FLD_POID POID [0] 0.0.0.1 /event/billing/debit 277675064525243563 0

Operation: send PCM_OP_TCF_AAA_ACCOUNTING (4012)

Flags - 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/sms -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 1
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_ne_1461366_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "08001234567"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1 PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2 PIN_FLD_DIRECTION ENUM [0] 0
2 PIN_FLD_CELL_ID STR [0] "0000c"
2 PIN_FLD_LOC_AREA_CODE STR [0] "064001000f"
1 10000 SUBSTRUCT [0] allocated 20, used 3
2 10001 STR [0] "Present"
2 10002 INT [0] 1
0 PIN_FLD_MSID STR [0] "004085752159"
0 PIN_FLD_MODE ENUM [0] 0

Result: received for operation PCM_OP_TCF_AAA_ACCOUNTING (4012)

0 PIN_FLD_POID POID [0] 0.0.0.1 /event/activity/telco/gsm/ncc 277675064525251874 0
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_ne_1461366_0"
0 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 114343 0
0 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/telco/gsm/sms 112359 20
0 PIN_FLD_RATING_STATUS ENUM [0] 0
0 PIN_FLD_BALANCES ARRAY [978] allocated 20, used 2
1 PIN_FLD_AMOUNT DECIMAL [0] 0.200
1 PIN_FLD_AVAILABLE_RESOURCE_LIMIT DECIMAL [0] 4.800
IVR credit transfer when recipient's account not found

IVR credit transfer when recipient's account not found flow
Here is an example message flow for an IVR credit transfer when the recipient's account is not found.

IVR credit transfer when recipient's account not found scenario
This scenario describes the sequence of messages that occur when the subscriber calls the toll free line to gift minutes but the recipient's account is not found.
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | • MSC sends InitialDP to the slee_acs process, with serviceKey set to a special value indicating “gift 100 free minutes”.  
    • Subscriber has called the special “gift 100 free minutes” number.  
    • xmsTrigger sends InitialDP to the slee_acs process.  
    • The CCS service loader looks up the subscriber and wallet in the SCP database on the SLC and determines the control plan to run, using the serviceKey to select the “gift 100 free minutes” control plan.  
    • The slee_acs process runs the control plan.  
    • The slee_acs process reaches a Collect Digits To Buffer node. |
| 2    | • The Collect Digits To Buffer feature node sends ConnectToResource, PromptAndCollectUserInformation to the MSC, instructing the MSC to prompt the caller for his PIN.  
    • MSC plays the specified announcement to the caller and collects the PIN. |
| 3    | • MSC sends the PromptAndCollectUserInformation result, which contains the PIN, to the slee_acs process.  
    • The slee_acs process stores the PIN and then reaches another Collect Digits To Buffer feature node. |
| 4    | • The Collect Digits To Buffer node sends PromptAndCollectUserInformation to the MSC, instructing the MSC to prompt the caller for the recipient's number.  
    • MSC plays the specified announcement to the caller and collects the recipient's number. |
| 5    | • MSC sends PromptAndCollectUserInformation result, containing the recipient's number to the slee_acs process.  
    • The slee_acs process stores the recipient's number and reaches a Credit Wallet Transfer feature node.  
    • The Credit Wallet Transfer node performs the following actions, all of which are implemented on the (FOX) subscriber domain using local database queries:  
      - AlternateWalletDetails  
      - AlternateSubscriberDetails  
    • The Credit Wallet Transfer feature node invokes the (wallet domain) GetCreditTransferDetails action on the BCD actions library. In spite of being in the BCD actions library, this action also does a local database look-up in exactly the same way as the equivalent FOX action.  
    • The Credit Wallet Transfer feature node invokes the (FOX) subscriber domain CompareSubscriberPINDetails action, which compares the entered PIN with the PIN stored in the subscriber's profile.  
    • The Credit Wallet Transfer feature node invokes the NamedEventReservation action on BCD actions library.  
    • The BCD actions library creates an event (FlisSleeEvent) containing a PCM_OP_TCF_AAA_AUTHORIZE operation with quantity set to 1 and units set to time (the default). SVC_TYPE is set to Gift100Mins (The name of the named event).  
    • The BCD Client finds a free connection on the least busy Connection Manager. |
| 6    | • The BCD Client invokes the PCM_OP_TCF_AAA_AUTHORIZE operation and sets a timer to the configured value for this type of operation.
Step | Action
--- | ---
7 | BRM responds to the operation by indicating that the charge for this service has been reserved against the caller's account. The BCD Client receives the operation output flist via and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer. The BCD actions library translates the output flist into the response of the NamedEventReservation action. The Credit Wallet Transfer feature node invokes the VoucherTypeRedeem action on the FOX actions library.
8 | The FOX actions library sends VI_Req to BeClient. BeClient sends VI_Req to VWS.
9 | VWS sends VI_Ack to BeClient, indicating that the voucher type exists, and returning the recharge amounts for each balance type. BeClient sends VI_Ack to the slee_acs process. The Credit Wallet Transfer feature node invokes a VoucherTypeWalletRecharge action on the BCD actions library.
10 | The BCD actions library constructs an event (FlistSleeEvent) that contains a PCM_OP_QUERY_BALANCE operation and sends it to the BCD Client. The BCD Client invokes the PCM_OP_QUERY_BALANCE operation and sets a timer for the configured value for this type of operation.
11 | BRM responds to the PCM_OP_QUERY_BALANCE operation indicating that the recipient's account cannot be found. The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer. The BCD actions library translates the output flist into the error response of the VoucherTypeWalletRecharge action. The Credit Wallet Transfer feature node invokes the RevokeNamedEventReservation action on the BCD actions library. The BCD actions library constructs an event (FlistSleeEvent) that contains a PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION operation.
12 | The BCD Client invokes the PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION operation and sets a timer to the configured value for this type of operation.
13 | BRM responds to the operation indicating that the reservation of the charge for this service has been revoked. The BCD Client receives the operation output flist and sends it in an event (BcdSleeEvent) to the slee_acs process. It also marks the BRM connection as free for reuse and cancels the operation timer. The BCD actions library takes the output flist and translates it to the response of the RevokeNamedEventReservation action.
14 | The slee_acs process takes the Error branch of the Credit Wallet Transfer feature node and reaches a Play Announcement feature node which sends PlayAnnouncement to the MSC. MSC plays an announcement to the caller, stating that the transfer was not successful.
15 | The announcement finishes and MSC sends a SpecializedResourceReport message to the slee_acs process.
Step | Action
--- | ---
16 | 
- The control plan reaches an end node and ACS sends DisconnectForwardConnection and ReleaseCall actions to MSC and clears the call context.
- The caller is disconnected

