

Oracle® Communications Convergent Charging Controller Configuration User's Guide



Release 15.2

G47589-01

January 2026

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 2026, Oracle and/or its affiliates.

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

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

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

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

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

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

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

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

Contents

1 Introduction to Convergent Charging Controller Configuration

Service Management System	1
Accessing SMS	1
Logging on to SMS	1
Adding the SMS User	2
Assigning Services to User	4
Create a Control Plan	6
Creating a Basic Control Plan	6
Allocating Additional Feature Nodes	7

2 Service Provider

Service Provider Creation	1
Allocating CCS Permissions	1
Creating Service Provider	2
Self Managed Customer	5
Create or Allocate Data Sets to your Service Provider	7
ACS Configuration	8
Geography Set	8
Holiday Set	9
Announcement Set	10
Feature Node Set	12
Service Provider Limits	15
Setting Limits	15

3 Product Type

Product Types	1
Creating Product Types	1

4 Subscribers

Subscriber Creation	1
Credit Type	1

Subscribers and MSISDNs	1
Subscriber Experience of Making a Call	2
Viewing Existing Subscribers	2
Creating a Subscriber Account	2
Wallet	3
Block a Subscriber	3
Unblock a Subscriber	3

5 Rating and Tariffs

Tariff Creation	1
Rating	1
Components	1
Tariff Plans	1
Tariff Configuration	2
Creating a Tariff	2
Tariff Plan	3
Rate Table	4
Tariff Plan Selector	5
Balance Type Cascade	5
Charge Period	6
Discount Set	7
Discount Period	8
CLI-DN	9
Weekly Tariff	10
Holiday Tariff	11
MFile	12

6 Adding a Service

Service Creation	1
Process	1
Triggering on Service Key	1
IDP Fields	2
Service Key	4
Service	5
Service Entry	5
CCS Capability	5

7 Tools

Basic SLPIT	1
-------------	---

Basic SLPIT Script	1
Analyze a Basic Script	3
Description of the Script acs_basic.slp	4
SLPIT Script for UATB Node	6
Provisioning Interface	6
Component Diagram	7
Process Descriptions	7
Configuring User Access to PI Screens	8
Configuring User and PC Connection to PI	9
Testing User and PC Connection	11
ACS Commands	13
CCS Commands	14

8 Integration with ECE

Control Plan Configuration	1
Configuring the Set Active Domain Feature Node	1
Configuring the ECE Service Identifier and Rating Group	1
About Variable Amount Recharge Feature Node Configuration	2
About CCS Capabilities	3
About Convergent Charging Controller Balance-Type Configuration	3
Enabling Balance Values to Pass Between Convergent Charging Controller and ECE	3
Balance-Type Configuration in eserv.config	3
Subscriber-Domain-Type Configuration in eserv.config	4
Balance-Query Configuration in eserv.config	5
Configuring Balance-Type Mapping Between Convergent Charging Controller and ECE	5
Configuring Balance-Type Announcements	6
Configuring Balance-Type Translations for Balance Notifications	7
Balance Type Translations Fields	7
Enabling First-Use Balance-Expiry Policies in Convergent Charging Controller	9
About DCD AVP Configuration	9
Configuring DCD AVPs to Integrate Convergent Charging Controller with ECE	9
About Sign Inversion	10
Excluding AVPs with Zero Values in Message Requests	10
Currency Code Configuration for Playing Announcements	11
Mapping Convergent Charging Controller Voucher-Expiry Periods to ECE Values	11
Voucher Top-Up Configuration for Multiple Balance Types	12
Balance Query Response Configuration for Multiple Balances	12
Preventing Balance Queries from Returning Invalid Values	13
Configuring AVPs to Support First Use	13
New Bucket AVP Configuration	13

About This Content

The scope of this document includes all information required to give the reader a complete view on how to configure services to ready a freshly installed Convergent Charging Controller platform for customer specific configuration.

Audience

This guide is written primarily for administrators operating the Convergent Charging Controller platform. However, the overview sections of the document are useful to anyone requiring an introduction.

A solid understanding of UNIX and a familiarity with IN concepts are an essential prerequisite for safely using the information contained in this technical guide. Attempting to install, remove, configure or otherwise alter the described system without the appropriate background skills, could cause damage to the system; including temporary or permanent incorrect operation, loss of service, and may render your system beyond recovery.

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.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Resources

See these Oracle resources:

- Installation Guide
- System Administrator's Guide

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction to Convergent Charging Controller Configuration

This guide is written to give you a complete view on how to configure services on a freshly installed Oracle Communications Convergent Charging Controller platform.

It describes the usage and configuration of the different components, and gives a presentation on the service templates delivered with the product. It also provides a walk-through for a custom service creation.

Service Management System

Accessing SMS

Follow these steps to launch Service Management System using Java.

For more information, see *Setting up the Screens* in *Service Management System Technical Guide*.

To launch the application:

1. Ensure the Java SE Runtime Environment version 21 is installed on your computer.
2. If required, obtain and install the trusted certificate for the database connection into your keystore.
3. Obtain the application zip file containing JARs and other files (`smsGui.bat` or `smsGui.sh`).
4. Start the application:
 - **For Windows:** Run `smsGui.bat` to start the application.
 - **For other systems:**
 - a. Change the permission of `smsGui.sh` using the command:

```
chmod 755 smsGui.sh
```

- b. Run the application using the command:

```
bash smsGui.sh
```

Logging on to SMS

To log in to the SMS application, follow these steps:

1. On the SMS Login screen:



Refer to the displayed login screen (see image for guidance).

2. Enter the account details for the default super-user:
 - **User name:** `su`
 - **Password:** Usually `ssob` on freshly installed servers.

Note

Please take care entering these account details as three incorrect attempts will lock out all other users. You will use your own user account as soon as you have created one. If you manage to lock the `su` account, please see your trainer or technical mentor to show you how to unlock it.

Adding the SMS User

The system is deployed with an already existing user called `su`. It is best to create a specific user, even with the same rights in order to keep `su` as a backup in case the other login becomes corrupted.

To create a new user and set up access in SMS:

1. Log in to the SMS server as `su`. See *Accessing SMS*.
2. Select **Operator Functions** and then **User Management**.

Result: The **User Management** screen, **User** tab appears.

3. Enter your details.

Note

The **User Name** should be in uppercase. You can click **Help** for a description of the fields.

4. Click **Set Password**.

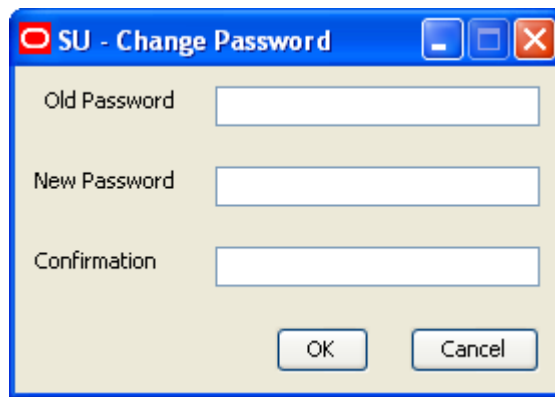
Result: The **Set SMS User Password** screen appears.

5. Set the user password.

Note

Please do not change passwords for any users not created by you.

6. Click **OK**.
7. Remember to Save and then Close.
8. Close the browser to log out of SMS.
When you log out of SMS, you will find you can no longer log in as another user without restarting the browser and re-accessing SMS.
9. Now log in to SMS using your new user.
10. If you set the lifetime of your password to 0 days, or if the lifetime has expired, you will immediately be prompted to change your password.



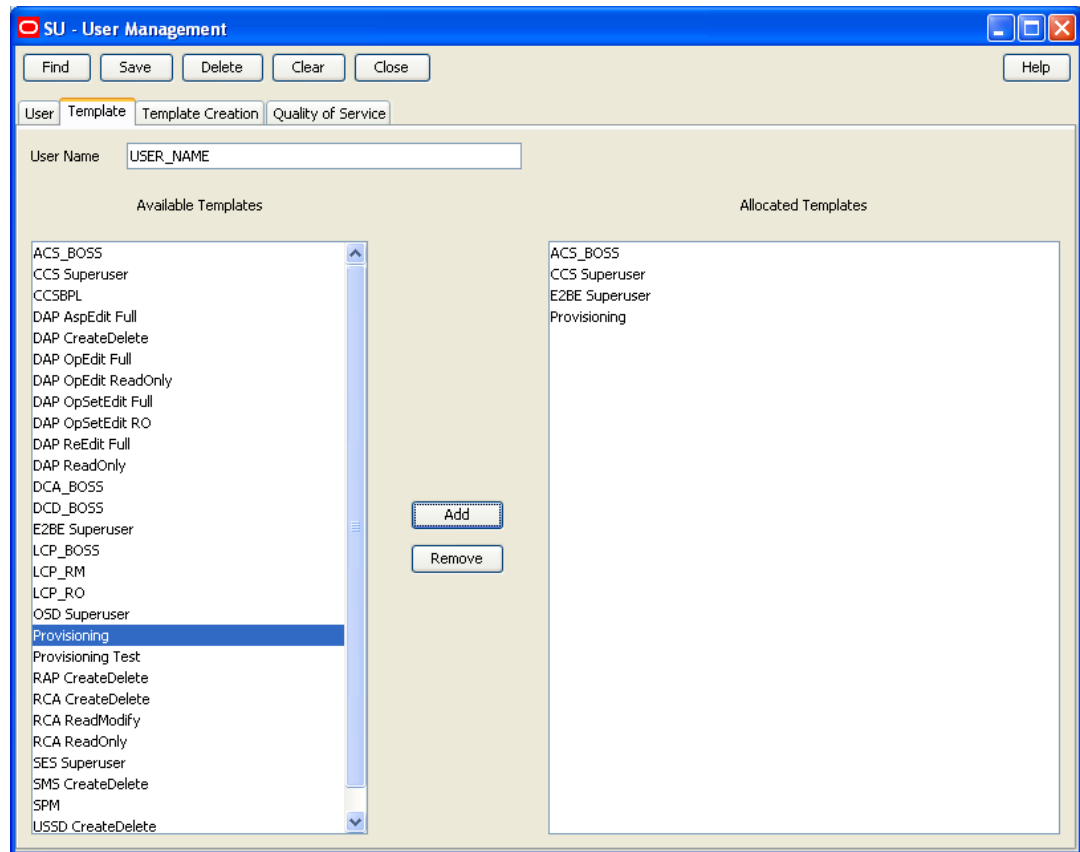
Change the password, if required, and click **OK**.

11. When your new SMS user first logs in, no services will have been allocated to them. On the SMS Main screen, select the Services menu to confirm this. All new users are initially created like this by default.

Assigning Services to User

To assign services to your user:

1. First log out of SMS, then log in again as the super user (su). Your user does not have the right to assign services.
2. Select **Operator Functions**, then **User Management**, and then click **Find**.
Result: You see the **SU – Find User** screen.
3. Type your user's name (in whole or part) into the **User Name** field and click **Search**.
Result: The **Search** screen will display all users matching your search terms.
4. Select your user by clicking the relevant row in the search results, then click **Close**.
Result: The **User** tab is now populated with the target user's ID, and all operations accessible via the tabs across the top of the screen will now relate to that user.
5. Select the **Template** tab to access Template related options for your user.



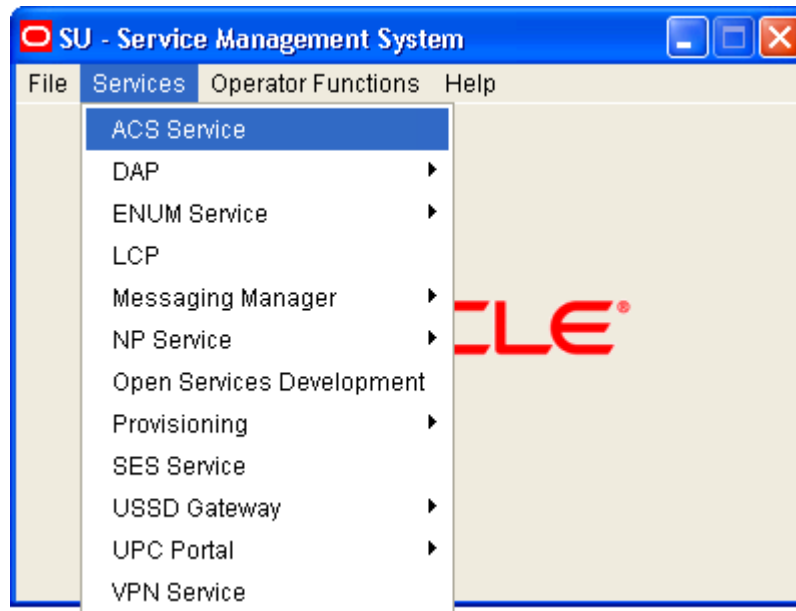
6. You allocate templates to the user to give them the ability to manage component services. To allocate templates, select a template from the **Available Templates** list and click **Add**, repeating as required.

Note

The contents of the **Available Templates** list will vary, depending on which packages have been installed.

7. Click **Save** and then **Close**.
8. Log out of SMS then log in again using your newly created SMS user.

Result: This time you will find you have the service allocated to the template, for example, for ACS_BOSS, the ACS Service is available.



Create a Control Plan

Control plans define how calls are processed. The functionality for creating control plans is available in the Advanced Control Services Control Plan Editor (ACS CPE) accessible from the ACS UI. It includes functionality that enables you to:

- Create control plans.
- Import and export control plans.

A control plan consists of a number of feature nodes connected together. These connections define the possible routes for a call. Each feature node belongs to a specific feature node group in the control plan palette.

When you create a new user, such as the SMS user, only the Base group, containing just the Start and End feature nodes, is available to them. So that your customer can create the control plans they need, you should ensure they have access to the full feature set within the ACS Control Plan Editor. When your customer has access to additional feature sets, they will be able to access additional palettes in the ACS Control Plan Editor.

For more information about using the ACS Control Plan Editor, see *CPE User's Guide*.

Creating a Basic Control Plan

To create a basic control plan that contains only the Start and End feature nodes:

1. On the **Service Management System** screen, select **ACS Service** from the **Services** menu.
The **Advanced Control Services** window appears.
2. Click **Control Plans** to open the CPE.
3. From the File menu in the **Control Plan Editor** window, select **New**.
The **Start** feature node is added to the control plan.
4. From the Base feature node group in the CPE palette, click and drag the End feature node so that it is below the Start feature node in the Control Plan Editor window.

5. Connect the Start feature node to the End feature node by clicking the connection point on the Start feature node and dragging a connecting line to the End feature node.
6. Save the control plan by selecting **Save** from the **File** menu and entering a name in the **Plan Structure** field. When you save the control plan it will be compiled at the same time.

Note

If you experience problems saving your control plan and you logged in through `acsGui.bat/acsGui.sh`, please log out, then access the CPE by selecting **Service Management System**, then **Services**, and then **ACS Service**.

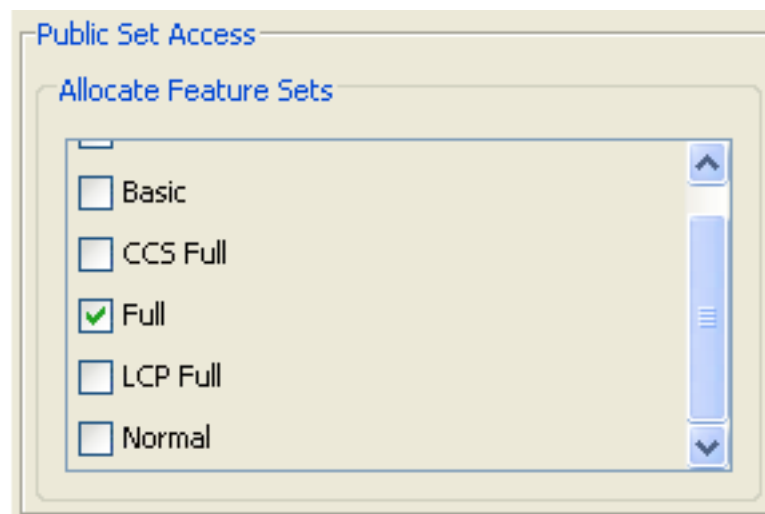
Allocating Additional Feature Nodes

To allocate additional sets of feature nodes to the ACS Control Plan Editor for your customer:

Note

Only the Start and End feature nodes are available by default.

1. Log in to ACS through one of the following:
 - SMS as a SMS administrator (privilege 6 or 7).
 - ACS directly with a user account that has privilege level 6.
2. In the ACS main window, click **Customer**.
3. In the ACS Customer window, select your customer from the **Customer** list, then select the **Resource Limits** tab.
4. Select your customer in the table and click **Edit**.
5. On the Edit Customer Resources Limits window, select the **Full** check box in the **Allocate Feature Sets** panel.



6. Click **Save**.
7. Click **Close** to close the Customer window, and then log out of SMS or ACS.

8. Log in to your customer's user account and re-open the CPE to see the extra feature nodes.

2

Service Provider

This chapter explains how to configure a service provider.

Service Provider Creation

A service provider provides services to your subscribers. You must have at least one service provider. Each service provider contains the specific configuration pertaining to them. For example, service provider configuration includes configuration for charging and accounts.

Each service provider can have their own set of product types. Each product type defines the available services to subscribers with that product type, and includes product-specific configuration for:

- Subscriber accounts
- Subscriber wallets
- Tariff plans
- Control plans

You can build new services against existing product types that can be unique to a specific service provider or subscriber.

When a call is made that relates to the Prepaid Charging service, the Service Loader performs a number of checks to determine which control plan to use. It looks up the following information:

- The personal wallet of the subscriber's account.
- The product type associated with the subscriber's personal wallet.
- The capability in that product type that matches the Service Logic Execution Environment (SLEE) service key.

The control plan is then matched to the product type capability.

Allocating CCS Permissions

Before you can create a service provider, you should allocate CCS permissions for the SMS user.

To allocate CCS permissions for the SMS user:

1. Log in to the SMS UI as the su user.
2. Create an SMS user, if you have not created one already. For more information, see *Adding the SMS user*.
3. Allocate the following templates to your SMS user to give them CCS permissions:
 - CCS_Superuser
 - ACS_BOSS

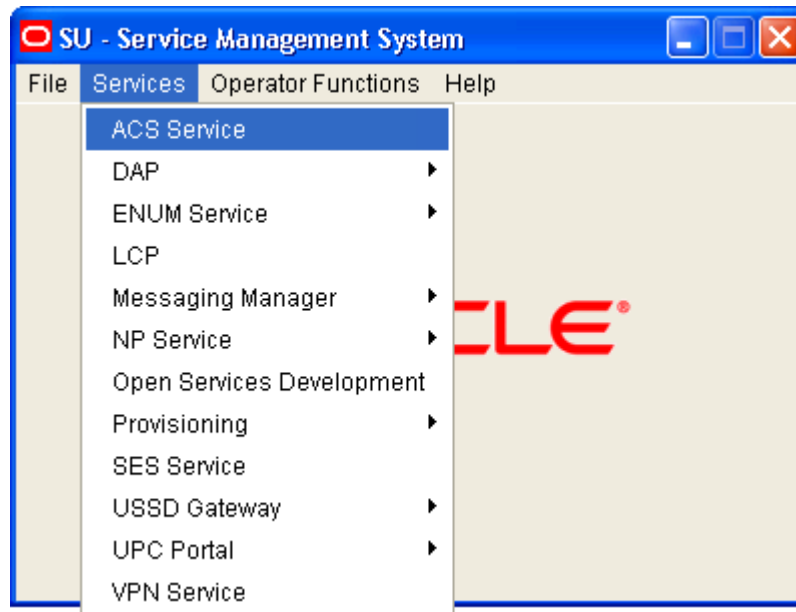
For more information, see *Assigning Services to user*.