Messages: IVR credit transfer when recipient's account not found
The following messages include operations sent to BRM and results returned by BRM for an IVR credit transfer when the recipient’s account cannot be found on BRM. The general message format is: nesting level (0; 1, or 2); field; data type; value.

**Operation:** send PCM_OP_TCF_AAA_AUTHORIZE (4002)

Flags = 0

<table>
<thead>
<tr>
<th>Flags</th>
<th>POID [0]</th>
<th>0.0.0.1 /service/telco/gsm/sms -1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_SESSION_ID</td>
<td>STR [0]</td>
<td>&quot;NCC_BCD_Client&quot;</td>
</tr>
<tr>
<td>PIN_FLD_AUTHORIZATION_ID</td>
<td>STR [0]</td>
<td>&quot;brmClient-tbalcomb-2013-3-12-6_ne_1462367_0&quot;</td>
</tr>
<tr>
<td>PIN_FLD_CALLING_NUMBER</td>
<td>STR [0]</td>
<td>&quot;004085752159&quot;</td>
</tr>
<tr>
<td>PIN_FLD_CALLED_NUMBER</td>
<td>STR [0]</td>
<td>&quot;08001234567&quot;</td>
</tr>
<tr>
<td>PIN_FLD_EXTENDED_INFO</td>
<td>SUBSTRUCT [0] allocated 20, used 2</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_MSID</td>
<td>STR [0]</td>
<td>&quot;004085752159&quot;</td>
</tr>
<tr>
<td>PIN_FLD_REQ_MODE</td>
<td>ENUM [0]</td>
<td>8</td>
</tr>
</tbody>
</table>

**Operation:** send PCM_OP_TCF_AAA_AUTHORIZE (4002)

Result: received for operation PCM_OP_TCF_AAA_AUTHORIZE (4002)

<table>
<thead>
<tr>
<th>Flag</th>
<th>POID [0]</th>
<th>0.0.0.1 /active_session/telco/gsm/ncc 187153 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_EXPIRATION_T</td>
<td>TSTAMP [0]</td>
<td>(1363911209) Fri Mar 22 00:13:29 2013</td>
</tr>
<tr>
<td>PIN_FLD_SERVICE_OBJ</td>
<td>POID [0]</td>
<td>0.0.0.1 /service/telco/gsm/sms 112359 20</td>
</tr>
<tr>
<td>PIN_FLD_RESERVATION_OBJ</td>
<td>POID [0]</td>
<td>0.0.0.1 /reservation 186129 0</td>
</tr>
<tr>
<td>PIN_FLD_BAL_GRP_OBJ</td>
<td>POID [0]</td>
<td>0.0.0.1 /balance_group 114087 1102</td>
</tr>
<tr>
<td>PIN_FLD_BALANCES</td>
<td>ARRAY [978] allocated 20, used 2</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_RESULT</td>
<td>ENUM [0]</td>
<td>1</td>
</tr>
<tr>
<td>PIN_FLD_RATING_STATUS</td>
<td>ENUM [0]</td>
<td>0</td>
</tr>
<tr>
<td>PIN_FLD_AUTHORIZATION_ID</td>
<td>STR [0]</td>
<td>&quot;brmClient-tbalcomb-2013-3-12-6_ne_1462367_0&quot;</td>
</tr>
<tr>
<td>PIN_FLD_ACCOUNT_OBJ</td>
<td>POID [0]</td>
<td>0.0.0.1 /account 114343 0</td>
</tr>
</tbody>
</table>

**Operation:** send PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

Flags = 0

<table>
<thead>
<tr>
<th>Flag</th>
<th>POID [0]</th>
<th>0.0.0.1 /service/telco/gsm/telephony -1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_PROGRAM_NAME</td>
<td>STR [0]</td>
<td>&quot;NCC_BCD_Client&quot;</td>
</tr>
<tr>
<td>PIN_FLD_OBJ_TYPE</td>
<td>STR [0]</td>
<td>&quot;gsm/ncc&quot;</td>
</tr>
<tr>
<td>PIN_FLD_CALLING_NUMBER</td>
<td>STR [0]</td>
<td>&quot;004085752159&quot;</td>
</tr>
<tr>
<td>PIN_FLD_CALLED_NUMBER</td>
<td>STR [0]</td>
<td>&quot;08001234567&quot;</td>
</tr>
<tr>
<td>PIN_FLD_EXTENDED_INFO</td>
<td>SUBSTRUCT [0] allocated 20, used 2</td>
<td></td>
</tr>
<tr>
<td>PIN_FLD_MSID</td>
<td>STR [0]</td>
<td>&quot;0040857521591&quot;</td>
</tr>
</tbody>
</table>
Result: received for operation PCM_OP_TCF_AAA_QUERY_BALANCE (4104)

0 PIN_FLD_POID POID [0] 0.0.0.1 /error_poid -1 0

Operation: send PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION (4004)

Flags = 0
0 PIN_FLD_POID POID [0] 0.0.0.1 /service/telco/gsm/sms -1 0
0 PIN_FLD_PROGRAM_NAME STR [0] "NCC_BCD_Client"
0 PIN_FLD_SESSION_ID INT [0] 1
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_ne_1462367_0"
0 PIN_FLD_OBJ_TYPE STR [0] "gsm/ncc"
0 PIN_FLD_CALLING_NUMBER STR [0] "004085752159"
0 PIN_FLD_CALLED_NUMBER STR [0] "08001234567"
0 PIN_FLD_EXTENDED_INFO SUBSTRUCT [0] allocated 20, used 2
1  PIN_FLD_GSM_INFO SUBSTRUCT [0] allocated 20, used 3
2  PIN_FLD_DIRECTION ENUM [0] 0
2  PIN_FLD_CELL_ID STR [0] "000c"
2  PIN_FLD_LOC_AREA_CODE STR [0] "0640100001"
1  10000 SUBSTRUCT [0] allocated 20, used 3
2  10001 STR [0] "Present"
2  PIN_FLD_LOCATION STR [0] "004085752159"
2  10002 INT [0] 1
0 PIN_FLD_MSID STR [0] "004085752159"

Result: received for operation PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION (4004)

0 PIN_FLD_POID POID [0] 0.0.0.1 /active_session/telco/gsm/ncc 187153 1
0 PIN_FLD_AUTHORIZATION_ID STR [0] "brmClient-tbalcomb-2013-3-12-6_ne_1462367_0"
Appendix A

Example BCD section of the eserv.config file

This appendix contains an example of the default BCD section of the NCC eserv.config file, which you can find in the /IN/service_packages directory on each of the SLC machines.

```
BCD = {
    bcdActionHandler = {
        serviceDomainInterfaceName = "bcdBeClient"
        clientIdString = "client1"
        loggedNotificationPeriod = 300
        lowCreditBufferTime = 10
        roundingScheme = 3
        cacheTimeout = 60
        maxOutstandingRequests = 1000
        voucherPinLength = 4
        poidPrefix = "/service/telco/"
        accountString = "/account"
    }
    ServiceProfileTagMapping = [
        {
            ServiceKey = -1 # default
            BearerID = -1 # default
            ScalingFactor = 1
            BRMField = "QUANTITY" # or BYTES_UPLINK, BYTES_DOWNLINK
            BRMReqMode = "DURATION" # or VOLUME
            BrmServicePoid = "/service/telco/gsm/telephony"
            BrmObjectType = "gsm"
            UsedUnitsCumulative = false
            DefaultUnitType = "QUANTITY" # or UP_BYTES, DOWN_BYTES
        },
        {
            ServiceKey = 1
            BearerID = 17
            ScalingFactor = 100000 # Bytes per deci-second (=1Mb/second)
            BRMField = "BYTES_UPLINK"
            BRMReqMode = "VOLUME"
            BrmServicePoid = "/service/telco/gsm/data"
            BrmObjectType = "gsm"
            UsedUnitsCumulative = false
            DefaultUnitType = "UP_BYTES"
        }
    ]
    BrmToNccCurrencyMapping = [
        {
            NCCCode = "NZD"
            BRMNum = 554
        },
        {
            NCCCode = "EUR"
            BRMNum = 978
        }
    ]
}
```
BrmBadPinEdrActive = false
BrmEdrObjectType = "/voucher"
BrmBadPinResourceId = 100001
NccInfoFieldNumber = 10000
NccInfoFieldDummyEntry = 10001

NccToBrmFieldMapping = [
    {
        NCCItem = "TARIFF_PLAN_ID"
        BRMType = "INT"
        BRMField = 10004
    },
    {
        NCCItem = "NUMBER_OF_EVENTS_ID"
        BRMType = "INT"
        BRMField = 10002
    },
    {
        NCCItem = "EXAMPLE"
        BRMType = "STRING"
        BRMField = 10005
    },
    {
        NCCItem = "VOUCHER"
        BRMType = "STRING"
        BRMField = 10007
    },
    {
        NCCItem = "PIN"
        BRMType = "STRING"
        BRMField = 10008
    }
]

BrmFieldToEdrMapping = [
    {
        BRMField = 7450
        EDRItem = "AUTHORIZATION_ID"
    },
    {
        BRMField = 3039
        EDRItem = "SESSION_ID"
    }
]

LocationNumberMapping = {
    BRMField = 1251  # Or zero to disable sending of Location Number.
    PrimaryLocationNumberProfileBlock = 18  # Call Context
    PrimaryLocationNumberProfileTag = 327692  # PT_CC_LOCATION_NUMBER. Must be
        # non-zero if BRMField is non-zero.
    SecondaryLocationNumberProfileBlock = 18  # Call Context
    SecondaryLocationNumberProfileTag = 327716  # PT_CC_LOCATION_INFO_LOCATION_NUMBER
        # or zero to disable secondary choice
}

CustomOpCodeMapping = [
    {
        BrmOpCode = 4007
        CustomOpCode = 11007
    },
Appendix A, Example BCD section of the eserv.config file

{  
  BrmOpCode = 4026  
  CustomOpCode = 11026  
}

InSessionNotificationMapping = {  
  ProfileBlock = 17  
  # NCC profile block to populate. Default = 17  
  (TEMPORARY STORAGE)  
  
  Language = {  
    Description = "Preferred Language"  
    # Text to match in  
    PIGGYBACK_NOTIFICATIONS  
    ProfileTag = 37  
    # NCC profile tag to use (PT_LANGUAGE)  
  }

  Channel = {  
    Description = "Preferred Channel"  
    # Text to match in  
    PIGGYBACK_NOTIFICATIONS  
    ProfileTag = 1312050  
    # NCC profile tag to use  
    (PT_ISN_PREF_CHANNEL)  
    InSessionTrigger = [ "Email", "SMS" ]  
    # Which channels require in-session trigger  
  }

  Time = {  
    Description = "Preferred Time"  
    # Text to match in  
    PIGGYBACK_NOTIFICATIONS  
    ProfileTag = 1312051  
    # NCC profile tag to use  
    (PT_ISN_PREF_CHANNEL)  
  }