Creating Service Provider

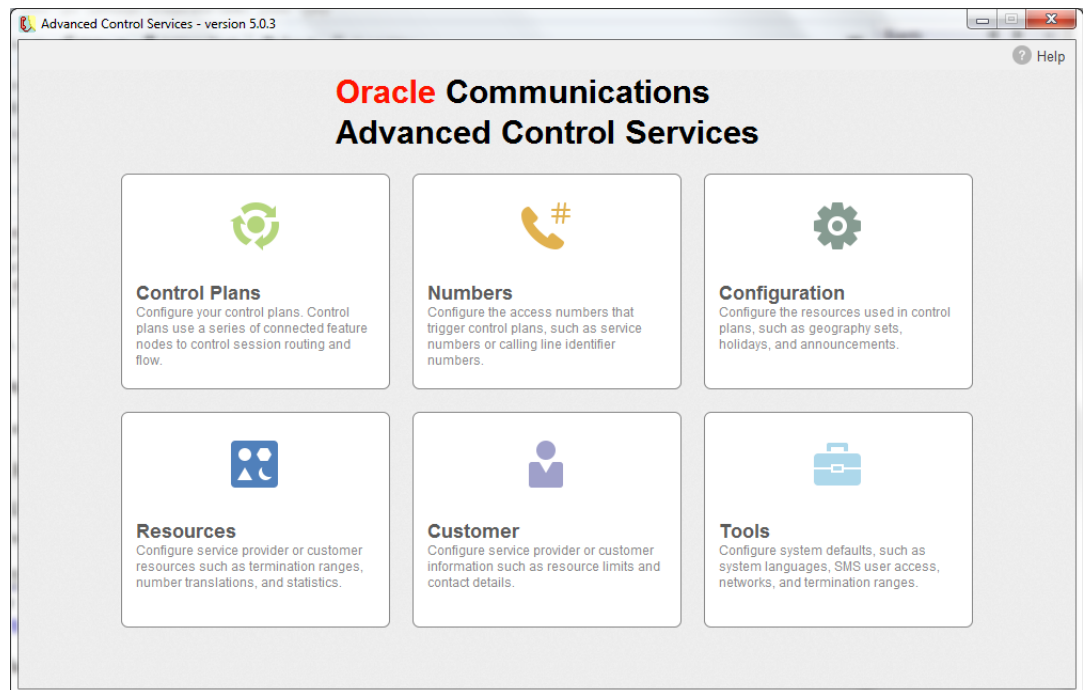
Prepaid Charging uses both CCS (Charging Control Services) and ACS (Advanced Control Services). In ACS, you can create and configure your service provider, and the control plans that will be mapped to Prepaid Charging product types.

To create a new ACS customer that will become the service provider:

1. Open the ACS UI by selecting **ACS service** from the **Services** menu in the **Service Management System** screen.

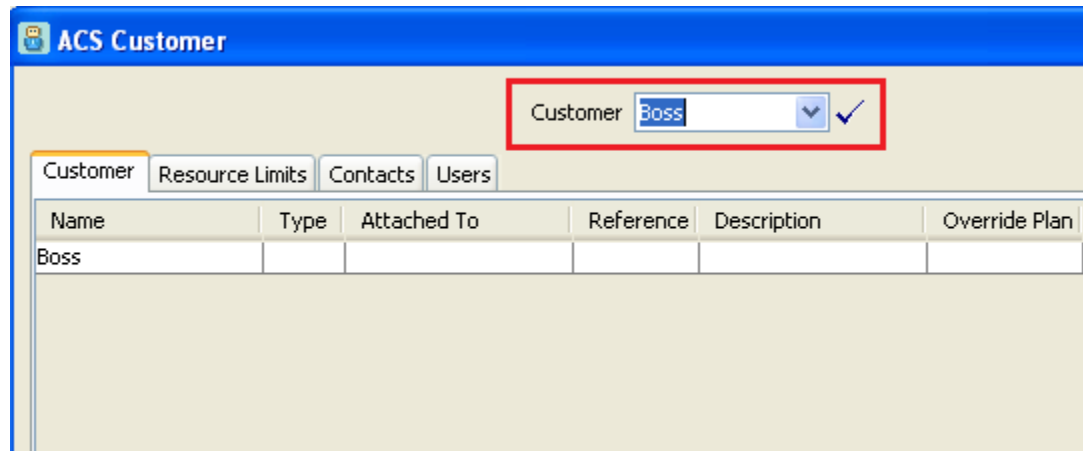


Result: The Advanced Calling Services screen is displayed.



2. Click **Customer**.

3. To see what customers are already on the system, clear the **Customer** drop down box and **Enter**.



The screenshot shows the 'ACS Customer' interface. At the top, there is a blue header with the text 'ACS Customer'. Below the header, there is a search bar labeled 'Customer' with the text 'Boss' entered. A red box highlights the search bar and the 'Boss' text. Below the search bar, there are four tabs: 'Customer', 'Resource Limits', 'Contacts', and 'Users'. The 'Customer' tab is selected. Below the tabs, there is a table with the following columns: 'Name', 'Type', 'Attached To', 'Reference', 'Description', and 'Override Plan'. The table contains one row with the text 'Boss' in the 'Name' column.

Name	Type	Attached To	Reference	Description	Override Plan
Boss					

4. Do one of the following:
 - Click **New** to create a new ACS customer.
 - Select an existing customer and click **Edit**.
5. Give your customer a unique name and reference number (**Customer Name** and **Customer Reference** fields).

New Customer

Customer Name

Description

Customer Reference

Customer Type **Normal** ▾

Attached To Telco
 Reseller **Asia** ▾ ✓
 Agent **Next-Door Telecoms** ▾ ✓

Resource Multiplier

Use override control plan

Override Control Plan **ACS Management** ▾ ✓ **Version: 135**

Language **English** ▾

PIN

Management ID

Managed Customer **Termination Number Range Rules**
 Own Range
 Default Range
 No Checking

Create User for Customer

Managed customer option means that it is managed through the SMS GUI, so requiring a login to those first. It is also possible to connect directly to the ACS UI by clearing the **Managed Customer** check box.

An administrative user for your new customer will be created for you if you select **Create User for Customer**.

6. Select the Termination Number Range Rules.

The default is **No Checking**. This will use the Termination Ranges set up on the **Termination Ranges** tab of the ACS Resources screen.

7. Click **Save** when you have finished adding your customer.

Note

You may need to clear the **Customer** drop down field (often defaulting to Boss) and **Enter** to see your new customer in the list.

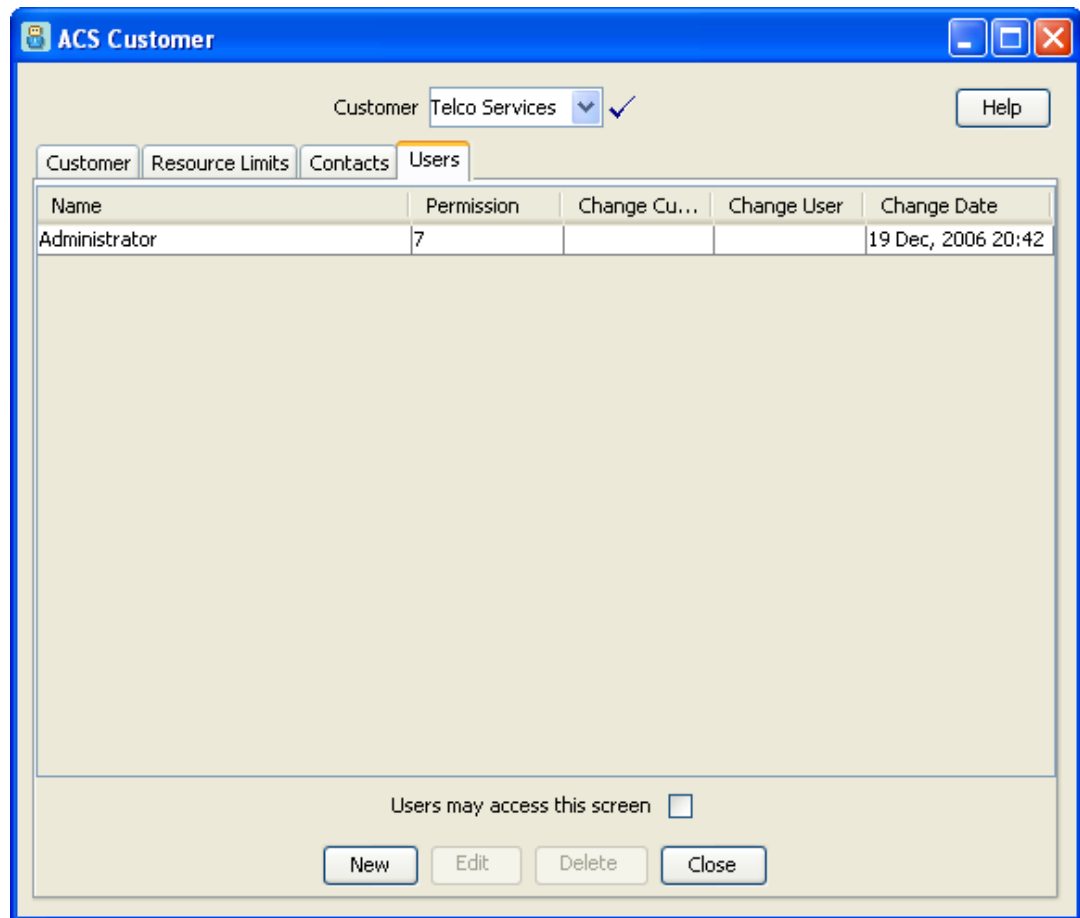
Self Managed Customer

A self managed ACS customer is a customer who is managed directly in ACS, instead of being managed by the operator in SMS. You must manually create an ACS user for each self managed ACS customer, to enable them to log in to ACS directly.

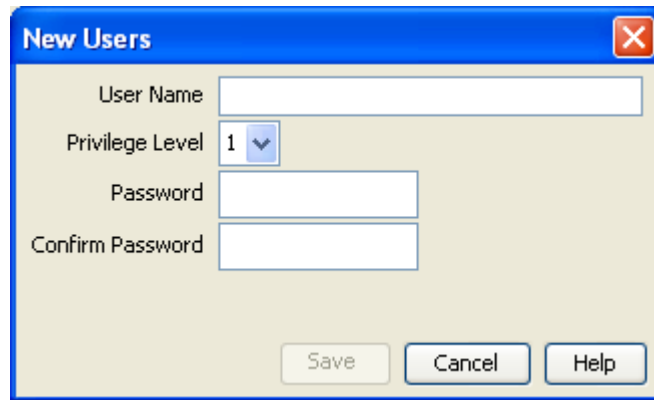
To create or edit a self managed customer user:

1. In the ACS Customer window, select your new customer from the **Customer** list.
2. Select the **Users** tab.

Result: You see a list of all users that the ACS customer has. If you selected **Create User for Customer** when creating a new customer, you will see Administrator in this list.



3. Click **New** to create a new user, or **Edit** to change an existing user. For example you can edit the Administrator user to rename them to be the customer's ACS user.



4. Enter a name, privilege level, and password for the user.

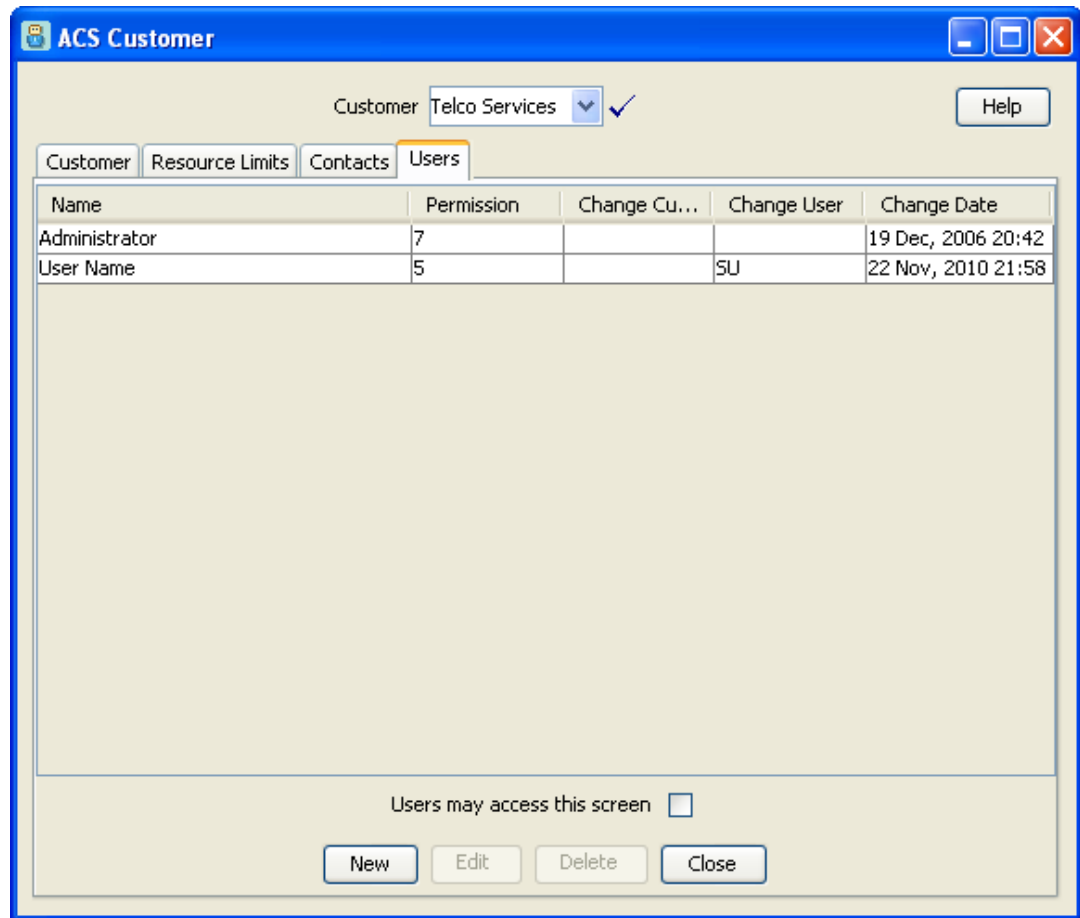
Whether you rename Administrator or create a new user, ensure your user has privilege level 5.

Note

You may also wish to create a level 6 user for your customer just to compare the differences between level 5 and level 6 users. To allocate privilege 6 to the user, you must log into the Service Management System with a privilege level 7 user (such as the su user).

5. When you have finished creating or editing your user, click **Save**.

The updated list of users is displayed.



6. Click **Close**.
7. To test the ACS user for your ACS customer, log in to ACS directly as the ACS user, by running **acsGui.sh** or **acsGui.bat**.

Create or Allocate Data Sets to your Service Provider

You can use existing reusable data sets (from the ACS Configuration screen). Ensure your ACS customer/service provider has at least a public set of the following:

- Geographical set
- Holiday set
- Announcement set
- Feature node set (including ACS and CCS feature nodes)

To create or allocate data sets to your service provider:

1. Select **Services**, then **ACS Service**, and then **Customer**.
2. Allocate Resource limits.

On the **Resource Limits** tab, select your customer and click **Edit**.

Note

If there are no publicly available data sets for you to use, on the system, you'll need to create your own data set(s).

3. On the Edit Customer Resource Limits screen, set the numbers allowed and click **Save**.

ACS Configuration

To configure the ACs component of Oracle Communications Convergent Charging Controller, select **Services**, then **ACS Services**, and then **Configuration**.

Configure the following:

- Geography set
- Holiday set
- Announcement set
- Feature node set

Geography Set

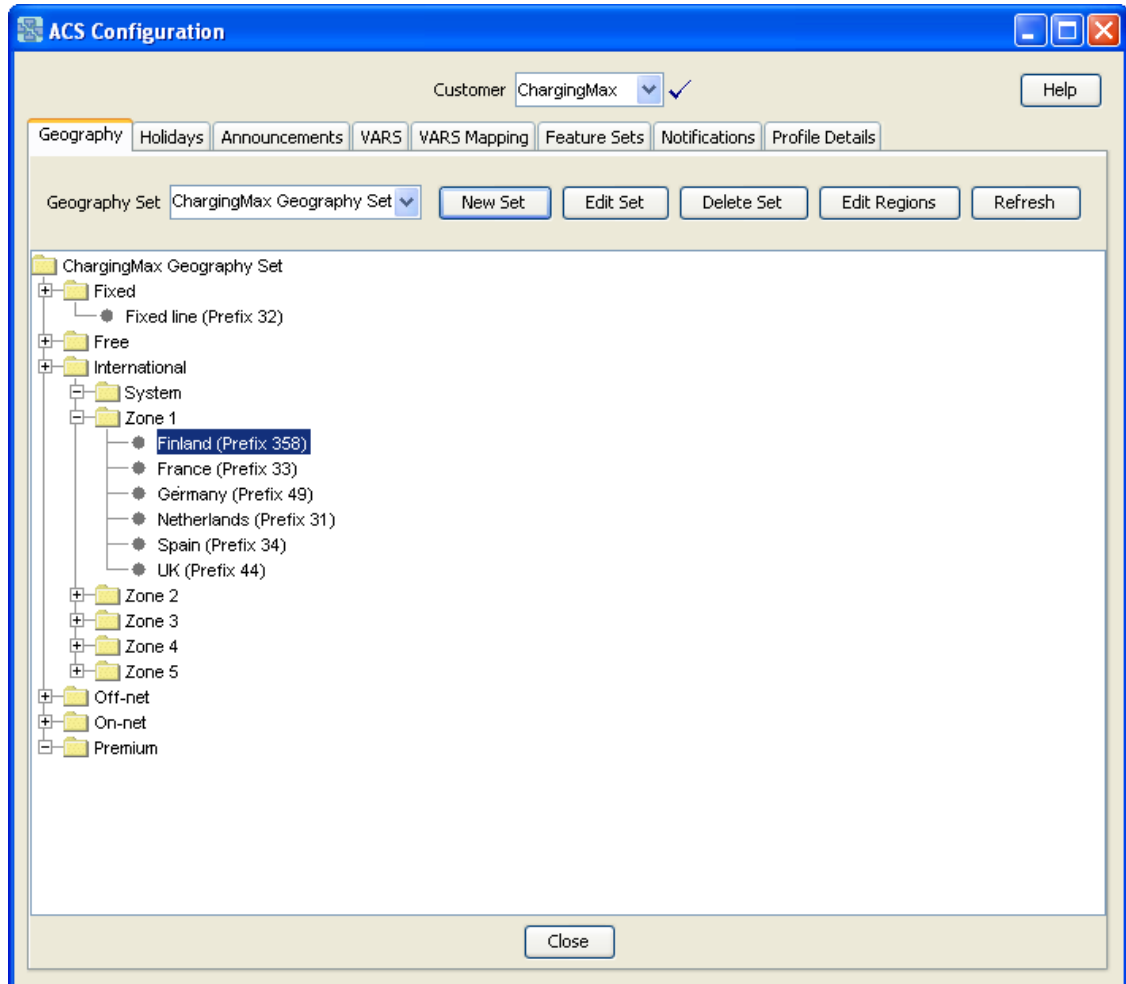
A geography set is used to determine the rating. This is done by grouping prefixes. This allows you to group sets of numbers in one geographical location, so you can create a tariff for all of them at once (but you can also create a tariff specific to each of the prefix).

For example, all mobile numbers (3249, 3247 and 3248) can be grouped under "Belgium Mobile" and a tariff to be created for them.

The prefix defines a range of numbers which is generally used to differentiate different operators.

The following screen shows the default prepaid charging geography set.

Figure 2-1 Prepaid Charging Geography Set

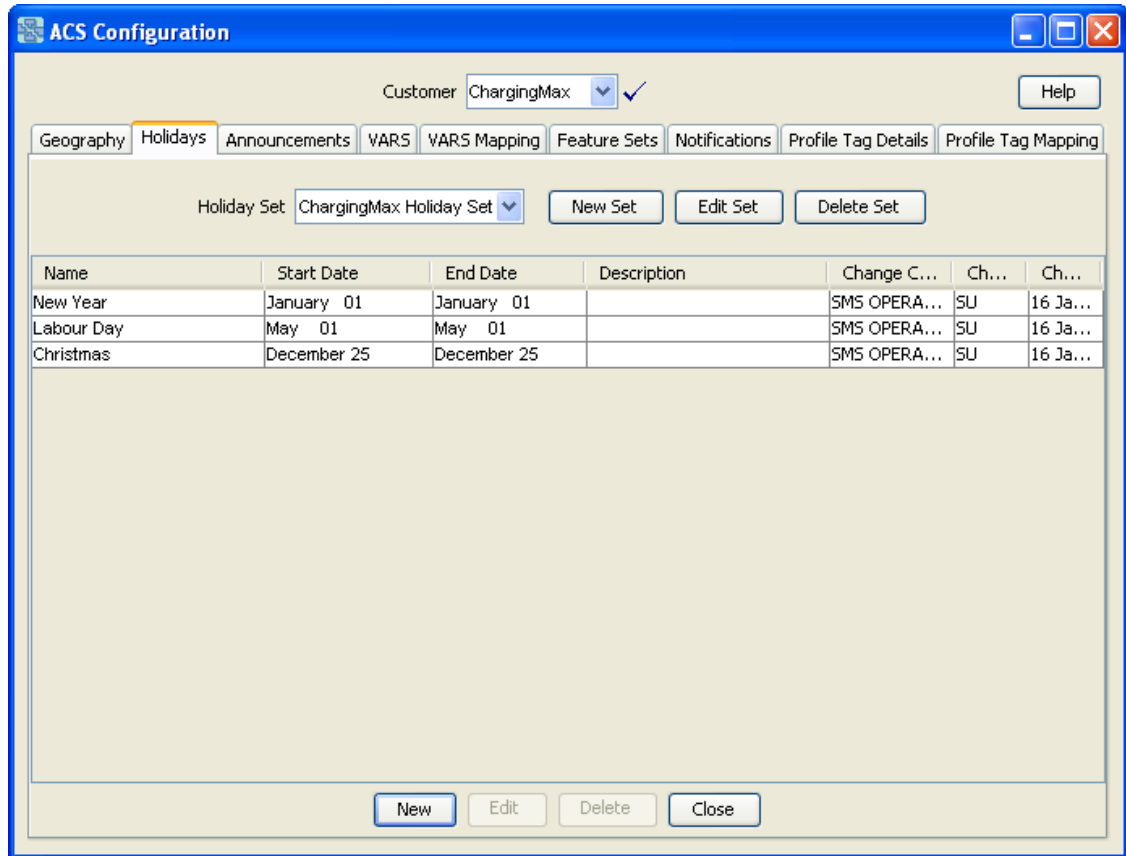


Holiday Set

This is where holidays are defined (for example, Christmas and New Year.)

Holidays can be defined so that there are cost reductions during that time. Holidays go from a start date to an end date.

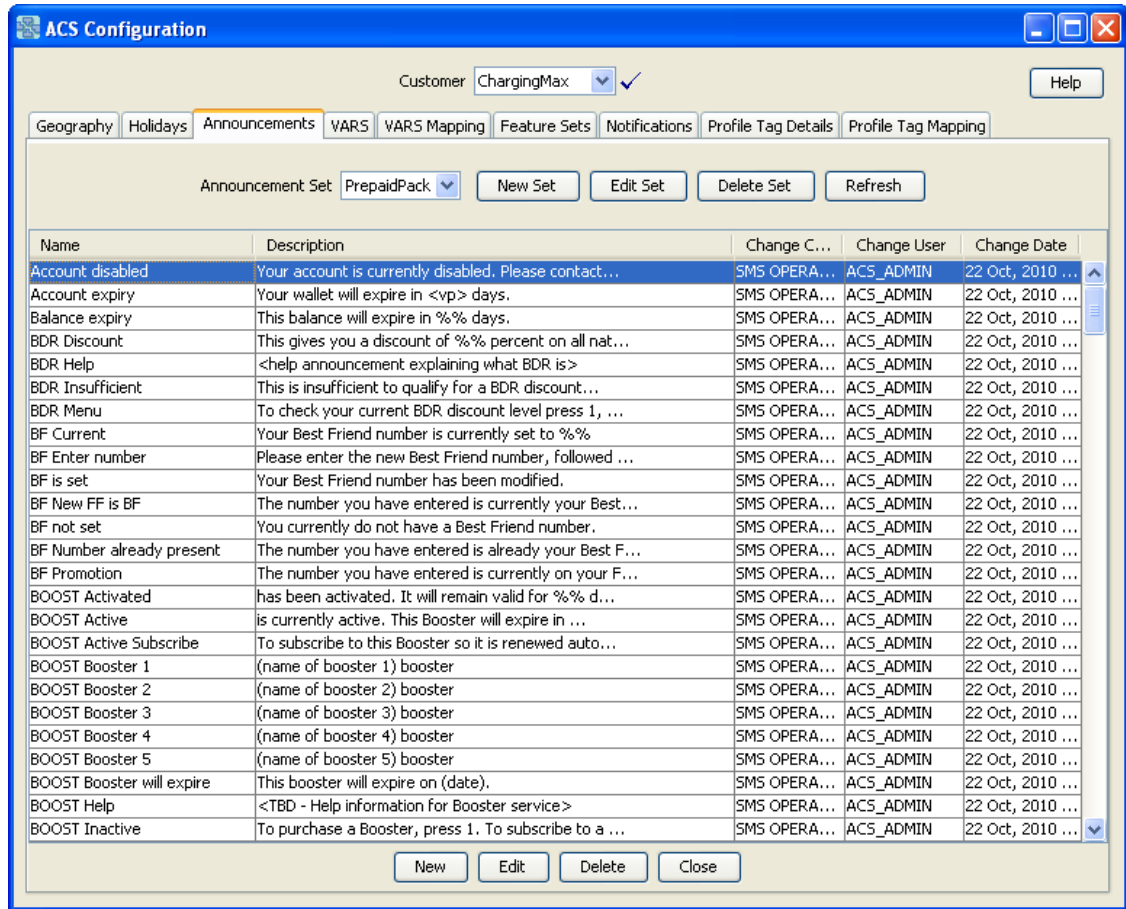
Figure 2-2 Holiday Set



Announcement Set

An announcement set is a collection of recorded voice announcements for subscriber interaction.

Figure 2-3 Announcement Set



In that announcement set, you have a collection of announcements which are linked to a resource ID (to be played on the IVR) and a corresponding language.

Figure 2-4 Edit Announcements

Entry

Name: Account disabled

Description: Your account is currently disabled. Please contact...

Language	Resource Name	Resource ID	VARS Mapping
English	SRF	100	-
French	SRF	100	-

Mapping Editor

Language: English

Type of Mapping:

- Resource ID + Resource Name
- VARS Mapping

Resource Name: []

Resource ID: []

Add

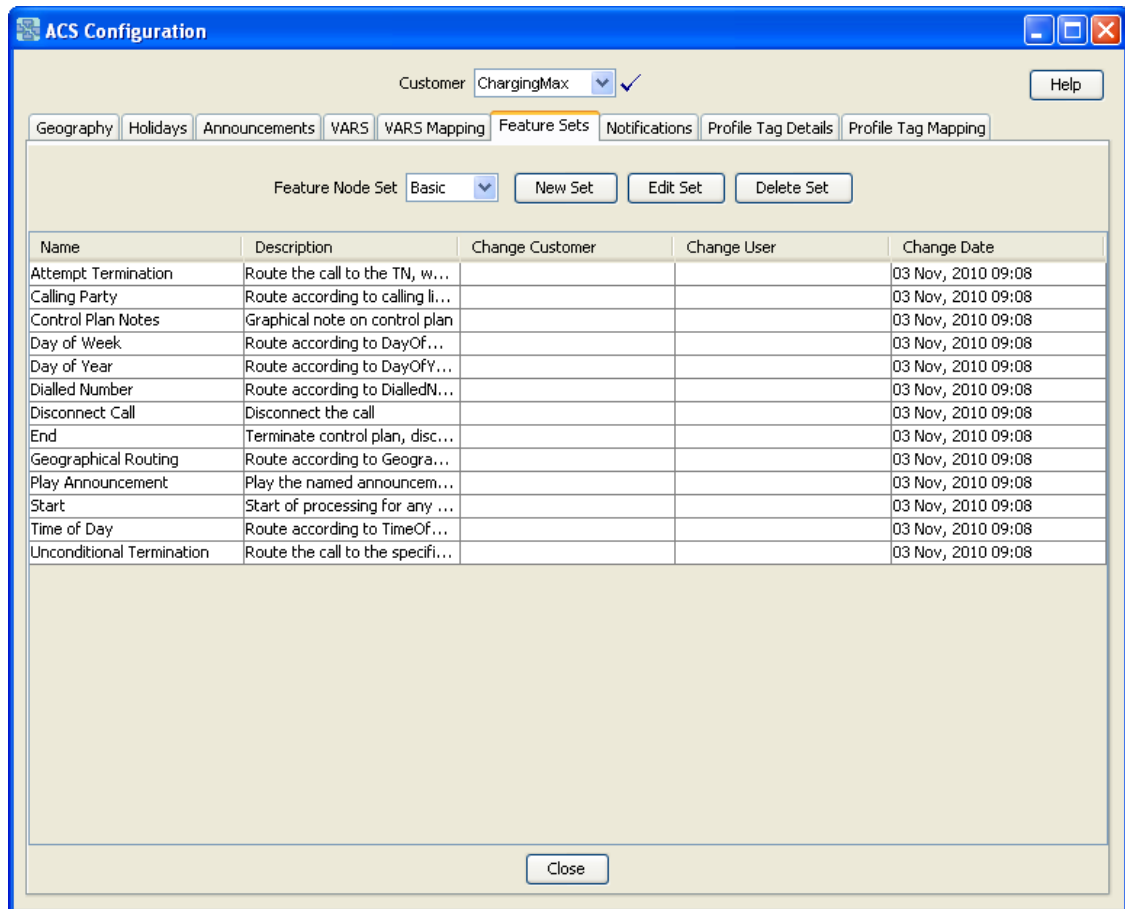
Remove

Save Cancel Help

Feature Node Set

This describes the feature nodes that are available for creating a control plan.

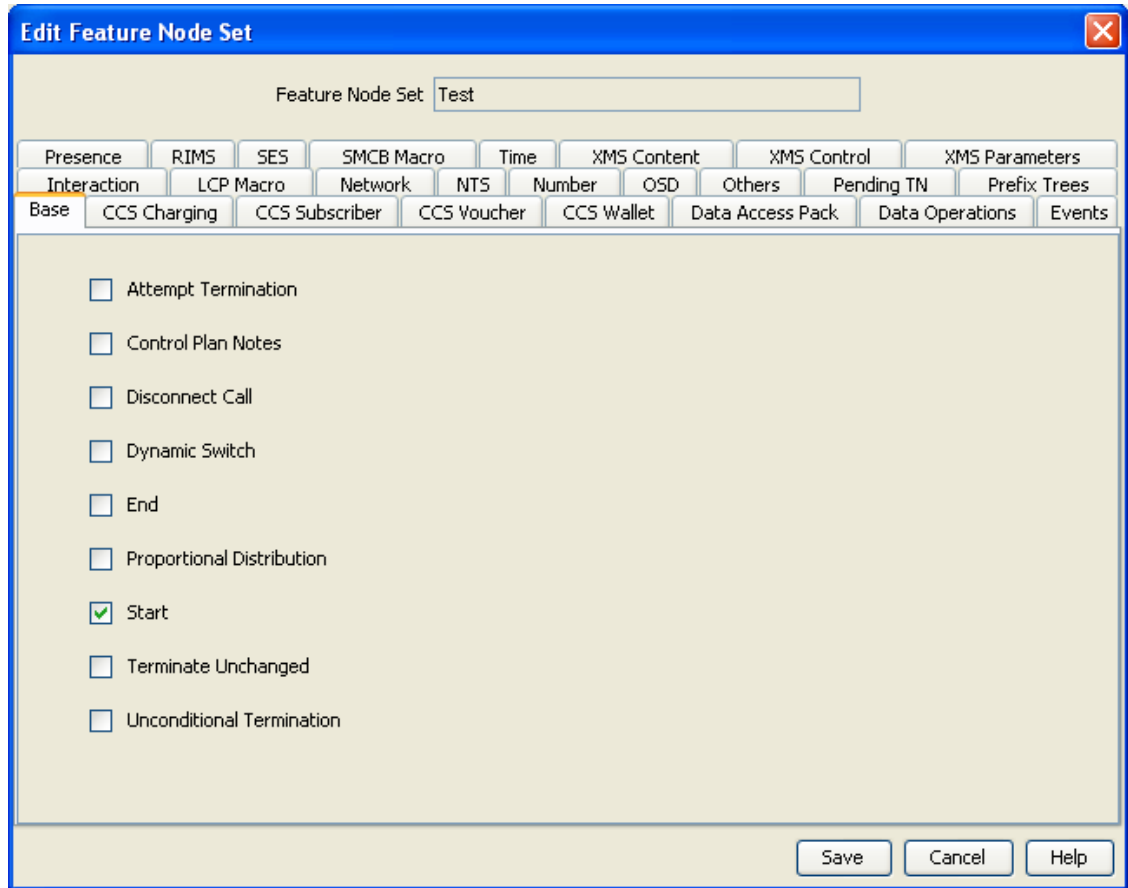
Figure 2-5 Feature Nodes



You can create a specific set to restrict rights for your users.

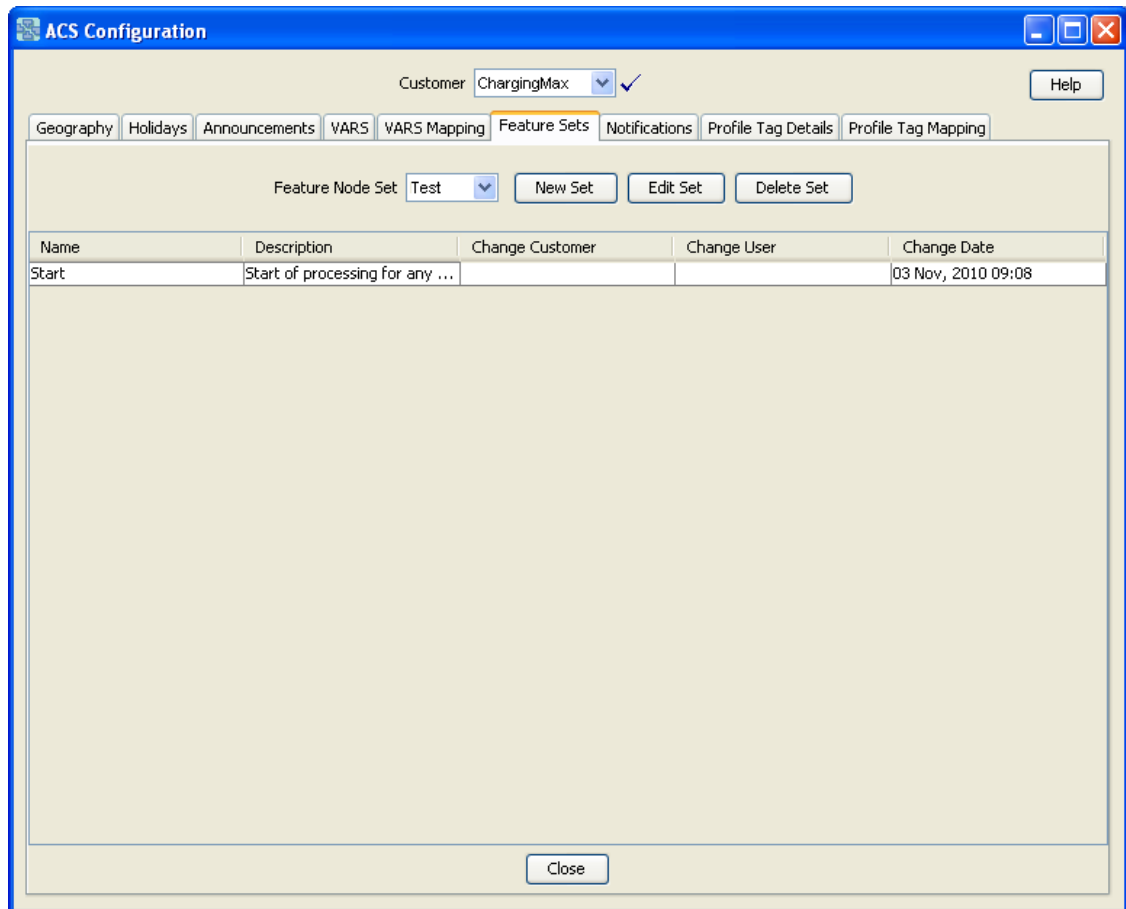
Example: If you create a feature set, then edit it to include only the Start feature node:

Figure 2-6 Edit Feature Node Set



The feature set will look like this:

Figure 2-7 Feature Set



Service Provider Limits

Using CCS, for the service provider, you can set the resource limits, in particular:

- Limits
- Business Prefix
- Sets
- Call barring
- Number translations

Setting Limits

To set resource limits:

1. Select **Services**, then **Prepaid Charging**, and then **Service Management**.
2. Select the **Resource Limits** tab.