  CreditThreshold = {  
    Description = "Credit Threshold Breach"  
    # Text to match in  
    PIGGYBACK_NOTIFICATIONS  
    ProfileTag = 1312052  
    # NCC profile tag to use  
    (PT_ISN_CT_BALANCE)  
    BalanceTypeTag = 1312054  
    # NCC profile subtag for balance type ID  
    (PT_ISN_CT_BAL_TYPE)  
    BalanceNameTag = 1312055  
    # NCC profile subtag for balance type name  
    (PT_ISN_CT_BAL_NAME)  
    AmountTag = 1312053  
    # NCC profile subtag for balance amount  
    (PT_ISN_CT_BAL_AMOUNT)  
    CurrentBalanceTag = 1312056  
    # NCC profile subtag for current balance  
    (PT_ISN_CT_BAL_CURRENT_BAL)  
    GroupObjTag = 1312057  
    # NCC profile subtag for group object  
    (PT_ISN_CT_BAL_GROUP_OBJ)  
    PercentTag = 1312058  
    # NCC profile subtag for percent  
    (PT_ISN_CT_BAL_PERCENT)  
    SourceObjTag = 1312059  
    # NCC profile subtag for source object  
    (PT_ISN_CT_BAL_SOURCE_OBJ)  
    AlertTypeTag = 1312060  
    # NCC profile subtag for alert type  
    (PT_ISN_CT_BAL_ALERT_TYPE)  
    ReasonTag = 1312061  
    # NCC profile subtag for reason  
    (PT_ISN_CT_BAL_REASON)  
    CreditFloorTag = 1312062  
    # NCC profile subtag for credit floor  
    (PT_ISN_CT_BAL_CREDIT_FLOOR)  
    CreditLimitTag = 1312063  
    # NCC profile subtag for credit limit  
    (PT_ISN_CT_BAL_CREDIT_LIMIT)  
    CreditThresholdTag = 1312064  
    # NCC profile subtag for percent threshold  
    (PT_ISN_CT_BAL_CREDIT_THRESH)  
    CreditThresholdFixedTag = 1312065  
    # NCC profile subtag for fixed threshold  
    (PT_ISN_CT_BAL_CREDIT_THRESH_FIXED)  
  }
BalanceUnitNameTag = 1312078  # NCC profile subtag for balance unit name (PT_ISN_CT_BAL_UNIT_NAME)
}

SubscriptionExpiry = {
    Description = "Subscription Expired"  # Text to match in PIGGYBACK_NOTIFICATIONS
    ProfileTag = 1312066  # NCC profile tag to use (PT_ISN_SUB_EXPIRY)
    ExpiryDateTag = 1312067  # NCC profile subtag for expiry date (PT_ISN_SUB_EXPIRY_DATE)
}

StreamingThreshold = {
    Description = "Streaming Threshold reached"  # Text to match in PIGGYBACK_NOTIFICATIONS
    ProfileTag = 1312068  # NCC profile tag to use (PT_ISN_STREAM_THRESH)
    CurrentBalanceTag = 1312069  # NCC profile subtag for current balance (PT_ISN_STREAM_THRESH_CURRENT_BAL)
}

Balance = {
    ProfileTag = 1312070  # NCC profile tag to use (PT_ISN_BALANCE)
    BalanceTypeTag = 1312076  # NCC profile subtag for balance type ID (PT_ISN_BALANCE_TYPE)
    BalanceNameTag = 1312077  # NCC profile subtag for balance type name (PT_ISN_BALANCE_NAME)
    AmountTag = 1312071  # NCC profile subtag for amount (PT_ISN_BALANCE_AMOUNT)
    AvailLimitTag = 1312072  # NCC profile subtag for amount (PT_ISN_BALANCE_AVAIL_LIMIT)
    BalanceUnitNameTag = 1312079  # NCC profile subtag for balance unit name (PT_ISN_BALANCE_UNIT_NAME)
}

Status = {
    RatingStatusTag = 1312073  # NCC profile subtag for rating status (PT_ISN_RATING_STATUS)
    LifecycleStateTag = 1312074  # NCC profile subtag for lifecycle state (PT_ISN_LIFECYCLE_STATE)
    FailureReasonTag = 1312075  # NCC profile subtag for failure reason (PT_ISN_FAILURE_REASON)
}

# End of InSessionNotificationMapping
# End of bcdActionHandler

bcdBillingClient = {
    contextOpenTimeoutMilliseconds = 5000
    recoverCmPtrSeconds = 5
    maxPollMillisecond = 1
    maxTries = 3
    latencyStatisticsInterval = 300
    recordCMIPAddressInStats = false
    recordOpcodeInStats = false
    recordPortInStats = false

    OpCodeMapping = [
        { operation = "PCM_OP_BAL_GET_BALANCE" , opCode = 3701 },
        { operation = "PCM_OP_CUST_MODIFY_CUSTOMER" , opCode = 64 },
        { operation = "PCM_OP_PYMT_TOPUP" , opCode = 3726 },
        { operation = "PCM_OP_SEARCH" , opCode = 7 },
        { operation = "PCM_OP_SUBSCRIPTION_PURCHASE_DEAL" , opCode = 108 },
        { operation = "PCM_OP_SUBSCRIPTION_READ_ACCT_PRODUCTS" , opCode = 81 }
    ]
}  /* End of bcdBillingClient */
{operation = "PCM_OP_TCF_AAA_AUTHORIZE" , opCode = 4002 }
{operation = "PCM_OP_TCF_AAA_CANCEL_AUTHORIZATION" , opCode = 4004 }
{operation = "PCM_OP_TCF_AAA_QUERY_BALANCE" , opCode = 4104 }
{operation = "PCM_OP_TCF_AAA_ACCOUNTING" , opCode = 4012 }
{operation = "PCM_OP_TCF_AAA_STOP_ACCOUNTING" , opCode = 4007 }
{operation = "PCM_OP_TCF_AAA_STOP_ACCOUNTING_PREP_INPUT" , opCode = 4031 }
{operation = "PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE" , opCode = 4026 }
{operation = "PCM_OP_TRANS_ABORT" , opCode = 13 }
{operation = "PCM_OP_TRANS_OPEN" , opCode = 12 }
{operation = "PCM_OP_WRITE_FLDS" , opCode = 5 }
{operation = "PCM_OP_READ_FLDS" , opCode = 4 }
{operation = "PCM_OP_ACT_ACTIVITY" , opCode = 151 }
{operation = "PCM_OP_CUST_POL_GET_DEALS" , opCode = 278 }
{operation = "PCM_OP_BILL_DEBIT" , opCode = 105 }
]