The screenshot shows a window titled "Service Management" with a "Service Provider" dropdown set to "Boss" and a "Help" button. Below the title bar is a menu bar with the following items: "Number Translation List", "Global Configuration" (highlighted), "Splash Screen Configuration", "Balance Type Mapping", "Security", "Exchange Rates", "MFile Generation", "Resource Limits", "Capability", "Domain", "Currency", "Barred List", and "Currency Code". The main area contains several input fields:

- Maximum Product Types: 0
- Maximum Voucher Types: 0
- Maximum Rate Tables: 0
- Grace Period (seconds): 0
- Maximum Bad PIN Count: 3
- Number of Product Types Used: 0
- Number of Voucher Types Used: 0
- Number of Rate Tables Used: 0
- Number of Promotion SMSs Sent: (empty)
- Global Business Prefix: 111

At the bottom of the window are "Edit" and "Close" buttons.

3. Click **Edit** to open the Edit Resource Limits screen.

The screenshot shows the 'Edit Resource Limits' dialog box. The left sidebar contains a tree view with the following items: Limits (selected), Business Prefix, Sets, Call Barring, Number Translations, Bypass Numbers, Bypass Number Discount, Profile Group Limitations, and Promotions. The main area displays the following settings:

- Maximum Product Types: 102
- Maximum Voucher Types: 102
- Maximum Rate Tables: 37
- Grace Period (seconds): 20
- Maximum Bad PIN Count: 3

At the bottom of the dialog are 'Save' and 'Cancel' buttons.

4. In the following options, choose and set appropriate limits for the service provider:
 - Limits
 - Business Prefix – Business Prefix must be provided otherwise the **Save** button will be disabled.
 - Sets – Select all the Reusable Data Sets which the new service provider should have access to.

Note: If the Business Prefix is entered by the subscriber at the beginning of a dialed telephone number the call will be charged to their business wallet instead of their personal wallet. Allowed values include 0-9, # and *.

This functionality is dependent on the configuration of the control plan.

5. In the Call Barring option, ensure the **Ignore** check box is selected for this service provider.
6. In the Number Translations option, set up a short number which will connect directly to the main company's call center.
 - a. Enter short code to be dialed into the **In** field.
 - b. Enter the termination number (actual phone number to be dialed) into the **Out** field.
 - c. Click **Add**.
7. Click **Save**.

3

Product Type

This chapter explains the product types.

Product Types

A product type is a collection of services that are provided to the group of subscribers who use that product type.

Creating Product Types

To create a product type:

1. You create a product type with:
 - A default tariff plan. See *Creating a Tariff* to create one.
 - A control plan associated with it. See *Create a Control Plan* to create one.
 - Some announcements. See *Announcement set* to create some.
2. On the Subscriber Management screen, select the **Product Type** tab, then **New**.
3. Set the **Initial Value** of the product type to 20 euros.
4. Create a new CCS Capability, see *Creating a Capability*.
Select a CCS Capability from the drop down menu (defined in the **acs.conf** file).
5. Select the control plans option, then **New**.

The screenshot shows a 'New Control Plan' dialog box with the following fields and values:

Field	Value
CCS Capability	HPLMN MO
Control Plan	HPLMN-MO
Tariff Plan Override	<input type="checkbox"/>
Override Tariff Plan with	Bronze Callback

6. Select the control plan created earlier.

7. Select the **Announcements** option and select the **Announcement set** created earlier.

4

Subscribers

This chapter explains how to create subscribers.

Subscriber Creation

The CCS component of Prepaid Charging provides subscriber account management and tariffing. The VWS provides subscriber account's fund management and implements charging reservations and debiting. The VWS supports these types of subscriber accounts:

- Prepaid
- Postpaid
- Limited Credit
- Prepaid
- Throwaway

A product type is assigned to a subscriber, so this must be first set up before you can create a subscriber. For this example, you will set up a test subscriber account. This will be used to test your product type created in the previous example. You will also use this subscriber for subsequent examples.

Credit Type

A subscriber's credit type is set by its Wallet Limit Type:

- Credit/Postpaid subscriber – Fully postpaid and rechargeable.
- Limited Negative Credit subscriber – Rechargeable subscriber account with a post-paid facility that allows it to go overdrawn to a configurable limit.
- Debit/Prepaid subscriber – Rechargeable prepaid.
- Single User Prepaid Card – Non-rechargeable credit (can be used as a promotional give-away).

Subscribers and MSISDNs

A subscriber account is linked to a specific MSISDN. The MSISDN is identified by one of the following:

- The Calling Line Identifier (CLI), also known as the Calling or Originating Number.
- A prefix dialed by the subscriber before a call.

You can configure a subscriber in the following ways:

- As a subscriber with a single balance.
- As a subscriber with one or more wallets, and multiple balance types.
- As a balance subscriber account used by multiple MSISDNs.

You can link more than one MSISDN to a subscriber.

Subscriber Experience of Making a Call

How a subscriber makes a call depends upon whether the caller is using the subscriber's known phone line (CLI), or making a call from an independent phone.

If using a configured CLI, the subscriber goes off-hook and dials a destination number. The control plan for the product type is triggered based on the CLI.

If the subscriber is making a call from an un-configured CLI, they call the service, (for example, by dialing a dedicated 0800 number) and Oracle Communications Convergent Charging Controller processes the call as follows:

1. They are played a beep and must enter their subscriber account ID and PIN.
2. If successfully validated, the caller is prompted to enter the intended destination number. The control plan for their product type is triggered on their subscriber account ID.
3. The service then checks their balance status and either connects the call as requested, disconnects (in case of no credit), or redirects the call (depending on the control plan).

Subscriber accounts must have product types (which define how the account is charged), and wallets (which hold value). The requests that Prepaid Charging makes on the billing engine contain information about the wallet it is accessing.

Viewing Existing Subscribers

To find all existing subscribers:

1. Select **Services**, then **Prepaid Charging**, and then **Subscriber Management**.
2. Select the **Subscriber** tab.
3. In the **Card Number** text field, type % and **Search**.

Result: The list of subscribers, if any are present, is displayed in the table.

For each you see: wallet reference, card number, and subscriber ID.

Creating a Subscriber Account

To create a new subscriber account:

1. Select **Services**, then **Prepaid Charging**, and then **Subscriber Management**.
2. Select the **Subscriber** tab.
3. Click **New**.

Result: The New Subscriber screen is displayed.

4. Enter:
 - Card Number
 - Subscriber ID
 - Language

Click **Save**.

Result: The New Wallet screen is displayed.

5. Select the tariff plan created earlier, the billing pair you want it to be on.
This is enough to start doing chargeable calls for this subscriber.

Wallet

A wallet, in concept, is used to hold the balances, for example, for SMS, voice, and general cash. A wallet is allocated to a subscriber to hold their balances. Each subscriber can have a personal and a business wallet (which may be shared). To create or edit wallet data, the SMS must be successfully communicating with the associated VWS. The **ccsBeOrb** process is used for this communication.

Complete the New Wallet screen. Set up the required data and accept the defaults by clicking **OK**.

Block a Subscriber

For this task, you will test that a frozen subscriber account cannot make a call.

Change your subscriber's account to freeze the account:

- On the Edit Subscriber screen, **Subscriber** and **Wallets** option, change the **Wallet State** to frozen.
- **Apply** and **Save** the changes.
- Run a Split script to emulate the above 'frozen' subscriber trying to make a call.

Unblock a Subscriber

To unblock the previously blocked subscriber account.

1. Search for the previously blocked subscriber's account and open it.
2. Unblock the account by changing the account status so that it is active.
3. Run a Slpit script to emulate the subscriber trying to make a call.

5

Rating and Tariffs

This chapter explains rating and tariffs.

Tariff Creation

Prepaid Charging calculates the standard tariff from the CLI (calling line identifier / origin) and SN (service number / destination) numbers of the call and accepts or rejects a call. Your new brand provides a standard calling rate which enables subscribers to call cell phones at a reduced cost.

Rating

Rating is provided for:

- Voice calls
- Data calls
- Specified Events (including SMS messages)

Call charges can include a:

- Maximum or minimum charge for each call/data session
- Minimum charge period after which charging starts
- Low credit warning threshold
- Rating can be based on complex configurable factors, including:
 - Duration
 - Destination number, (including its geographical location)
 - Foreign roaming network
 - Day of year, day of week, and time of day

Components

Calls are controlled using ACS.

The Voucher and Wallet Server (VWS) handles money.

CCS deals with charging rules, for example: it costs x amount of money to call from place A to place B for n minutes. There are also further calculations related to discounts for certain days/weeks and specific sections of a call. The charge for a call or data session is set by the subscriber's product type in CCS.

Tariff Plans

Tariff plans enable you to define which rate table applies to a product type at a particular time.

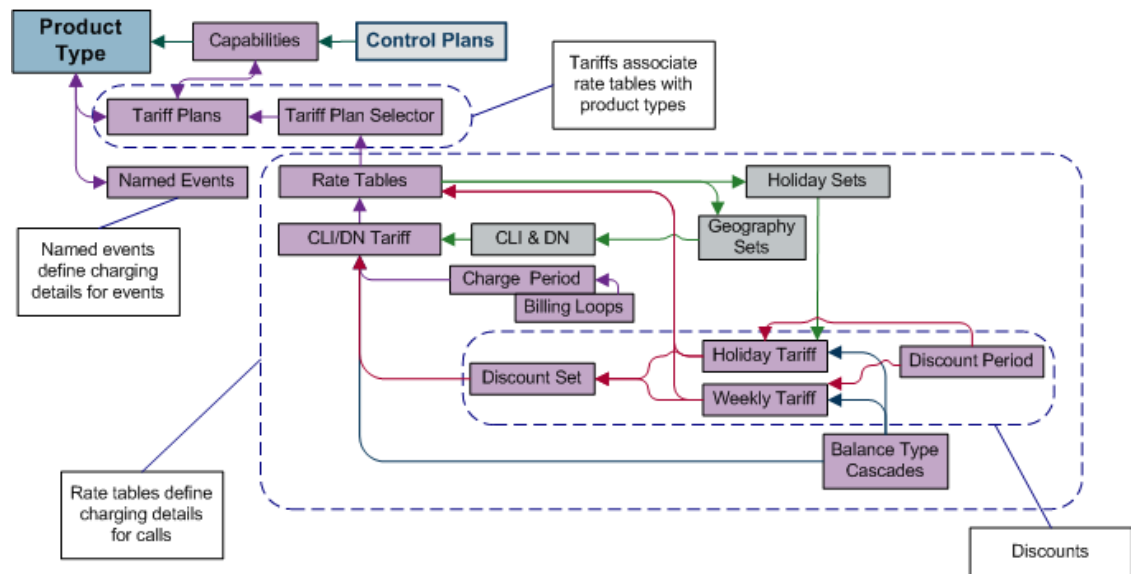
Rate tables define how Prepaid Charging charges for calls.

An ACS customer requires a tariff plan, such that they can provide a service which charges special rates for some locations.

Tariff Configuration

This diagram shows tariff configuration.

Figure 5-1 Tariff Configuration



Creating Tariff Plan

The tariff plan defines rules for how a call by your subscribers will be charged. You will also generate the memory mapped file used on the billing engine.

For your product type, you will create a tariff, by:

- Configuring a rate table.
- Rate table, charge periods, balance type cascade, discount sets, CLIxDN tariffs, and the rest.
- Configuring a tariff plan.
- Link tariff plan to rate table using the **Tariff Plan Selector** tab.
- Configuring discounts.
- Create discount periods, and weekly and holiday tariffs.
- Creating named events.
- Creating a MFile (memory mapped file for BE containing rating information).

Tariffing needs to be created in order to be able to do billing. It all starts by creating a new tariff plan, this will be the way of billing used in the product type.

Creating a Tariff

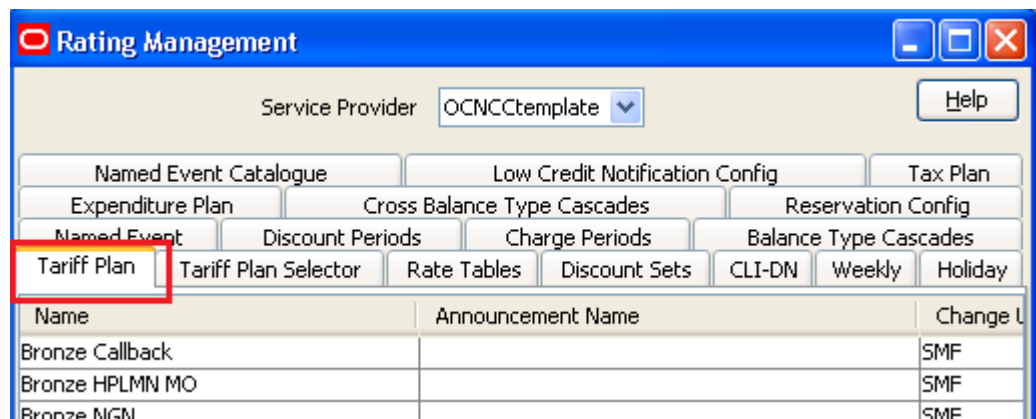
In order to create a proper tariff, you'll need to create the following, in this order.

- Geography set
- Holiday set
- Tariff plan
- Rate table
- Tariff plan selector
- Balance type cascade
- Charge period
- Discount set
- Discount period
- CLI-DN
- Weekly tariff
- Holiday tariff
- MFile

Tariff Plan

To create a tariff plan:

1. Select the **Tariff Plan** tab.



2. Click **New**.

Result: The New Tariff Plan screen is displayed.

3. Give your tariff plan a name.

- Click **Save**.

Rate Table

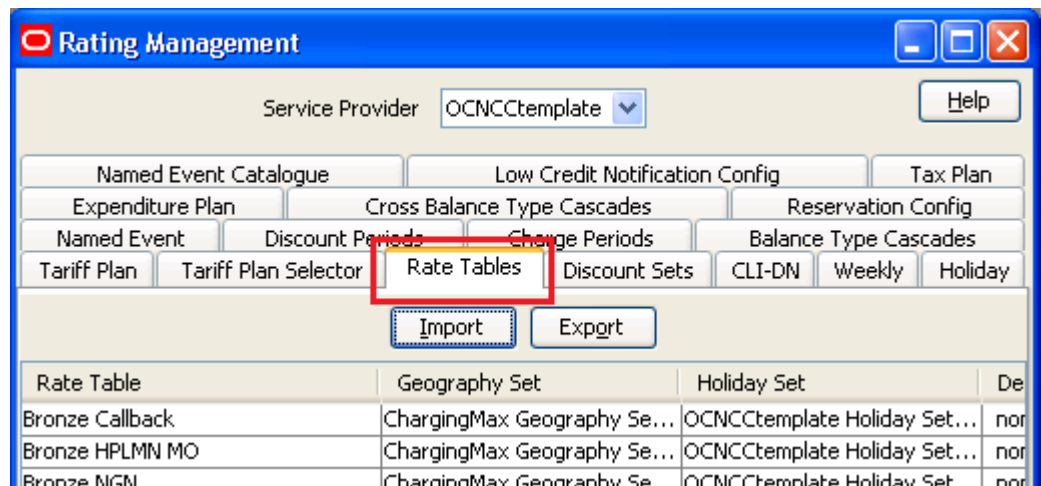
Rate tables define charging details for calls. This data is used to set the charging rates according to specific dates and geographical location.

Note

It is not possible to change the geography set for a rate table once saved. You first must delete all records associated with the rate table, then delete the rate table and re-create it.

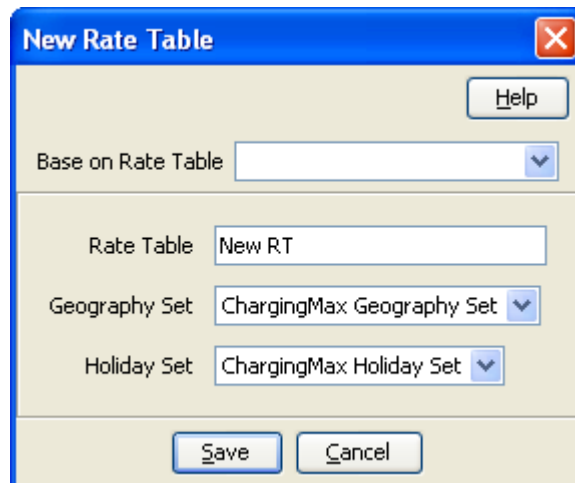
To create a rate table:

- Select the **Rate Tables** tab.



- Click **New**.

Result: The New Rate Table screen is displayed.



- Give your rate table a name.
- Select a Geography set.

5. Select a Holiday set.
6. Click **Save**.

Tariff Plan Selector

The tariff plan selector links the tariff plan and the Rate Table.

To create a tariff plan selector:

1. Select the **Tariff Plan Selector** tab.
2. Click **New**.

Result: The New Tariff Plan Selector screen is displayed.

3. Select a tariff plan.
4. Select a rate table.
5. Update the effective date.
6. Click **Save**.

Balance Type Cascade

A Balance Type Cascade includes the different Balance Types that are available to pay for the cellphone calls. Here you will set the order in which the balance types will be exhausted.

Note

You may want to use this Balance Cascade pattern in other tariffs, so give it a name which is descriptive of the pattern, not the tariff you are planning to use it in.

The Balance cascade is how the balance is going to be used, if we take the example bellow, we will first use the Promotional Cash, then the General Cash, then the Free SMS. For example:

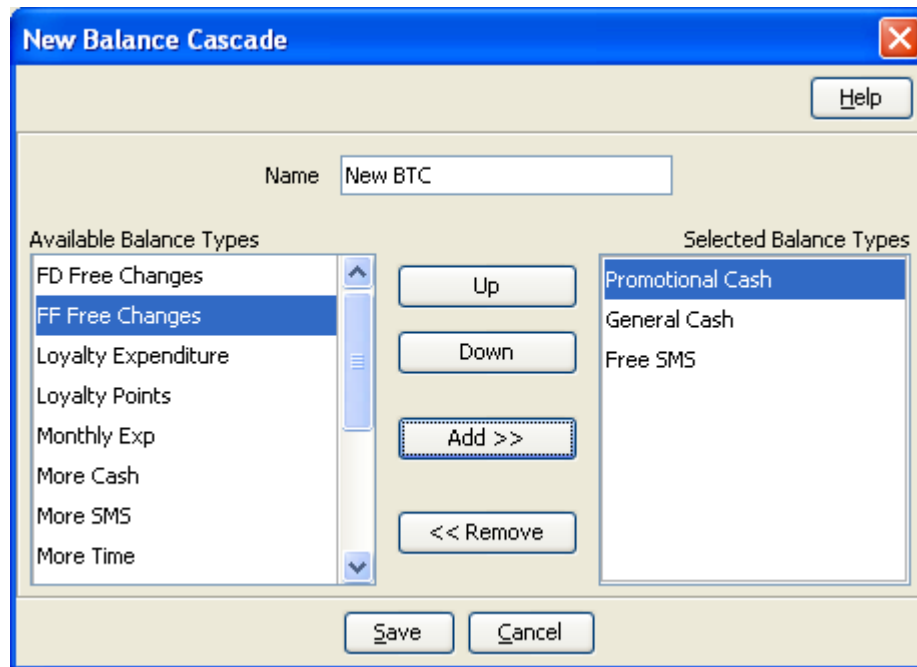
1. Promotional Cash
2. General Cash

3. Free SMS

To create a balance cascade:

1. Select the **Balance Type Cascades** tab.
2. Click **New**.

Result: The New Balance Cascade screen is displayed.



3. Type a Balance Type Cascade name.
4. Select an available balance type and **Add**. Repeat until you have selected all balance types you require.
5. Use **Up / Down** to order the selected balance types.
6. Click **Save**.

Charge Period

A charge period is charging rate per minute. First you create a charge period set by simply giving it a name, then you choose a rate table to give you the charge rates. With this information you can then create the charge periods for that set.

For example: 10 cents the first 60 seconds, then 5 cents per minute for the rest of the call.

To create a charge period set with charge periods:

1. Select the **Charge Periods** tab.
2. Click **New Set**.

Result: The New Charge Period Set screen is displayed.

3. Type a name for the charge period set.
4. Select the associated rate table.
5. Select the billing indicator.
6. Click **Save**.
7. Highlight the charge period and click **New**.

Result: The New Charge Period screen is displayed.

8. The first period starts at 0 sec and is charged at 10c/min.

Create a charge period with a:

- Period start of 0.
- Charge of 10.

and click **Save**.

9. The second period starts at 60 seconds and is charged at 5c/min, but the charge is per second.

Create another charge period with a:

- Period start of 60.
- Charge of 5.

and click **Save**.

Refer to *CCS User's Guide* for more information about the fields.

Discount Set

The discount set is only a name and description.

This set is associated with the holiday and weekly tariffs and is used to provide a logical link from the rate table to the CLI-DN data. If no holiday or weekly discounts are set, the discount set will not apply.

A discount period defines the discount percentage to be applied to a charge period.

To create a new discount set and discount periods for the discount period set:

1. On the **Discount Sets** tab, select, from the **Rate Table** drop down box, the rate with which to associate a discount set.
2. Click **New**.

Result: You see the New Discount Set screen.

The **Rate Table** field displays the rate table selected.

3. Enter a name and description.
4. Click **Save**.

Discount Period

A discount period defines the discount percentage to be applied to a charge period.

To create a new discount set and discount periods for the discount period set:

1. Select the **Discount Periods** tab.
2. Click **New Set**.

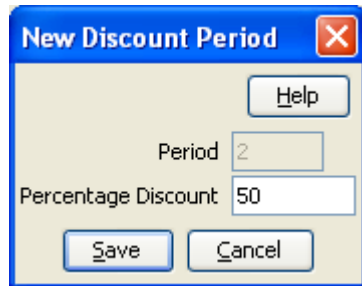
Result: You see the New Discount Period Set screen.

3. Enter a name and click **Save**.
4. Then create a new discount period where a percentage of discount is defined.

On the **Discount Periods** tab, select the discount period set from the **Discount Period** drop down box.

5. Click **New**.

Result: You see the New Discount Period screen.



The **Period** field displays the amount of time before the charge is initiated.

In the **Percentage Discount** field, type the numeric percentage of the discount to be applied to the charge period, in this example, 50%.

6. Click **Save**.
7. Create as many discount periods as required to match your charge period.

Refer to *CCS User's Guide* for more information about discount usage.

CLI-DN

A CLI-DN (Calling Line Identifier – Destination Number) record defines, for the rate table, the rate for calls between two geographical areas.

Use the records you have defined so, to configure a CLI-DN record applicable to the rate table you created. Each rate table includes rates for calls between two geographical areas.

The tariff code is a 2 digit hexadecimal code that is usually provided by the switch manufacturer. The tariff code is set in **Tariff Codes**. You can find it if you click **ACS Services**, then **Resources**, and then **Tariff Codes**. There you will see SCI (Set Charging Info) and FCI (Furnish Charging Info). SCI is set when you receive the tariff code from the switch.

To create a new CLI-DN:

1. Select the **CLI-DN** tab.
2. Click **New**.

Result: The New CLI-DN Tariff screen appears.

Select the following from its respective drop down list:

- DN
- Discount set
- Charge period
- Cross balance type cascade

Set the following fields:

- Maximum charge
- Minimum length
- Billing resolution

- Low credit

3. Click **Save**.

Weekly Tariff

Weekly tariffs are used to define specific discounts during periods of the week (off-peak for example).

To create a weekly tariff:

1. Select the **Weekly** tab.
2. Click **New**.

Result: The New Weekly Tariff screen appears.

New Weekly Tariff

Help

Rate Table: Bronze Callback

Discount Set: No discounts

Start Time (local TZ): Sunday 00:00

End Time (local TZ): Saturday 23:30

Discount Period: Bronze holiday discount

Maximum Charge (\$):

Minimum Length (secs):

Billing Resolution (secs):

Low Credit Configuration: -- No Notifications --

Cascade: dummyWASCascade

Tariff Code:

Save Cancel

3. Fill in the fields.
Refer to the *CCS User's Guide* for information about the fields.
4. Click **Save**.

Holiday Tariff

Holiday tariffs are used to define the specific discounts that will apply during holidays.

1. Select the **Holiday** tab.
2. Click **New**.

Result: The New Holiday Tariff screen appears.

3. Fill in the fields.
Refer to the CCS User's Guide for information about these fields.
4. Click **Save**.

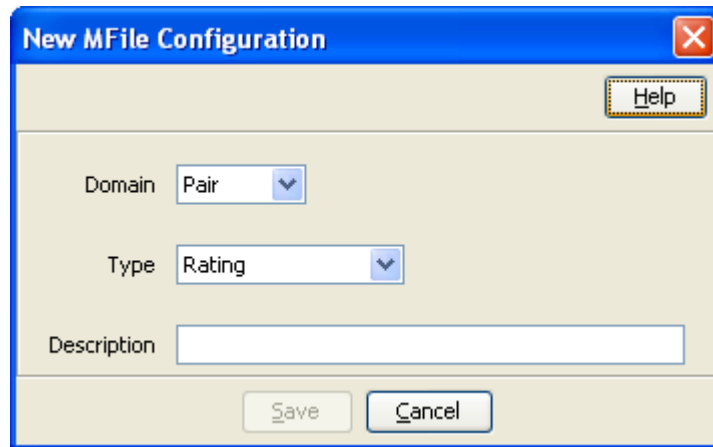
MFile

MFile is the core of the rating system. It is a file that is mapped into memory (using standard Unix mmap - memory mapping functionality). The data contained within the file is generated from all the rating data configured within the database. This file removes the requirement for database access for call rating data, which offers speedier access to the data.

To create an MFile:

1. Select **Prepaid Charging**, then **Service Management**, and then **MFile Generation** tab.
2. Click **New**.

Result: The New MFile Configuration screen appears.



The screenshot shows a dialog box titled "New MFile Configuration". It features a blue title bar with a close button (X) in the top right corner. Below the title bar, there is a "Help" button. The main content area contains three fields: "Domain" with a dropdown menu currently showing "Pair", "Type" with a dropdown menu currently showing "Rating", and a text input field for "Description". At the bottom of the dialog, there are two buttons: "Save" and "Cancel".

3. Select the Pair on which creating the MFile.
Select the Type (Rating or Named Event Catalogue).
Enter a description (optional).
4. Click **Save**.
Result: The MFile data is replicated and the copied MFile is compiled.

6

Adding a Service

This chapter explains how to add an Oracle Communications Convergent Charging Controller service.

Service Creation

After the installation processes, you now have a functional platform which contains all the required components to run the Convergent Charging Controller service.

As seen before some services are delivered as template as part of the installation. Those can be used as such with additional minor configuration. That configuration requires some advanced knowledge due to the service complexity.

Process

Here we shall develop a bit how to create a new service by explaining the different steps. Most of them have been already detailed in this guide, so we shall just refer to them.

To create a service:

1. Determine how your service will be triggered, let assume by SK (not an internal service).
2. Create the *Service Key* entry in **SLEE.cfg**.
3. Create the *Service* in **SLEE.cfg**.
4. Create the *Service Entry* in **acs.conf**.
5. Create a *Service Provider Creation*.
6. Create a *control plan* that will define what the service will be doing.
7. Create a *capability* for your service using the above control plan as default.

For now, you will be able to trigger that control plan for any subscriber.

If you want to be able to restrict or customize the action for a range of subscribers, for example, you will need to create product types and add subscribers in those product types or have multiple capabilities based on CdPN.

To create a product type, you will first need to *Rating and Tariffs*.

8. Create a *product type*.
9. *Creating a subscriber* account in that product type.

Now you have a service that will be triggered using the same service key, but which could have different behavior, tariff and configuration based on the product type.

Triggering on Service Key

Most of the time your service will be triggered from the network using an IDP.

That IDP would contain the call information that will be used to trigger your service.

IDP Fields

This table describes the function of each field.

Field	Description
appContext	This holds the value of application-context-name in the TCAP section of the initialDP. Possible values for this parameter include: "0,4,0,0,1,0,50,1" // CAP2 "0,4,0,0,1,21,3,4" // CAP3 "0,4,0,1,1,1,0,0" // CDMAGW
CallingPartyNumber	This specifies the A party number (the subscriber making the call) and is defined in SLPIT in the following format: (NOA) "MSISDN" Some examples include: (2) "200" // Unknown NOA for shortcode dialing (3) "0241497936" // National NOA (4) "64241497936" // International NOA
CallingPartysCategory	This indicates the type of calling party (for example, operator, pay phone, ordinary subscriber). Values are defined in ETS 300 356-1, an ordinary subscriber will be most common.
LocationNumber	This indicates the Location Number for the calling party. This number represents the geographical location of the A party. In a scripted test LocationNumber is often fictional as it is rarely examined in service logic.
Bearer type	The bearer parameters indicate whether the call is voice, video, etc. Some common examples are: bearerCapITC 0 // VOICE bearerCapITC 8 // VIDEO bearerCapITC 24 // VIDEO

Field	Description
EventTypeBCSM	<p>This indicates the BCSM detection point event. This parameter is used in the IDP that begins the call and also in the trigger detection point that ends the call. Trigger detection points are used for both originating and terminating voice traffic. The full list of possible values for this parameter is provided below, with the values relevant to the IDP highlighted in bold:</p> <ul style="list-style-type: none"> • origAttemptAuthorized (1) • collectedInfo (2) • analyzedInformation (3) • routeSelectFailure (4) • oCalledPartyBusy (5) • oNoAnswer (6) • oAnswer (7) • oMidCall (8) • oDisconnect (9) • oAbandon (10) • termAttemptAuthorized (12) • tCalledPartyBusy (13) • tNoAnswer (14) • tAnswer (15) • tMidCall (16) • tDisconnect (17) • tAbandon (18)
IMSI	<p>The IMSI is a unique number associated with all GSM and UMTS network mobile phone users. It is stored in the SIM inside the phone and is sent by the phone to the network. The first five digits of the IMSI are identical to the countryCode and networkCode parameters. From the example above:</p> <pre>imsi "530240100000536" countryCode "530" networkCode "24"</pre>
vlrNumber	<p>This specifies the GT of the visitor location register. In scripted tests the most important part of this number is the Country Code prefix. In the example, the subscriber is attached to a New Zealand VLR (Country Code 64):</p> <pre>vlrNumber (1) "64241420003"</pre>

Field	Description
CellGlobalIdOrServiceAreaIdFixedLength	<p>The parameter in the snoop trace is broken into four parts in the SLPIT script. These parts are extracted following the rule below:</p> <pre>CellGlobalIdOrServiceAreaIdFixedLength ::= OCTET STRING (SIZE (7))- Refers to Cell Global Identification or Service Area Identification-- defined in 3GPP TS 23.003.-- The internal structure is defined as follows:-- octet 1 bits 4321 Mobile Country Code 1st digit-- bits 8765 Mobile Country Code 2nd digit-- octet 2 bits 4321 Mobile Country Code 3rd digit-- bits 8765 Mobile Network Code 3rd digit-- or filler (1111) for 2 digit MNCs-- octet 3 bits 4321 Mobile Network Code 1st digit-- bits 8765 Mobile Network Code 2nd digit-- octets 4 and 5 Location Area Code according to 3GPP TS 24.008-- octets 6 and 7 Cell Identity (CI) value or -- Service Area Code (SAC) value -- according to 3GPP TS 23.003</pre> <p>From the example snoop above, the CellGlobalIdOrServiceAreaIdFixedLength is 35F04200023C55, which gives the following fields in the SLPIT script:</p> <pre>countryCode "530"networkCode "24"locationAreaCode 0020cellID 50005</pre>
callreference	<p>This parameter refers to a call reference number allocated by a call control MSC. This is unused in the call plan, so any previously used callReference can be used.</p> <pre>callreference "1E1B7CE329"</pre>
mscAddr	<p>This specifies the GT of the mobile switching subsystem (MSC). In scripted tests the most important part of this number is the Country Code prefix. In the example, the subscriber is attached to a New Zealand MSC (Country Code 64):</p> <pre>mscAddr (1) "64241420003"</pre>
CalledPartyBCDNumber	<p>This specifies the B party number (the number dialed on the handset) and is defined in SLPIT in the following format:</p> <pre>calledPartyBCDNumber (8) "0241499216" numberPlan 1</pre>

Service Key

When the IDP arrives on the platform it would contains an SK. This can be used to trigger the service. This is done using a mapping in the **SLEE.cfg** configuration file as follows:

The CCS_HPLMN_MO service is attached to service key 102, which is included in the file in a decimal and hex format as below:

```
SERVICEKEY=INTEGER 102 CCS_HPLMN_MOSERVICEKEY=INTEGER  
0x19200000015 CCS_HPLMN_MO
```

Refer to *SLEE Technical Guide* for more information about Service Key definitions.

Service

This service is mapped to CCS_HPLMN_MO by the following line in **SLEE.cfg**:

```
SERVICE=CCS_HPLMN_MO 1 slee_acs CCS
```

Refer to *SLEE Technical Guide* for more information about Service definitions.

Service Entry

In this example, the CCS service is triggered for mobile originating voice calls as the CCS service library is used to load the service. The Service Entry in **acs.conf** appears as below:

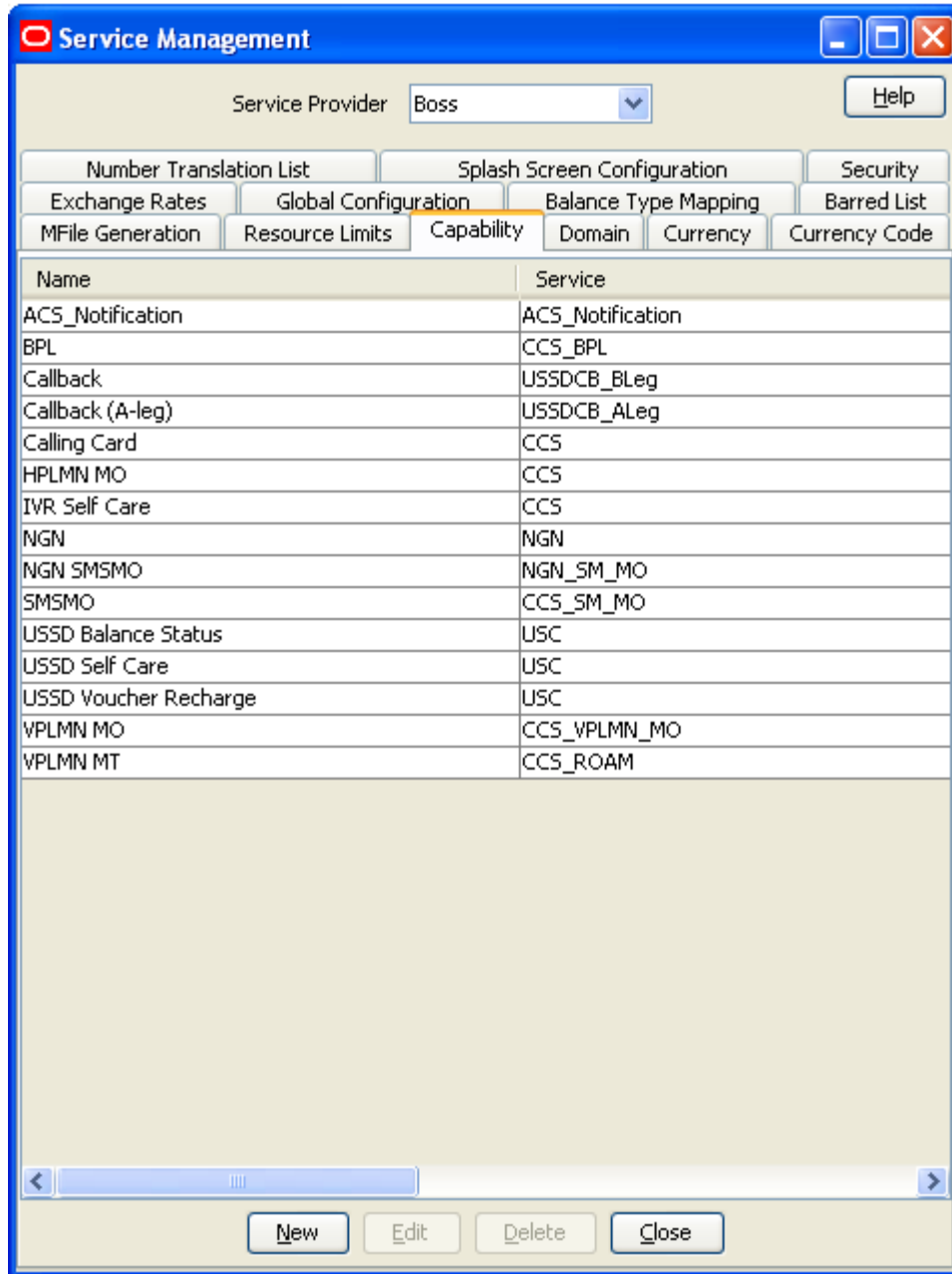
```
ServiceEntry (CCS,cCANLan1,lLcCaAnN,ccsSvcLibrary.so)
```

Refer to *ACS Technical Guide* for more information about Service Entry definitions.

CCS Capability

The CCS capabilities are defined in the Service Management screen, **Capability** tab.

Figure 6-1 Capability Tab



For example, click **Prepaid Charging**, then **Service Management**, and then **Capability**.