ConnectionManager = {
  database = 1
  service = "/service/pcm_client"
  cmPointers = [
    { cmPtr = "ip 192.168.111.111 12010", poolSize = 15 }
    { cmPtr = "ip 192.168.111.112 12010", poolSize = 50 }
    { cmPtr = "ip 192.168.111.111 12011", poolSize = 15 }
    { cmPtr = "ip 192.168.111.112 12011", poolSize = 15 }
  ]
}

defaultOperationTimeout = 250
OperationTimeouts = [
  {operation = "PCM_OP_BAL_GET_BALANCE" , timeoutMilliseconds = 250 }
  {operation = "PCM_OP_CUST_MODIFY_CUSTOMER" , timeoutMilliseconds = 250 }
  {operation = "PCM_OP_PYMT_TOPUP" , timeoutMilliseconds = 250 }
  {operation = "PCM_OP_SEARCH" , timeoutMilliseconds = 250 }
  {operation = "PCM_OP_SUBSCRIPTION_PURCHASE_DEAL" , timeoutMilliseconds = 250 }
  {operation = "PCM_OP_TCF_AAA_AUTHORIZE" , timeoutMilliseconds = 100 }
  {operation = "PCM_OP_TCF_AAA_QUERY_BALANCE" , timeoutMilliseconds = 20 }
  {operation = "PCM_OP_TCF_AAA_STOP_ACCOUNTING" , timeoutMilliseconds = 300 }
  {operation = "PCM_OP_TCF_AAA_UPDATE_AND_REAUTHORIZE" , timeoutMilliseconds = 250 }
  {operation = "PCM_OP_TCF_AAA_STOP_ACCOUNTING_PREP_INPUT" , timeoutMilliseconds = 250 }
  {operation = "PCM_OP_TRANS_ABORT" , timeoutMilliseconds = 250 }
  {operation = "PCM_OP_TRANS_OPEN" , timeoutMilliseconds = 250 }
  {operation = "PCM_OP_WRITE_FLDS" , timeoutMilliseconds = 250 }
  {operation = "PCM_OP_READ_FLDS" , timeoutMilliseconds = 250 }
]
NCC Glossary of Terms

AAA

ACS
Advanced Control Services configuration platform.

API
Application Programming Interface

BE
Billing Engine

BFT
Billing Failure Treatment - the process that is applied if the system has lost all connections to a billing engine. It allows for limited continuation of call processing functions, if configured.

CAMEL
Customized Applications for Mobile network Enhanced Logic
This is a 3GPP (Third Generation Partnership Project) initiative to extend traditional IN services found in fixed networks into mobile networks. The architecture is similar to that of traditional IN, in that the control functions and switching functions are remote. Unlike the fixed IN environment, in mobile networks the subscriber may roam into another PLMN (Public Land Mobile Network), consequently the controlling function must interact with a switching function in a foreign network. CAMEL specifies the agreed information flows that may be passed between these networks.

CC
Country Code. Prefix identifying the country for a numeric international address.

CCA
Credit-Control-Answer, used in Diameter by the credit-control server to acknowledge a Credit-Control-Request (CCR) from the credit-control client.

CCR
Credit-Control-Request, used in Diameter by the credit-control client to request credit authorization from the credit-control server.

CCS
1) Charging Control Services (or Prepaid Charging) component.
2) Common Channel Signalling. A signalling system used in telephone networks that separates signalling information from user data.
CDR
Call Data Record

Note: The industry standard for CDR is EDR (Event Detail Record). Over time EDR will replace CDR in the Oracle documentation.

Connection
Transport level link between two peers, providing for multiple sessions.

CORBA
Common Object Request Broker Architecture. It is a framework that provides interoperability between objects built in different programming languages, running on different physical machines perhaps on different networks. It specifies an Interface Definition Language, and API that allows client / server interaction with the ORB.

CPU
Central Processing Unit

Diameter
A feature rich AAA protocol. Utilises SCTP and TCP transports.

DP
Detection Point

DTMF
Dual Tone Multi-Frequency - system used by touch tone telephones where one high and one low frequency, or tone, is assigned to each touch tone button on the phone.

EDR
Event Detail Record

Note: Previously CDR. The industry standard for CDR is EDR (Event Detail Record). Over time EDR will replace CDR in the NCC documentation.

ENUM
E.164 Number Mapping.

FDA
First Delivery Attempt - the delivery of a short message directly to the SME rather than relaying it through the MC.

FOX
Fast OSA eXtensions. A TCP/IP billing protocol intended for use with external vendors. Based on OSA, it fills in functional gaps missing in OSA, and defines "combined" OSA operations to increase platform throughput. Uses a non-CORBA transport layer in order to provide enhanced fail-over and connection redundancy.
GPRS
General Packet Radio Service - employed to connect mobile cellular users to PDN (Public Data Network- for example the Internet).

GSM
Global System for Mobile communication.
It is a second generation cellular telecommunication system. Unlike first generation systems, GSM is digital and thus introduced greater enhancements such as security, capacity, quality and the ability to support integrated services.

HLR
The Home Location Register is a database within the HPLMN (Home Public Land Mobile Network). It provides routing information for MT calls and SMS. It is also responsible for the maintenance of user subscription information. This is distributed to the relevant VLR, or SGSN (Serving GPRS Support Node) through the attach process and mobility management procedures such as Location Area and Routing Area updates.

HPLMN
Home PLMN

HTML
HyperText Markup Language, a small application of SGML used on the World Wide Web.
It defines a very simple class of report-style documents, with section headings, paragraphs, lists, tables, and illustrations, with a few informational and presentational items, and some hypertext and multimedia.