Figure 6-2 Edit Capability

The screenshot shows a dialog box titled "Edit Capability". It contains the following fields and controls:

- Name:** Text input field containing "Calling Card".
- Service:** Text input field containing "CCS".
- Called Party Number:** Text input field containing "9999".
- Default Control Plan:** Drop-down menu showing "Calling Card".
- Statistics Category:** Drop-down menu showing "Voice".
- Statistics Protocol:** Drop-down menu showing "CAP2".
- Bearer Capability:** Empty text input field.
- CDR Type:** Empty text input field.
- Buttons:** "Save" and "Cancel" buttons at the bottom.
- Help:** "Help" button in the top right corner.

Creating a capability

To create a capability:

1. On the **Capability** tab, click **New**.
Result: You see the New Capability screen.
2. In the **Name** field, enter the name to use to identify the capability.
3. In the **Service** field, enter name of the service for the capability.
4. In the **Called Party Number** field, optionally enter the CdPN prefix (so that the capability is called only for those terminating numbers).
5. Select the **Default Control Plan** from the drop down list (that can be overwritten in the product type if set).
6. From the **Statistics Category** drop down list, select the service to count calls through this capability against in a license report.

- From the **Statistics Protocol** drop down list, select the protocol to count calls through this capability against in a license report.

The screenshot shows a 'New Capability' dialog box with the following fields and values:

- Name: HPLMN MO
- Service: CCS
- Called Party Number: (empty)
- Default Control Plan: Postpaid
- Statistics Category: Voice
- Statistics Protocol: CAP2
- Bearer Capability: (empty)
- CDR Type: (empty)

Buttons: Save, Cancel

- Click **Save**.

 **Warning**

You must restart the SLEE in order for the changes to take effect. For more information about restarting the SLEE, see SLEE Technical Guide.

7

Tools

This chapter explains the tools to use for testing calls and provisioning the service.

Basic SLPIT

Oracle Communications Convergent Charging Controller IN Applications 'talk' a common language – a subset of CS1-INAP known as G8-INAP. Using this common language, IN Applications can be created that focus on functionality without worrying about the lower-level languages spoken by the telephony networks. To cater for the many different protocols and languages implemented in the physical telephony network Interfaces are designed and/or implemented. The interfaces are responsible for communicating with the physical network in whichever protocol the network demands. They translate the messages from the physical network into G8-INAP so that the applications can understand what is going on (The advantage to this approach is that the applications are portable and plug-able to any network, as long as an effective interface can be implemented). The passing of messages back and forth between the applications and the interfaces takes place in the SLEE, where it is possible for many interfaces to be communicating with many applications concurrently. So, where does the SLPIT test tool fit into this picture?

The functional testing of Oracle Communications Convergent Charging Controller applications, using the SLPIT (Service Logic Program Instance Tester), can be done without concern for the protocol of a given network. As long as the application provides the correct functionality in G8-INAP it can be assumed that it will perform the same way on a given network with the appropriate interface(s).

SLPIT is a testing tool which is capable of sending and receiving G8-INAP messages across the SLEE from the application under test. SLPIT communicates with the application through the SLEE, just like a regular interface. It 'receives' messages from a text file script rather than a real network and sends these messages through the SLEE as G8-INAP. It then 'parses' the responses from the application under test, comparing them to the responses expected by the script. From the perspective of the application under test, SLPIT is a real interface converting the network messages to and from G8-INAP.

The main advantage that SLPIT provides is the ability to effectively test IN applications without the need for a physical telephony network, or a low-level network specific test tool.

The main disadvantage with SLPIT is that it is not a real network, and so the quirks and variation between networks and their protocols is not simulated.

SLPIT acts as a TCAP interface to trigger Intelligent Network platform service logic, instead of using a 'real' Service Switching Point (SSP). SLPIT supports the following IN protocols: CAP, MAP, SCCP, GPRS and IS41.

You can view supported protocols using the `-h` option when running SLPIT.

Basic SLPIT Script

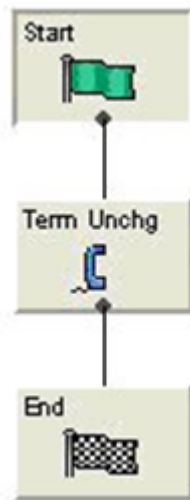
Log in to the SLC `hostname_of_SLC` using the user account: `acs_oper`. This will place you in directory: `//IN/service_packages/ACS`

To test that everything is set up and working, run a basic SLPIT script against the control plan you created in a previous exercise.

1. Create the following script and name it `acs_basic.slp`. Save the script in the scripts directory of your backup area.

```
define call basic { // replace Destination No. (DN) below
with your customer's service numberDN ?= "015111111"CLI ?=
"0139411111"send { initialdp calledpartynumber DN
callingpartynumber CLI callingpartyscategory 10 locationnumber CLI
eventtypebcm analyzedinformation } receive { continue }}startcall
basic using once
```

2. Test the `acs_basic.slp` script against the control plan shown below (you might need to create it and assign it to your customer's service number first).



Test the `acs_basic.slp` script against the control plan shown below (you might need to create it and assign it to your customer's service number first).

Note

The Terminate Unchanged feature node indicates that the call should commence without changing the calling party number.

Your SLPIT script should expect to receive `continue` from the SLC.

3. Place your script in: `/volA|volB/training/your LDAP id/ scripts`

Note

The `calledpartynumber` (Dialed Number) should relate to the service number of your customer. Although this script will succeed even with a non-existent service number, as the default will be to `continue`.

4. Run the script:

- In one terminal window - start 'tailing' the slee acs log file:

```
$ tail -f
/IN/service_packages/ACS/tmp/slee_acs_
```

your LDAP id.log

- In another terminal window, run the SLPIT script. Remember to set your SLEE_FILE if required, then run the script and look at the output:

```
$ slpit -v -k 111 < acs_basic.slp
```

Note

The Service Key (-k) for ACS is 111 and for CCS it is 1. You will be using 111 throughout this workbook (see **SLEE.cfg**)

The outcome should be Successful, for example.:

```
Call success summary: call type basicCalls Run: 1Calls
Succeeded: 1, 100.00%Calls Failed: 0, 0.00%Calls Aborted: 0, 0.00%
Total duration in call processing 0 seconds.Call rate based on that
is Inf CAPS.
```

5. Try running SLPIT without `-v` and also with `> acs_basic.out`, for example:

```
slpit -v -k 111 < acs_basic.slp >
acs_basic.out
```

Note

You will be looking at the contents of a basic script in the next task.

If the script failed:

- a. Create the above control plan and assign it to your customer's service number. Then schedule the control plan to run against your customer's service number.
- b. Alter the calledpartynumber in the SLPIT script to reflect this change and retest.

If you have no luck getting the basic script to work, please see your trainer or mentor, indicating what checks you have already made. Failure to run a SLPIT script could indicate that there is a server configuration problem.

Analyze a Basic Script

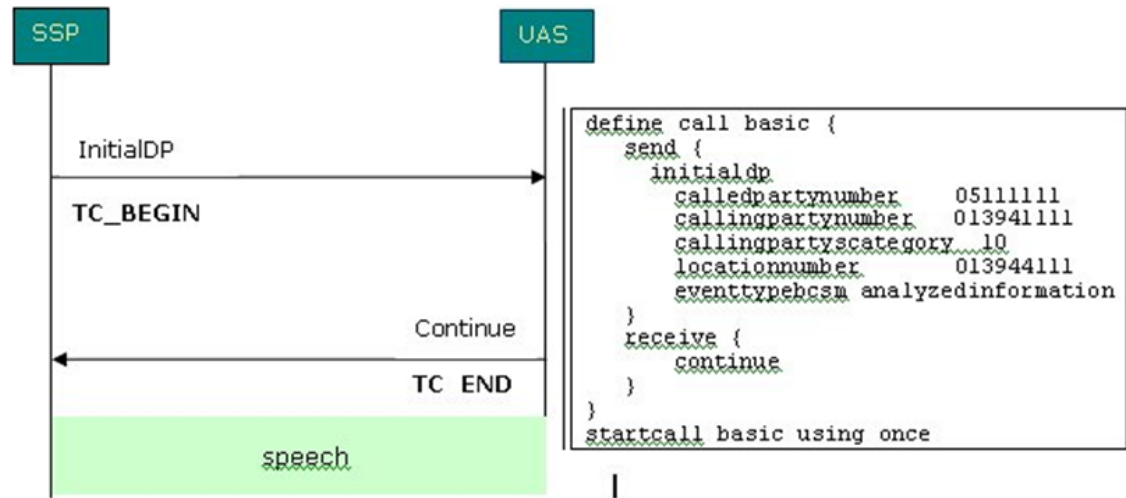
Open your **acs_basic.slp** from you backup area: **/voIA | voIB/training/your_LDAP_id/scripts/**.

The is a very basic script that currently emulates making a call to a service number (setup earlier) and then passes control back to the Service Switching Point (SSP) with no database

lookup or change to the calling number. For this script, the calling party number (CLI) can be anything.

The call represented as a call flow.

Figure 7-1 Call Flow



You will see in one example variables have been defined to hold the called and calling party numbers, but you can also just specify them directly if you choose:

```

...initialdpcalledpartynumber 0151111111
callingpartynumber 0139411111callingpartycategory 10locationnumber
0139411111...
    
```

Remember when testing your scripts, set your SLEE_FILE. Run the script and look at the output:

```

$ cd /volB/training/your_LDAP_id/scripts $ slpit -v -k
111 < acs_basic.slp > acs_basic.out
    
```

Description of the Script acs_basic.slp

This table gives you a description of each part of the script.

Script	Description
define call basic {	This is the start of the script definition and names it basic and uses { to define the start of the call.
DN ?= "0151111111" CLI ?= "0139411111"	DN is the Destination / called phone number and CLI is the Calling Line Identifier. A value assigned using ?= is used to specify default values for a variable. If a value is assigned using =, this will take priority over one with ?=. So DN = "015122222" and DN ?= "015111111". If the first DN was then blanked out, the second DN (?=) would be used.
Send {	Send defines the data to be sent to the IN platform. The start of the send message block is defined using {.

Script	Description
Initialdp	The SLC expects to receive an Initial Detection Point containing called and calling numbers. The InitialDP procedure (with parameters, for example required data) is sent by the service switching function (SSF) after trigger detection point TDP-R in the basic call state model to request for instructions to complete the call. This is referred to as the "Waiting for instructions" state.
Calledpartynumber DN	The calledpartynumber refers to the called party in the forward direction
Callingpartynumber CLI	The callingpartynumber refers to the calling party number signaling information
Callingpartyscategory 10	Callingpartyscategory indicates the type of calling party, for example. operator, pay phone, ordinary subscriber, etc. This information is in standards document Q764E. Category 00000010 = operator, English language. Code 00001101 is used for a 'test call'.
Locationnumber CLI	Locationnumber is used when callingpartynumber does not contain any information about the geographical location of the calling party, for example origin dependent routing when dealing with a mobile subscriber
Eventtypebcsm analyzedinformation }	eventtypebcsm specifies the type of event that is being reported. For analyzedinformation it will contain the calledpartynumber. The end of the send message is defined using }.
Receive {	Receive defines what will be received from the SLC. The start of the receive message block is defined using {.
Continue }	Continue returns control back to the SLC without changing the destination number. Connect would change the dialed number. The end of the receive message block is defined using }.
} startcall basic using once	The end of the call block is defined using }.This runs the script block named basic. Note: Instead of running it once, you can call a script several times using startcall id using uniform delay count, for example startcall basic using uniform 0.5 10 , runs the call every 0.5 seconds 10 times (for delay you must specify a number with a decimal point). Try this if you wish and base your new script on edited version of acs_basic.slp , for example name it acs_basic_delay.slp

SLPIT scripts are based on the sending and receiving of INAP messages. You can find details of each message by searching the INAP standards (for example CS-2, CAMEL).

Refer to the INAP standards document for details of INAP messages, for example initialDP.

SLPIT Script for UATB Node

To test that your product type calls the correct control plan and runs as expected, write a SLPIT script. The SLPIT script will test the Universal Attempt Terminate with Billing node. If you completed Induction Volume 1, you will notice that this script is similar to the script used to test the conditional terminate node. You will also need to add a receive statement to apply charging. See the *SLPIT User's Guide* for more information.

Example Script - your script can be based on the following example:

```
define call ccs_uatb { send {initialdpappContext
"0,4,0,0,1,0,50,1"calledpartynumber
"441394100005"callingpartynumber "441394100009" // change this to
your subscriber's numbercallingpartyscategory 10eventtypebcs
analyzedinformation} // The maxDuration parameter tells the switch
the maximum time permitted before// the next reservation request
should be sent to the SCP receive { applychargingmaxDuration
2400release 0tone 0 requestreportbcsmeventeventtypebcs
oCalledPartyBusy (2)eventtypebcs oNoAnswer (2)dpspecificcriteria
applicationtimer 10eventtypebcs oAbandon (1)eventtypebcs
RouteSelectFailureeventtypebcs oDisconnect monitormode interrupted
(2)eventtypebcs oDisconnect monitormode notifyAndContinue(1)
connect destroutingaddr "441394100005" } send
{eventreportbcsmeventtypebcs oDisconnect miscallinfo request
(1)// The timeNoTariffSwitch parameter tells the SLC how long to
charge for in decisecondsapplychargingreportreceivingSide
2timeNoTariffSwitch 1200callActive 1 } } startcall ccs_uatb using
once
```

Please do not run your script yet. You will first create and check your environment.

Provisioning Interface

The Provisioning Interface (PI) provides a mechanism for manipulating data in the Convergent Charging Controller solution using an API. It enables bulk or scripted changes to SMF data, where it would be inefficient or more prone to errors for an operator to do so using the Java administration screens.

The provisioning interface uses TCP/IP based Unix sockets to receive commands and parameters that are effectively translated into SQL commands to update the application tables on the SMF database. The PI supports the querying, addition, deletion and modification of database records. It is used by many operators to integrate database-driven software with an existing customer care solution or a web front end.

The PI can be used to provide access to the billing engine for customer care operators or by the operator to enable web based self-care for their customers.

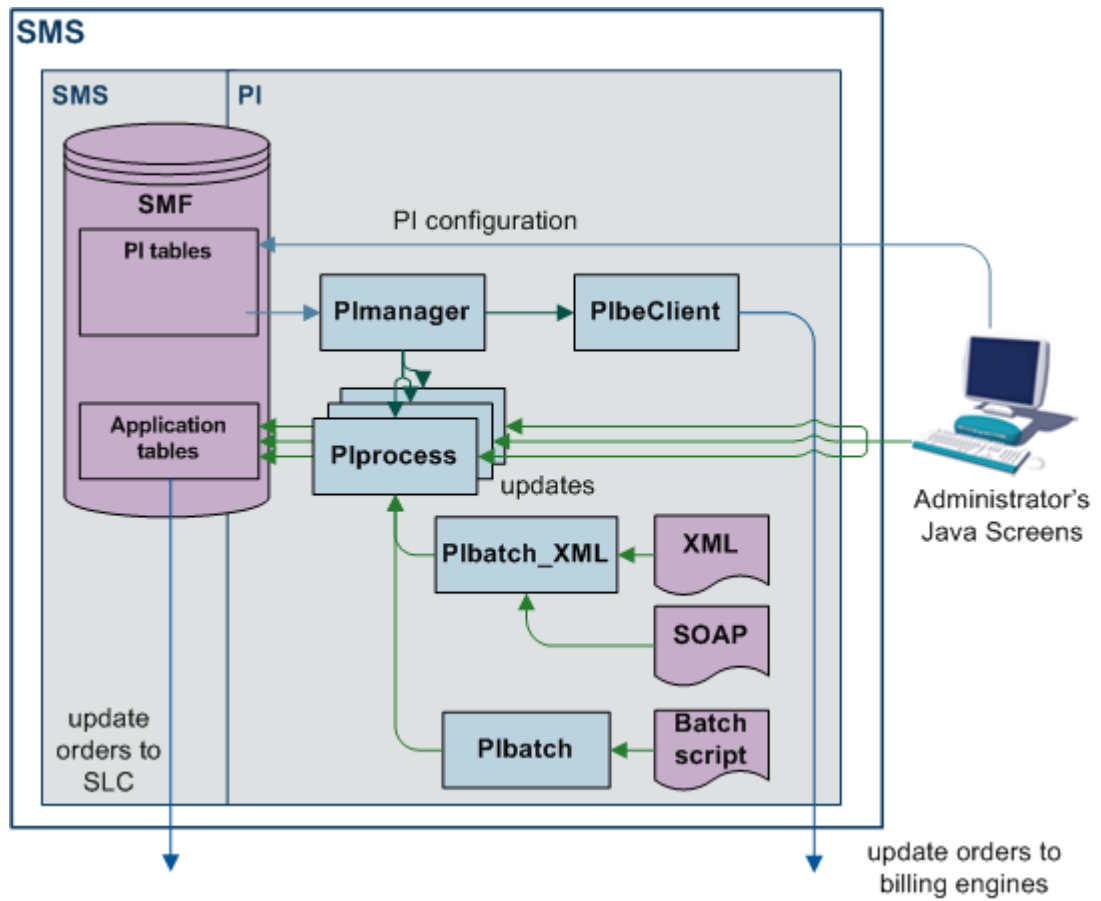
It is a reliable, extensible, network-aware interface based on interoperability standards.

Security on the provisioning interface is achieved by allowing only recognized hosts the ability to connect. For this a username and password is required and checksums are used within the dialogs to ensure messages are not modified or additional messages are not added to the information. The first task you need to perform if you wish to use PI is to configure your PC to access the service. This exercise takes you through the necessary steps in order to set this up.

Component Diagram

This diagram shows the PI components and processes.

Figure 7-2 Component and Processes



Process Descriptions

This table describes the processes involved in the PI application.

Table 7-1 Process Description

Component	Description
Pprocess	Updates the SMF database and the SMS sends the data through replication to the other nodes in the network. Pprocess waits for TCP/IP connections and processes commands sent to it. Commands are processed through a shared library or by running a PL/SQL function in the SMF database.