IMSI
International Mobile Subscriber Identifier. A unique identifier allocated to each mobile subscriber in a GSM and UMTS network. It consists of a MCC (Mobile Country Code), a MNC (Mobile Network Code) and a MSIN (Mobile Station Identification Number).
The IMSI is returned by the HLR query (SRI-SM) when doing FDA. This tells the MSC exactly who the subscriber is that the message is to be sent to.

IN
Intelligent Network

INAP
Intelligent Network Application Part - a protocol offering real time communication between IN elements.

IP
1) Internet Protocol
2) Intelligent Peripheral - This is a node in an Intelligent Network containing a Specialized Resource Function (SRF).

IP address
Internet Protocol Address - network address of a card on a computer
ISDN
Integrated Services Digital Network - set of protocols for connecting ISDN stations.

ITU
International Telecommunication Union

IVR
Interactive Voice Response - systems that provide information in the form of recorded messages over telephone lines in response to user input in the form of spoken words or, more commonly, DTMF signalling.

MAP
Mobile Application Part - a protocol which enables real time communication between nodes in a mobile cellular network. A typical usage of the protocol would be for the transfer of location information from the VLR to the HLR.

MC
Message Centre. Also known as SMSC.

MCC
Mobile Country Code. In the location information context, this is padded to three digits with leading zeros. Refer to ITU E.212 ("Land Mobile Numbering Plan") documentation for a list of codes.

Messaging Manager
The Messaging Manager service and the Short Message Service components of Oracle Communications Network Charging and Control product. Component acronym is MM (formerly MMX).

MIN
Mobile Identification Number, also known as an MSID.

MM
Messaging Manager. Formerly MMX, see also XMS (on page 226) and Messaging Manager (on page 222).

MNC
Mobile Network Code. The part of an international address following the mobile country code (MCC), or at the start of a national format address. This specifies the mobile network code, that is, the operator owning the address. In the location information context, this is padded to two digits with a leading zero. Refer to ITU E.212 ("Land Mobile Numbering Plan") documentation for a list of codes.

MO
Mobile Originated

MOC
Managed Object Class
MS
Mobile Station

MSC
Mobile Switching Centre. Also known as a switch.

MSID
Mobile Subscriber Identification, also known as an MIN.

MSIN
Mobile Station Identification Number.

MSISDN
Mobile Station ISDN number. Uniquely defines the mobile station as an ISDN terminal. It consists of three parts; the country code (CC), the national destination code (NDC) and the subscriber number (SN).

MT
Mobile Terminated

MTC
Mobile Terminated Call. The part of the call associated with a subscriber receiving an inbound call.

Oracle
Oracle Corporation

ORB
Object Request Broker. Within an Object based communication system, an ORB keeps track of the actual addresses of all defined objects and thus is used to route traffic to the correct destination. The CORBA defines the ORB in a series of standards enabling different platforms to share common information.

OSA
Open Service Access provides a standard interface through which developers can design services that may interact with functions within the network.

PI
Provisioning Interface - used for bulk database updates/configuration instead of GUI based configuration.

PIN
Personal Identification Number
PLMN
Public Land Mobile Network

SCP
Service Control Point. Also known as SLC.

SCTP
Stream Control Transmission Protocol. A transport-layer protocol analogous to the TCP or User Datagram Protocol (UDP). SCTP provides some similar services as TCP (reliable, in-sequence transport of messages with congestion control) but adds high availability.

Session
Diameter exchange relating to a particular user or subscriber access to a provided service (for example, a telephone call).

SGML

SGSN
Serving GPRS Support Node

SIM
Usually referred to as a SIM card, the Subscriber Identity Module is the user subscription to the mobile network. The SIM contains relevant information that enables access onto the subscribed operator’s network.

SIP
Session Initiation Protocol - a signaling protocol for Internet conferencing, telephony, event notification and instant messaging. (IETF)

SLC
Service Logic Controller (formerly UAS).

SLEE
Service Logic Execution Environment

SME
Short Message Entity - an entity which may send or receive Short Messages. It may be located in a fixed network, a mobile, or an SMSC.

SMS
Depending on context, can be:
- Short Message Service
- Service Management System platform
- NCC Service Management System application

**SN**
Service Number

**SRF**
Specialized Resource Function - This is a node on an IN which can connect to both the SSP and the SLC and delivers additional special resources into the call, mostly related to voice data, for example play voice announcements or collect DTMF tones from the user. Can be present on an SSP or an Intelligent Peripheral (IP).

**SRI**
Send Routing Information - This process is used on a GSM network to interrogate the HLR for subscriber routing information.

**SSP**
Service Switching Point

**STR**
Session message: Session-Termination Request

**System Administrator**
The person(s) responsible for the overall set-up and maintenance of the IN.

**TCP**
Transmission Control Protocol. This is a reliable octet streaming protocol used by the majority of applications on the Internet. It provides a connection-oriented, full-duplex, point to point service between hosts.

**Telco**
Telecommunications Provider. This is the company that provides the telephone service to customers.

**Telecommunications Provider**
See Telco.

**USSD**
Unstructured Supplementary Service Data - a feature in the GSM MAP protocol that can be used to provide subscriber functions such as Balance Query and Friends and Family Access.

**VLR**
Visitor Location Register - contains all subscriber data required for call handling and mobility management for mobile subscribers currently located in the area controlled by the VLR.
Voice Call

The term “voice call” in this document is intended to denote any call controlled by CAMEL or INAP InitialDP. In practice this also includes fax calls, data-over-voice calls, and also includes 3G voice and video conference calls.

VWS

Oracle Voucher and Wallet Server (formerly UBE).