Table 7-1 (Cont.) Process Description

Component	Description
PIManager	<p>Stops and starts the Pprocess. PImanager is started by init - you should find an entry in /etc/inittab called pim1.</p> <p>The log file for this command can be found at /IN/service_packages/PI/tmp/PImanager.log</p> <p>You can restart PI in two ways:</p> <ul style="list-style-type: none"> • /IN/service_packages/PI/bin/PIreread.sh - re-reads SMF database when all connections have been dropped. • /IN/service_packages/PI/bin/restart.sh - terminates PImanager and all Pprocesses, which are then restarted by /etc/inittab.
PIbatch	<p>Allows multiple PI commands to be sent to Pprocesses through a script file: /IN/service_packages/PI/bin/PIbatch script server</p> <p>Results are placed in a file of the same name but with a .result extension.</p>

Configuring User Access to PI Screens

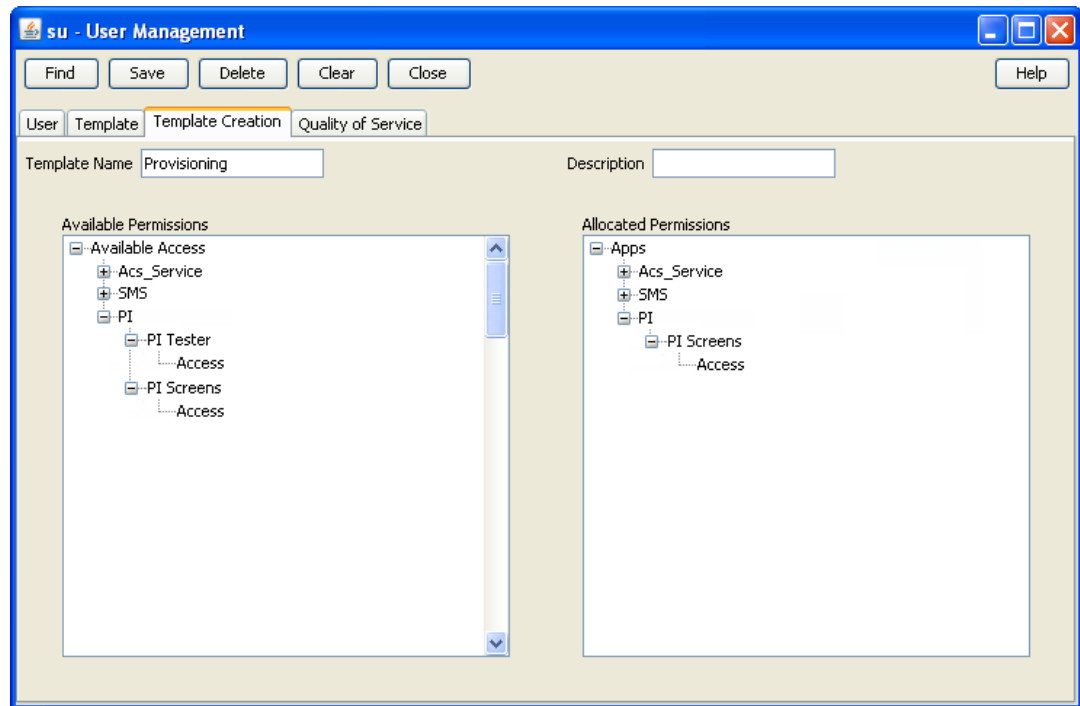
To configure access to PI screens for a specified user:

1. Open the SMS user interface (UI) using **smsGui.bat/smsGui.sh**.
2. Log in to the SMS UI as the systems administrator (for example by logging in as the user **su**).
3. Select **User Management** from the **SMS Operator Functions** menu.
4. Find the user to whom you want to give PI permissions by clicking **Find**, and then **Search**.

✓ Tip

Leave the **User Name** field empty to find all users. To find a specific user, specify the first few letters of the user's name.

5. Select the user you want and click **Close**.
6. On the **Template Creation** tab, select the **PI** group (including **PI Screens** and **PI Tester**) from the list in the **Available Permissions** area and drag and drop them to the **Allocated Permissions** area.



7. Click **Save**.
8. Click **Close**.

The specified user now has the ability to administer the PI through the PI UI.

Configuring User and PC Connection to PI

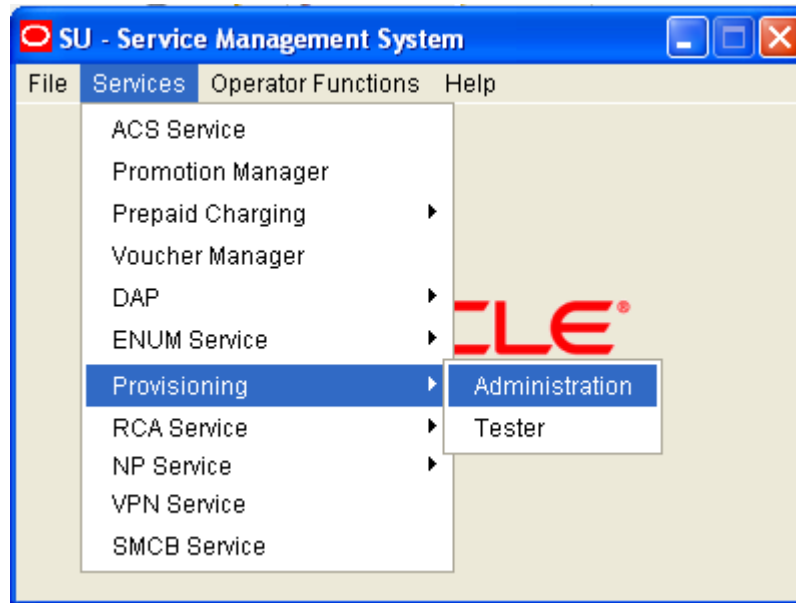
To configure personal computer (PC) connection details to the PI for a specified user.

1. Open the SMS UI, and log in as the user who requires a PC connection to the PI.

Note

The user should already have screens access to the PI configured. For more information, see [Configuring user access to PI screens](#).

2. Select the **Administration** option from the SMS **Services, Provisioning** menu.



- On the **Hosts** tab, click **New**.
The PI Hosts window is displayed.
- In the **IP Address** field, enter the IP address of the PC.

✓ **Tip**

You can find the PC IP address by using the ipconfig command in a CMD window or by entering ifconfig from a UNIX terminal.

- On the **Users** tab, click **New**.
The PI Users window is displayed.

- Enter the SMS user details for the user who will be using the PC. You must allocate security level 99 to the user in the **Security Level** field.

7. Restart the PI to activate the changes by performing the following steps:
 - a. Log in to the SMS as the `smf_oper` user.
 - b. Go to the following directory:

```
1/IN/service_packages/PI/bin
```

- c. Check the owner of the **Pirestart.sh** file and su to this user.
- d. Restart the PI by entering the following command:

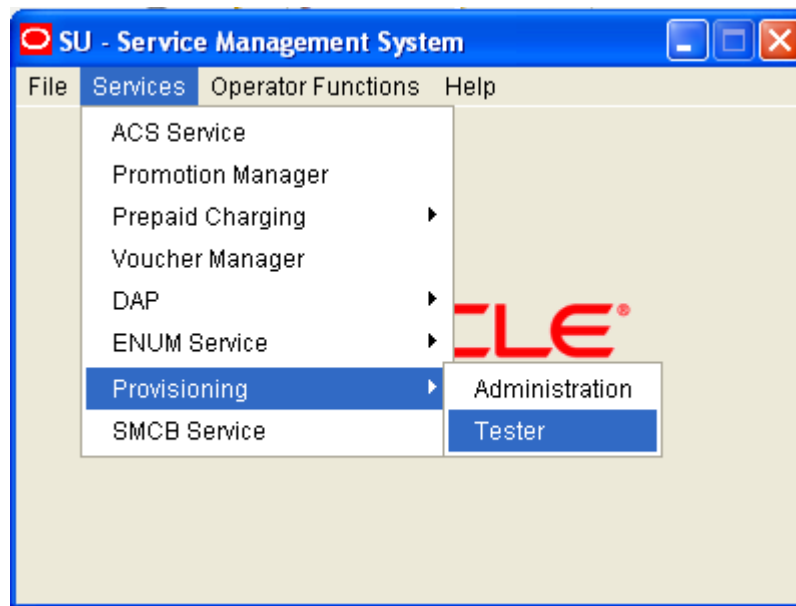
```
1Pirestart.sh
```

Testing User and PC Connection

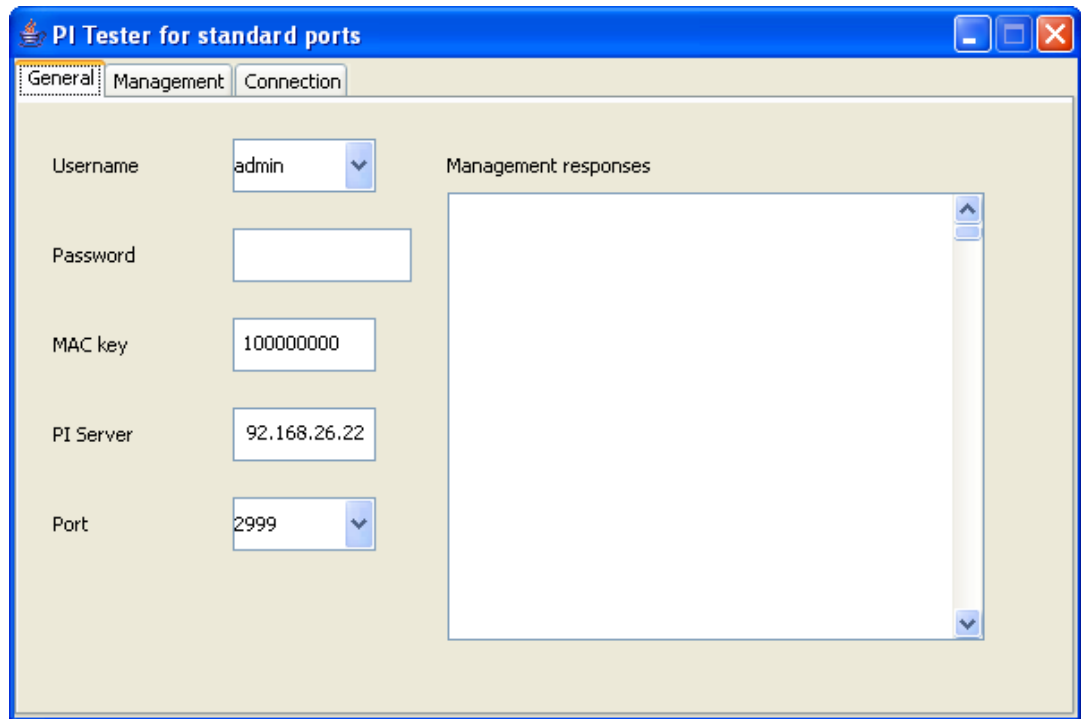
To test your user and PC connection:

1. Give PI a moment to restart, then, through `smGui.bat/smsGui.sh`, log in as your SMS user.

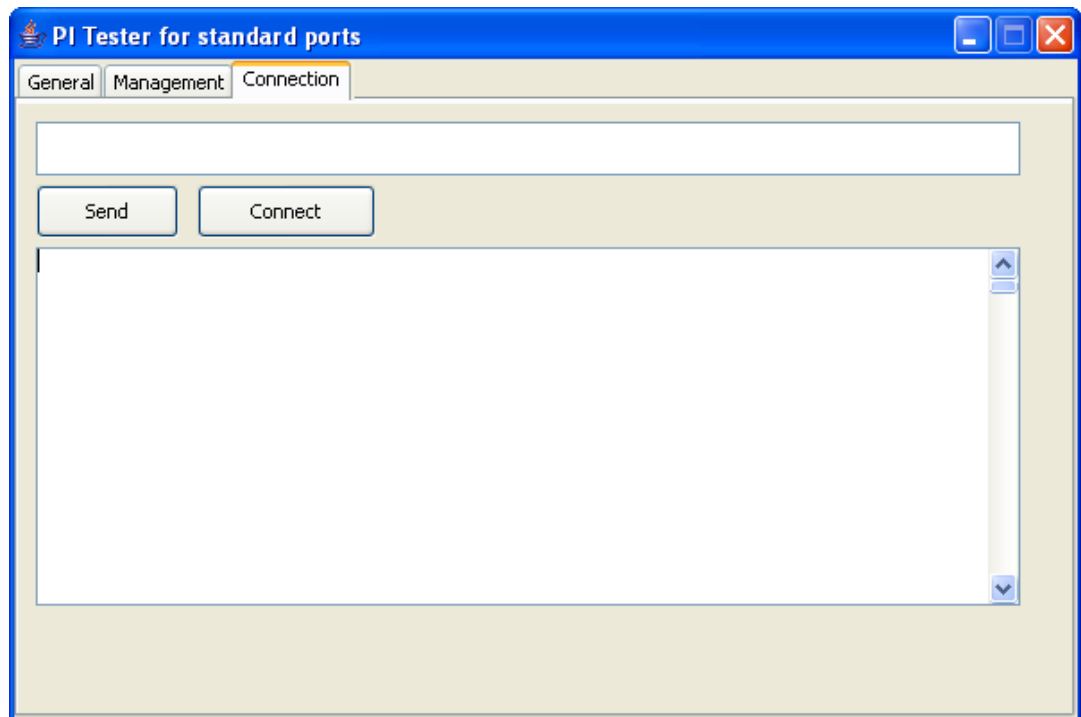
Click **Services**, then **Provisioning**, and then **Tester**.



2. On **General** tab, select your username from the drop-down list and enter the password.



3. On **Connection** tab, click **Connect**.



Result: If connection is:

- Successful the following should be displayed in the Connection window:

```
>username ,*****;
```

```
<ACK ,SYNSTAMP=2007041108352386 ;
```

- Unsuccessful, for example invalid username/password, the following should be displayed in the Connection window:

```
>username,*****;  
<NACK,72-INVALIDLOGON - username,password;
```

ACS Commands

Click **Service Management System**, then **Provisioning** =, and then **Administration** to view the list of PI commands available. You will find these in the **Commands** tab.

A list of ACS PI commands and their expected format can be found at
ProjectDrive:\Provisioning Interface (PI)\PI_v2.6\Design\ACS Product\Commands

PI ACS Commands

Once you have established a connection to the SMS using PI Tester you should be able to run the following ACS commands. The command should be entered into the 'command' window (window above the **Send** and **Connect** buttons). PI processes interrogate the timestamp SYNSTAMP at the end of each command and expect this value to be later than the previous timestamp recorded. Therefore, this SYNSTAMP value has to be manually incremented by '1' each time a new command is sent.

CSAdd CLI

ACSLI=ADD – add a CLI

Use this command to add a new CLI to the ACS customer that the specified control plan belongs to.

When you connect to PI, you will receive a SYNSTAMP. copy and paste this and use it as your SYNSTAMP. For each command you issue, increment the number by 1. See an example of the command entered below:

```
ACSLI=ADD:CLI=01206888888,CALLPLAN=callplan,ALLOWED=1|2|3|4|  
5,SYNSTAMP=2007041108352387;
```

Try sending some commands where the CLI entered does not exist.

Note

It was observed during testing that the only control plans (call plans) that CLIs could be added and related to were those that belonged to the ACS customer 'Boss'.

ACS Query CLI (Calling Line Identifier)

ACSLI=QRY – query an existing CLI, this will return a list of allowed CLIs.

Use this command to perform a database query on one of your ACS customers' CLIs created earlier. See an example of the command entered below:

```
ACSLI=QRY:CLI=01473666666,SYNSTAMP=2007041108352386;
```

Note

If the commands are not present (for some reason), you will get this:

```
<NACK,75-UNKNOWN COMMANDACSLI=QRY;
```

CCS Commands

Query Subscriber

CCSCD1=QRY – query a CCS subscriber

Use this command to perform a database query on one of your CCS subscribers created earlier in this induction book. PI commands have mandatory parameters that must be entered and optional parameters that can be entered to gain more information from the database. For this command the mandatory parameter is the subscriber MSISDN. More information may be returned depending on what applications are loaded on the server for example if piWalletSms is loaded onto the server then details of the subscriber's wallet and balances will also be returned. See an example of the command entered below:

```
CCSCD1=QRY:MSISDN=44123457,SYNSTAMP=2007041213581396;
```

The response should be similar to this:

```
<CCSCD1=QRY:ACK:MSISDN=44123457,ACCOUNT_NUMBER=1044123457,PRODUCT=PT1,SERVICE_PROVIDER=Boss,STATUS=A,CREATION_DATE=20070131175807,WALLET_EXPIRY_DATE=,BALANCE_EXPIRY_DATE=20070904165847,BALANCE=56815,INITIAL_BALANCE=100000,LANGUAGE=english,LAST_RECHARGE_DATE=20070405150304,LAST_CC_RECHARGE_DATE=,LAST_USE_DATE=20070405150230,LAST_RECHARGE_AMOUNT=0,PREV_WALLET_EXPIRY_DATE=,PREV_BALANCE_EXPIRY_DATE=,PREV_BALANCE=0,LAST_EXP_CREDIT=0,TOTAL_EXP_CREDIT=0,LAST_EXP_DATE=20070302231001,FIRST_ACTIVATION_DATE=,LAST_STATE_CHANGE_DATE=20070131180057,LAST_STATE_CHANGE_REASON=,BYPASS_NUMBER=,WALLET_TYPE=Personal,CHARGING_DOMAIN=5,FFD=,FFN=,FDN=,CUG=,CURRENCY=EUR,SYNSTAMP=2007041213581396;
```

Add Friends and Family Number

Ensure your subscriber's product type is configured to have Friends and Family numbers. It is set to 0 by default:

Click **Subscriber Management** and then **Product Type**. Edit the product type you are querying and select Friends and Family from the list presented on the left hand side. Change Maximum F&F numbers allowed to say, 5.

Check that F&F is activated for the subscriber you are adding F&F numbers for.

To add a number to the Friends & Family list, use CCSCD8=ADD

```
CCSCD8=ADD:MSISDN=44123457,FFNUM=01473222222,SYNSTAMP=2007041213581402;
```

If the number is added to the database successfully the response should be similar to this:

```
<CCSCD8=ADD:ACK,SYNSTAMP=2007041213581402;
```

Verify that new number has actually been added to your subscriber's Friends & Family list using the CCS screens and by querying the database. Did you have any problems adding the new number? You could try to delete the number now.

8

Integration with ECE

This chapter explains how to configure Oracle Communications Convergent Charging Controller to integrate with Elastic Charging Engine (ECE).

Control Plan Configuration

When Convergent Charging Controller is integrated with ECE, Convergent Charging Controller forwards calls for external subscribers to ECE. To enable Convergent Charging Controller to process calls for external subscribers successfully, you add the following feature nodes to the beginning of each control plan that will be invoked by ECE subscribers:

- Add the Set Active Domain feature node after the Start feature node in the control plan. You use the Set Active Domain feature node to set the active domain to the ECE domain.
- Add two Set feature nodes after the Set Active Domain feature node. You use the Set feature nodes to populate the Service-Identifier and Rating Group values that are sent in request messages.

Configuring the Set Active Domain Feature Node

To configure a Set Active Domain feature node:

1. In the Control Plan Editor window, open the control plan you want to edit.
2. Add a Set Active Domain feature node to the control plan immediately after the Start feature node.
3. Open the configuration window for the Set Active Domain feature node.
4. In the **Name** field, select the name of the ECE domain; for example, select `ECE_Domain`.
This will set the active domain to the ECE domain.
5. In the Management area, select the **Wallet** option. Ensure that all other Management options are deselected.
6. Click **Save**.
7. Save and close the control plan.

Configuring the ECE Service Identifier and Rating Group

You add two Set feature nodes to control plans to set the Diameter Service Identifier and Diameter Rating Group profile tags to the values that ECE will use to direct calls to the correct service.