XMS

Three letter code used to designate some components and path locations used by the Oracle Communications Network Charging and Control Messaging Manager (on page 222) service and the Short Message Service. The published code is MM (on page 222) (formerly MMX).
Index

A
A Voice Call Charged Against BRM • 65
A Voice Call When Funds Expire • 72
AAA • 219
About adding custom fields • 20, 43
About BRM • 1
About Configuring BRM • 43
About configuring NCC for the BRM Charging
Driver • 13
About Creating Balance Type Mappings • 13,
19, 38
About Creating Products and Deals • 51
About creating products and deals for direct
named events • 52
About Editing eserv.config Parameters • 15
About Installing the BRM Charging Driver • 11
About integrating NCC and BRM • 1
About PIN_FLD_DIRECTION • 4
About Statistics and Reports • 63
About the BRM Charging Driver • 1
About the session ID • 4
About This Document • v
About Usage Scenarios • 65
About Using In-Session Notifications • 57
accountString • 25
ACS • 219
Adding Custom Fields • 43
Adding Storable Classes • 45
All BCD clients busy • 100
All BCD clients busy flow • 100
All BCD clients busy scenario • 101
API • 219
Audience • v

B
Balances • 24
bcdActionHandler Content • 16
bcdBillingClient content • 29
BE • 219
BearerID • 17
BFT • 219
BRM and NCC components • 2
BRM Charging Driver features • 6
BRM Charging Driver reports • 5
BRM Integration Summary • 9
BrmBadPinEdrActive • 25
BrmBadPinResourceld • 25
BrmEdrObjectType • 26
BRMField • 18
BrmObjectType • 18
BRMReqMode • 18
BrmServicePoid • 18

C
cacheTimeout • 26
Called subscriber busy • 114
Called subscriber busy flow • 114
 Called subscriber busy scenario • 114
Caller hangs up • 66
Caller hangs up flow • 66
Caller hangs up scenario • 66, 73
CAMEL • 219
CC • 219
CCA • 219
CCR • 219
CCS • 219
CDR • 220
Channel • 24
Charging for a Data Session • 96
clientIDString • 26
Configuring additional bcdActionHandler
Parameters • 16, 25
Configuring additional bcdBillingClient
parameters • 29, 31
Configuring bcdActionHandler • 16
Configuring bcdBillingClient • 29
Configuring BRM connections • 29, 30
Configuring BRM for ExtendTimeReservation •
48
Configuring BRM for named event reservations
and reservable direct named events • 53
Configuring BRM for the BRM Charging Driver •
9, 43
Configuring BRM for vouchers • 54
Configuring Calls, Events, and Vouchers • 34
Configuring Credit Threshold Notifications • 59
Configuring In-Session Notifications • 58
Configuring named event refunds • 35, 53
Configuring NCC for BRM vouchers • 36
Configuring NCC for data calls • 34
Configuring NCC for named event reservations
and reservable direct named events • 35
Configuring NCC for named events • 35
Configuring NCC for the BRM Charging Driver •
9, 13
Configuring NCC for voice calls • 34
Configuring non-reservable direct named
events • 54
Configuring Replication • 14
Configuring the SLEE to run the scripts • 28, 37
Configuring the Subscriber in the Customer
Center • 58
Connection • 220
customOpenTimeOutMilliseconds • 31
Copying PCM input flist fields to an ACS EDR •
16, 20
Copyright • ii
CORBA • 220
CPU • 220
Creating a customer • 54
Creating a Customer • 54
Creating a Product and Deal for Data Calls • 52, 95
Creating a Product and Deal for Named Events • 52
Creating a Product and Deal for Voice Calls • 51, 65
Creating additional custom fields • 45
Creating Balance Type Mappings • 38
Creating BRM Vouchers • 36, 54, 81
Creating custom classes • 45
Creating Header and Library Files for the Custom Classes • 46
Creating Products and Deals • 9, 51, 55
Creating the BRM Domain • 14
Creating the BRM product and deal for data calls • 52
Creating the BRM product and deal for voice calls • 51
Creating the custom header and library files • 46
CreditThreshold • 24

D
Data Charging • 95
Data session charging • 96
Data session charging flow • 96
defaultOperationTimeout • 31
DefaultUnitType • 19
Diameter • 220
Document Conventions • vi
DP • 220
DTMF • 220

E
Editing eserv.config parameters for the BRM Charging Driver • 15
EDR • 220
Enabling In-Session Notifications • 57
Enabling refunds for direct named events • 35, 53
ENUM • 220
Error on IVR recharge account topup failed • 171
Error on IVR recharge account topup failed flow • 171
Error on IVR recharge account topup failed scenario • 172
Error on IVR Recharge using a VWS voucher with invalid PIN • 168
Error on IVR Recharge using a VWS voucher with invalid PIN flow • 168
Error on IVR recharge using a VWS voucher with invalid PIN scenario • 168
Error on SMS recharge with account topup failed • 178
Error on SMS recharge with account topup failed flow • 178
Error on SMS recharge with account topup failed scenario • 178
Error on SMS recharge with invalid PIN • 174
Error on SMS recharge with invalid PIN flow • 174
Error on SMS recharge with invalid PIN scenario • 175
Error on USSD recharge with account topup failed • 185
Error on USSD recharge with account topup failed flow • 185
Error on USSD recharge with account topup failed scenario • 186
Error on USSD recharge with invalid PIN • 182
Error on USSD recharge with invalid PIN flow • 182
Error on USSD recharge with invalid Pin scenario • 182
Example BCD section of the eserv.config file • 213

F
FDA • 220
FOX • 220
Funds expire • 71
Funds expire flow • 71

G
Generating reports • 63
Generating statistics • 63
Generating Statistics and Reports • 5, 10, 63
Generating the Custom JAR File • 46
GPRS • 221
GSM • 221

H
HLR • 221
HPLMN • 221
HTML • 221

I
IMSI • 221
IN • 221
INAP • 221
Installing the BRM Charging Driver • 9, 11
Integrating NCC and BRM • 9
IP • 221
IP address • 221
ISDN • 222
ITU • 222
IVR • 222
IVR BRM voucher recharge • 82
IVR BRM voucher recharge flow • 82
IVR BRM voucher recharge scenario • 82
IVR credit transfer when recipient’s account not found • 208
IVR credit transfer when recipient’s account not found flow • 208
IVR credit transfer when recipient’s account not found scenario • 208
IVR recharge with invalid PIN • 152
IVR recharge with invalid PIN flow • 152
IVR recharge with invalid PIN scenario • 152
IVR VWS multi-balance recharge flow • 93
IVR VWS multi-balance recharge scenario • 94
IVR VWS voucher recharge • 89
IVR VWS voucher recharge flow • 89
IVR VWS voucher recharge scenario • 89

L
Language • 23
latencyStatisticsInterval • 31
Loading the Subscriber Profile • 58
loggedNotificationPeriod • 26
lowCreditBufferTime • 27

M
MAP • 222
Mapping BRM Piggyback Notifications to NCC Profile Tags • 16, 22, 57
Mapping NCC currency codes to BRM values • 16, 19
Mapping NCC information to BRM fields • 16, 19, 43
Mapping NCC sessions to BRM services • 16, 34, 48
Mapping opcodes • 29
Mapping the location number • 16, 21, 44
maxContextIdleTimeSeconds • 32
maxOutstandingRequests • 32
maxPollMilliseconds • 32
maxSelectMicroseconds • 32
maxTries • 32
MC • 222
MCC • 222
Messages
called subscriber busy • 115
caller hangs up • 68
data session • 98
Error on IVR recharge account topup failed • 173
error on IVR recharge using a VWS voucher with an invalid PIN • 169
error on SMS recharge with account topup failed • 180
error on SMS recharge with invalid PIN • 176
error on USSD recharge with account topup failed • 187
error on USSD recharge with invalid PIN • 183
funds expire • 74
IVR credit transfer when recipient’s account not found • 211
IVR recharge with invalid PIN • 153
mid-call tariff change • 107
PCM operation timeout • 104
post call charging of voice call • 190
redemption of BRM voucher • 83
redemption of VWS voucher against BRM account • 91
redemption of VWS voucher using bad PIN • 92
service subscription - not allowed to purchase deal • 198
SMS call info and SMS account balances • 132
SMS charging with direct named event - insufficient funds • 145
SMS charging with named event reservation • 140
SMS charging with named event reservation fail • 143
successful credit transfer using IVR scenario • 205
successful service subscription • 193
successful SMS recharge using a VWS voucher • 162
successful USSD recharge • 148
successful USSD recharge using a VWS voucher • 166
top up with invalid voucher number or PIN • 188
USSD recharge with invalid PIN • 157
with ASB voice call costs and cumulative balance • 121
Messaging Manager • 222, 226
Mid-call tariff change • 105
Mid-call tariff change flow • 105
Mid-call tariff change scenario • 106
MIN • 222
MM • 222, 226
MNC • 222
MO • 222
MOC • 222
Modifying the BCD Client Startup Script • 36
Modifying the BRM Configuration Files • 47
MS • 223
MSC • 223
MSID • 223
MSIN • 223
MSISDN • 223
MT • 223
MTC • 223

N
NccInfoFieldDummyEntry • 27
NccInfoFieldName • 27
No free connections to BRM • 102
No free connections to BRM flow • 102
No free connections to BRM scenario • 102
No NCC subscriber, default control plan • 111
No NCC subscriber, default control plan flow • 111
No NCC subscriber, default control plan scenario • 112

O
Oracle • 223
ORB • 223
OSA • 223
Other NCC components • 5
Other Scenarios • 100
Overview • 1
Overview of Installing BRM Charging Driver • 11
Overview of Installing the BRM Charging Driver • 11
Overview of Statistics and Reports • 63
Overview of the BRM Charging Driver • 1

P
PCM operation timeout • 103
PCM operation timeout flow • 103
PCM operation timeout scenario • 104
PI • 223
PIN • 223
PIN_FLD_LOCATION • 44
PIN_FLD_NCC_FIELD • 44
PIN_FLD_NCC_INFO • 44
PIN_FLD_NCC_NUMBER_OF_EVENTS • 45
PIN_FLD_NCC_TARIFF_PLAN_ID • 45
PLMN • 224
poidPrefix • 27
Post call charging of voice call • 189
Post call charging of voice call flow • 189
Post call charging of voice call scenario • 189
Preconditions for data session charging on BRM • 95
Preconditions for recharge using BRM vouchers • 81
Preconditions for recharge using VWS vouchers • 88
Preconditions for voice call scenarios • 65
Prerequisite • 57
Prerequisites • v
Procedure to create a BRM domain • 14, 36, 54
Procedure to modify the script • 36

R
Recharge using BRM Vouchers • 81
Recharge using VWS Vouchers • 88
recordCMIPAddressInStats • 33
recordOpcodeInStats • 33
recordPortInStats • 33
recoverCmPtrSeconds • 33

Related Documents • v
Replication for the BRM Charging Driver tables • 14, 63
Restarting BRM • 58
roundingScheme • 28

S
ScalingFactor • 17
Scope • v
SCP • 224
SCTP • 224
Service subscription - not allowed to purchase deal • 197
Service subscription - not allowed to purchase deal flow • 197
Service subscription - not allowed to purchase deal scenario • 197
serviceDomainInterfaceName • 28
ServiceKey • 17
Session • 224
Setting the PiggyBack parameter • 57
Setting the SubscriberLifeCycle parameter • 58
SGML • 224
SGSN • 224
SIM • 224
SIP • 224
SLC • 224
SLEE • 224
SME • 224
SMS • 224
SMS BRM voucher recharge - invalid PIN • 87
SMS BRM voucher recharge - invalid PIN flow • 87
SMS BRM voucher recharge - invalid PIN scenario • 87
SMS call delivered successfully • 78
SMS call delivered successfully flow • 78, 80
SMS call delivered successfully scenario • 78
SMS call info and SMS account balances • 128
SMS call info and SMS account balances flow • 128
SMS call info and SMS account balances scenario • 130
SMS call that fails permanently • 80
SMS call that fails permanently flow • 80
SMS call that fails permanently scenario • 80
SMS charging with direct named event - insufficient funds • 144
SMS charging with direct named event - insufficient funds flow • 144
SMS charging with direct named event - insufficient funds scenario • 144
SMS charging with named event reservation • 138
SMS charging with named event reservation fail • 141