You must configure values that match the corresponding values configured in ECE for the service required; for example, if ECE includes the following service configuration in the ECE Diameter Mediation table:

Service-Context-Id |

```
Service-Identifier | Rating-Group | ProductType | EventType Version | ValidFrom  
|"32260@3gpp.org" |
```

```
"1" | "10" | "TelcoGsmTelephony"
```

```
|"EventSessionTelcoGsm" | 1.0 | "2012-12-31T12:01:01" | ***** Account Level *****
```

you would configure Set feature nodes for the service "TelcoGsmTelephony" to set:

- The Diameter Service Identifier profile tag to 1
- The Diameter Rating Group profile tag to 10

To configure the ECE service identifier and rating group:

1. In the Control Plan Editor window, open the control plan you want to edit.
2. Add a Set feature node to the beginning of the control plan, immediately after the Set Active Domain feature node.
3. Open the configuration window for the Set feature node.
4. Set the Diameter Service Identifier profile tag to the ECE service identifier value by doing the following:
 - a. In the **Value** field, enter the ECE service identifier value.
 - b. From the **Target Data Type** list, select `Temporary Storage`.
 - c. From the **Target Location** list, select `Temporary Storage`.
 - d. From the **Target Field** list, select `Diameter Service Identifier`.
5. Click **Save**.
6. Add a second Set feature node immediately after the first Set feature node.
7. Open the configuration window for the second Set feature node.
8. Set the Diameter Rating Group profile tag to the ECE rating group value by doing the following:
 - a. In the **Value** field, enter the ECE rating group value.
 - b. From the **Target Data Type** list, select `Temporary Storage`.
 - c. From the **Target Location** list, select `Temporary Storage`.
 - d. From the **Target Field** list, select `Diameter Rating Group`.
9. Click **Save**.
10. Save and close the control plan.

About Variable Amount Recharge Feature Node Configuration

You use the Variable Amount Recharge feature node to send voucher recharge details to ECE, such as the balance to recharge and the recharge amount. When you configure the Variable Amount Recharge feature node, ignore the wallet-type fields that can be used to set the Convergent Charging Controller wallet type.

If you specify configuration in the wallet-type fields, Convergent Charging Controller includes a `setWallet` request in the recharge message sent to ECE. Because the `setWallet` request is not supported by Diameter, this will result in calls failing.

About CCS Capabilities

CCS capabilities enable calls sent to the same service key to be handled differently depending on the bearer capability in their IDP. For example, voice and video for same service key can have different control plans.

A capability definition includes the default control plan for the capability. You must set the default control plan that will be invoked for external subscribers for each CCS capability. You set the default control plan for a CCS capability in the Prepaid Charging, Service Management window in the Convergent Charging Controller UI. See the discussion of CCS capability configuration in *CCS User's Guide* for more information.

About Convergent Charging Controller Balance-Type Configuration

This section describes the updates required to Convergent Charging Controller balance-type configuration to enable Convergent Charging Controller to send balances to, and receive balances from ECE. This section assumes that the balance types that will be used are already configured in the Convergent Charging Controller UI.

See *CCS User's Guide* for information about configuring balance types in Convergent Charging Controller.

Enabling Balance Values to Pass Between Convergent Charging Controller and ECE

Perform the following updates to balance-type configurations in Convergent Charging Controller to enable balance values to pass between Convergent Charging Controller and ECE:

- Update the balance-type configuration in the `ccsMacroNodes` section of the **eserv.config** file. See *Balance-Type Configuration in eserv.config* .
- Update the subscriber-domain-type configuration in the **eserv.config** file. See *Subscriber-Domain-Type Configuration in eserv.config* .
- Update the DCD-balance-query configuration in the **eserv.config** file. See *Balance-Query Configuration in eserv.config* .
- Configure balance-type mapping between Convergent Charging Controller and ECE. See *Configuring Balance-Type Mapping Between Convergent Charging Controller and ECE* .
- Configure balance-type announcements. See *Configuring Balance-Type Announcements* .
- Configure balance-type translations for balance notifications. *Configuring Balance-Type Translations for Balance Notifications* .
- Enable first-use policies for vouchers. See *Enabling First-Use Balance-Expiry Policies in Convergent Charging Controller* .

Balance-Type Configuration in eserv.config

You use the following feature nodes in control plans to play balance announcements to subscribers:

- Account Status
- Balance Status
- Balance State Branch

To configure which balance types to check for a specified service provider before playing balance announcements, include the following configuration in the CCS, `ccsMacroNodes` section of the **eserv.config** file:

```
CCS = { ccsMacroNodes = { BSAnnBalanceTypes = [
  {acsCustomerId = customer_id balTypeIds = [
    balance_ids] } ] BSBCheckBalanceTypes = [{acsCustomerId =
    customer_id balTypeIds = [ balance_ids] } ] }
}
```

where:

- `BSAnnBalanceTypes` defines the list of service providers and balance types for which announcements can be played by the Account Status and Balance Status feature nodes.
- `BSBCheckBalanceTypes` defines the list of service providers and balance types for which announcements can be played by the Balance State Branch feature node.
- `customer_id` is the ID in the `ACS_CUSTOMER` database table of the service provider. Specify the ID for the service provider you are using for the external subscribers.
- `balance_ids` is the ID in the `CCS_BALANCE_TYPE` database table of the balance type that will be checked for the service provider specified in `acsCustomerId`. To check multiple IDs, use a comma as a separator. For example:

```
balTypeIds =
[15, 16]
```

To specify configuration for multiple service providers, for each additional service provider configure the `acsCustomerId` and `balTypeIds` parameters; for example the following configuration is for two service providers:

```
CCS = { ccsMacroNodes = { BSAnnBalanceTypes = [{acsCustomerId
= customer_id_1 balTypeIds = [ balance_ids] } {
acsCustomerId = customer_id_2 balTypeIds = [
balance_ids] } ] BSBCheckBalanceTypes = [{acsCustomerId =
customer_id_1 balTypeIds = [ balance_ids] }
{acsCustomerId = customer_id_2 balTypeIds = [
balance_ids] }] } }
```

Subscriber-Domain-Type Configuration in `eserv.config`

To enable the Balance State Branch feature node to work correctly without the need for an additional Set Active Domain feature node in the control plan, include the following configuration in the CCS, `ccsServiceLibrary` section of the **eserv.config** file on the SLC node:

```
CCS = {ccsServiceLibrary = { SubscriberDomainType = 1 }
}
```

where the `SubscriberDomainType` parameter value is the ID of the VWS domain set in the `DOMAIN_TYPE_ID` field in the `CCS_DOMAIN` database table. The default value for the VWS domain ID is 1 (one).

Balance-Query Configuration in `eserv.config`

DCD supports `BALANCE_ENQUIRY` Diameter message requests. You must configure DCD to send `BALANCE_ENQUIRY` Diameter message requests using the balance-check method. The balance-check method triggers balance queries based on a balance-check message with a service identifier that is set to "Information".

Configure the balance-check method in DCD by including the following configuration in the `DomainTypes` section of the `eserv.config` file on the SLC node:

```
DomainTypes = [{ balanceEnquiryMethod = "balanceCheck"
}]
```

See *DCD Technical Guide* for more information about DCD configuration.

Configuring Balance-Type Mapping Between Convergent Charging Controller and ECE

Convergent Charging Controller and ECE store balances in their systems with different balance types and currency units. To send and receive balances between each product, map Convergent Charging Controller balance types to their equivalent ECE balance IDs. For example, map the Convergent Charging Controller balance type for currency balances to its equivalent ECE balance ID, 840.

You map a Convergent Charging Controller balance type to an ECE balance ID by specifying:

- The Convergent Charging Controller balance type name

The equivalent ECE balance ID

- The Diameter domain
- The conversion scale

The conversion scale is a number that converts balance values between Convergent Charging Controller and ECE. For example, if ECE stores 1 British pound as 1.0 and Convergent Charging Controller stores 1 British pound as 100 (1 British pound equals 100 pence), the conversion scale must be set to 100 to multiply the incoming ECE British pounds value to the Convergent Charging Controller pounds value, denominated in pence.

To configure a balance-type mapping between Convergent Charging Controller and ECE:

1. Log in to the SMS UI.
2. From the **Services** menu, select **Prepaid Charging**, then **Service Management**.
The Service Management screen is displayed.
3. From the **Service Provider** list, select the service provider you are using for external subscribers.
4. Click the **Balance Type Mapping** tab.
5. Click **New**.

The New Balance Type Mapping dialog box is displayed.

6. From the **Balance Type** list, select name of the Convergent Charging Controller balance type.
7. In the **Third-Party Resource** field, enter the equivalent ECE balance ID.
8. In the **Third-Party Resource Scale** field, enter the conversion scale to apply between Convergent Charging Controller and ECE balances.

For example, to convert Convergent Charging Controller dollar balances held in cents to ECE dollar balances held in dollars, set this field to 100. In this case, Convergent Charging Controller balances are divided by 100 before they are sent to ECE. That is, an Convergent Charging Controller balance of 500 cents is divided by 100 to convert it into an ECE balance of 5 dollars.

9. From the **Domain Type** list, select the Diameter domain.
10. Click **Save**.
11. Open the **eserv.config** file in a text editor.
12. In the DCD section, set the `conversionScale` parameter to 0:

```
1DIAMETER = { DomainTypes = [ AVPs = [ {
  conversionScale = 0 } ] ] }
```

This configures DCD to use the conversion scale specified in the **Balance Type Mapping** tab on appropriate balance-type AVPs.

13. Save and close the file.

See the discussion about service management in *CCS User's Guide* for more information.

Configuring Balance-Type Announcements

To configure which announcements to play to subscribers for balance expiries:

Note

This procedure assumes that ACS announcements are already configured. See the discussion of configuring announcements in *ACS User's Guide* for more information about configuring announcements in ACS.

1. In the SMS UI, open the Prepaid Charging, Wallet Management window.
2. From the **Service Provider** list, select the service provider you are using for external subscribers.
3. Click the **Balance Type** tab.
4. From the list of balance types on the tab, select the balance type for which you want to configure announcements.
5. Click **Edit**. The Edit Balance Type screen appears.
6. In the Edit Balance Type screen, select the **Balance Type Announcements** tab.
7. From the available announcement lists, select the announcements to play subscribers for balance expiries. For each type of announcement, do the following:
 - a. From the **Announcement Set** list, select the announcement set that holds the announcement you want to play.

- b. From the **Announcement** list, select the announcement to play.
8. Currency-to-announcement mappings define, for cash balance types only, the announcements to play when the subscriber's balance is positive or negative and the currencies in which the announcements can be played.

If this is a cash balance, add or edit currency mappings as required. See the discussion of balance-type-announcement configuration in CCS User's Guide for more information.

9. Click **Save**. The balance-type-announcement configuration is saved to the database.

Configuring Balance-Type Translations for Balance Notifications

You configure Convergent Charging Controller to send balance-type notifications in the subscriber's language and currency by configuring balance-type translations for the balance information contained in notifications.

To configure a balance-type translation for a balance notification:

1. In the SMS UI, open the Prepaid Charging, Wallet Management window.
2. From the **Service Provider** list, select the service provider you are using for external subscribers.
3. Select the **Balance Type Translations** tab.
4. If you want to:
 - Add a balance-type translation, click **New**. The New Balance Type Translation dialog box appears.
 - Edit a balance-type translation, select the balance-type translation on the tab and click **Edit**. The Edit Balance Type Translation dialog box appears.
5. Configure the balance-type translation by specifying values in the available fields. See *Balance Type Translations Fields* for information about the available fields.
6. Click **Save**.

Balance Type Translations Fields

This table describes the balance-type translation fields.

Table 8-1 Balance Type Translations Fields

Field	Description
Language	The language the short message is being translated to. This field cannot be changed after the record is first saved. This list is populated by the Language tab on the ACS Tools screen. For more information about configuring languages, see <i>ACS User's Guide</i> .
Balance Type	The balance type this translation will be for. This list is populated by the records configured in the Balance Types tab. This field cannot be changed after the record is first saved. Tip: You can create only one balance-type translation for each language and balance-type combination.
Single Balance Currency Type	If required, you can select a different single balance currency type to use for the balance type. Note: This field is available only if the Balance Type field is set to a balance type that has a unit of cash.

Table 8-1 (Cont.) Balance Type Translations Fields

Field	Description
Dual Balance Currency Type	<p>If required, you can select a different dual-balance currency type to use for the balance type.</p> <p>Note: This field is available only if the Balance Type field is set to a balance type that has a unit of cash.</p>
Translated Name	<p>The translated balance-type name to send to the subscriber in short messages. If nothing is entered in this field, this part of the message will not be sent.</p> <p>Tip: The maximum number of characters allowed is 20.</p>
Single Format for Balance	<p>The structure of the data to send to the subscriber in short messages for a single (or first) currency.</p> <ul style="list-style-type: none"> For balance types with a cash unit, the structure of the data to send to the subscriber in short messages when the balance is in one currency. The subscriber's actual details are inserted into the variables. The variables must include the following, in order: <ul style="list-style-type: none"> <code>%s %d %c</code> <code>%02d</code> <p>Where:</p> <ul style="list-style-type: none"> – <code>%s</code> = Currency Code (example, AUD) – <code>%d</code> = Main unit of the currency (example, \$) – <code>%c</code> = Separator (example, .) – <code>%02d</code> = Subunit of the currency (example, c) <p>For balance types with non-cash units (for example, time balances are in seconds), you may only include the <code>%d</code> variable in the text in the Single Format for Balance field.</p> For balance types with non-cash units (for example, time balances are in seconds), you may only include the <code>%d</code> variable in the text in the Single Format for Balance field. <p>For balance type unit, <code>%d</code> variable is replaced by:</p> <ul style="list-style-type: none"> – Time: Time balance in seconds – SMS: Number of named events – Bad PIN: Number of bad PIN attempts during the last 24 hours.
Dual Format for Balance	<p>The structure of the data to send to the subscriber in short messages for the second currency.</p> <p>Note: This field is not available for non-cash balance types or balance types that have only one currency.</p>
Separator	<p>The character to use to separate the different balances when put together in an SMS Notification.</p> <p>Example: If you enter a comma (,) and there are two balances, General Cash and Promotional Cash, the short message sent in response to a balance inquiry looks like this:</p> <pre>General Cash balance \$20.00 , Promotional Cash balance \$10.00 .</pre>

Table 8-1 (Cont.) Balance Type Translations Fields

Field	Description
Balance Expiry	The balance-expiry text. Warning: You must include '%d' in the text. This variable will be converted to the number of days remaining until the balance expires.
No Balance Expiry	The text to output for balances that have no expiry date.

Enabling First-Use Balance-Expiry Policies in Convergent Charging Controller

A first-use balance is a balance for a subscriber in which the balance-expiry date is offset from the date and time that the balance is first used. Convergent Charging Controller supports the following first-use balance-expiry policy options for voucher types:

- First Use With Offset. The expiry date is based on the expiry offset value specified in the voucher-type configuration.
- First Use – Account Cycle. The expiry date is based on the ECE Account Cycle value.
- First Use – Bill Cycle. The expiry date is based on the ECE Billing Cycle value.

For each of these options, the balance start date and time is set to the date and time that the balance is first used. DCD passes the first-use values to ECE in the ORA-First-Usage-Validity AVP.

You specify the first-use balance-expiry policy for a voucher type in the multi-balance configuration in the New Voucher Type and Edit Voucher Type windows in the Convergent Charging Controller UI.

First-use expiry policy options are disabled in the Convergent Charging Controller UI by default. To enable first-use expiry policy options, you set the `jnlp.ECEExtensions` Java application property to true in the `smsGui.bat/smsGui.sh` file:

```
-Djnlp.ECEExtensions = "true"
```

About DCD AVP Configuration

This section describes the changes to make to DCD AVP configuration to enable Convergent Charging Controller to integrate with ECE. You configure DCD AVPs in the DCD, DomainTypes, and AVPs section of the `eserv.config` file on SLC nodes. See the discussion about AVP configuration in *DCD Technical Guide* for more information.

Configuring DCD AVPs to Integrate Convergent Charging Controller with ECE

Perform the following updates to DCD AVP configuration:

- Configure sign inversion for AVPs of type Integer32 or Integer64. See *About Sign Inversion*.
- Configure DCD to exclude ORA-Validity-End-Relative AVPs with the value 0 (zero). See *Excluding AVPs with Zero Values in Message Requests*.

- Configure Currency-Code AVPs to set the system and user currencies that will be used for playing announcements. See *Currency Code Configuration for Playing Announcements* .
- Configure mapping between Convergent Charging Controller voucher expiry periods and ECE expiry periods. See *Mapping Convergent Charging Controller Voucher-Expiry Periods to ECE Values* .
- Configure ORA-Account-Topup AVPs to top up multiple balances. See *Voucher Top-Up Configuration for Multiple Balance Types* .
- Configure ORA-Balance AVPs to handle balance queries that return multiple balances in request messages. See *Balance Query Response Configuration for Multiple Balances* .
- Configure AVPs to prevent balance queries from returning invalid values. See *Preventing Balance Queries from Returning Invalid Values* .
- Configure ORA-First-Usage-Validity AVPs. See *Configuring AVPs to Support First Use* .
- Configure the ORA-Extend-Bucket-Validity AVP to send the correct new bucket information to ECE. See *New Bucket AVP Configuration* .

About Sign Inversion

For pre-paid services, Convergent Charging Controller and ECE store balance values from different perspectives; positive balance values in Convergent Charging Controller are stored as negative balance values in ECE and negative balance values in Convergent Charging Controller are stored as positive balance values in ECE.

To configure Convergent Charging Controller to convert positive integer values to negative values (and vice versa) for both inbound and outbound values, you set the `signInversion` parameter to **true** in the DCD AVP configuration in the **eserv.config** file.

When `signInversion` is set to **true** for an AVP of type Integer32 or Integer64:

- An outbound positive DCD concept value is converted to a negative value in the AVP.
- An outbound negative DCD concept value is converted to a positive value in the AVP.
- An inbound positive AVP value is converted to a negative DCD concept value.
- An inbound negative AVP value is converted to a positive DCD concept value.

For example, you configure sign inversion in the Value-Digits AVP as follows:

```
{name = "Value-Digits"avpCode = 12003ccsConcept =
"voucherInfoValue"type = "Integer32"signInversion =
trueconversionScale = 0}
```

Excluding AVPs with Zero Values in Message Requests

DCD can send the ORA-Validity-End-Relative AVP in outgoing message requests to ECE for vouchers with multiple balance entries. If a balance-expiry-period value is 0 (zero) — for example, if the expiry value has not been changed from the default value — the message request sent to ECE will fail.

To exclude ORA-Validity-End-Relative AVPs with the value 0 in message requests, set the `excludeWhenIn` parameter to 0 in the DCD AVP configuration:

```
{avpCode = 218name =
"ORA-Validity-Offset-End-Relative"ccsConcept =
```

```
"voucherInfoBalanceExpiryExtension" type = "Unsigned32" excludeWhenIn
= "0" }
```

where the `excludeWhenIn` parameter specifies to exclude the ORA-Validity-Offset-End-Relative AVP from the DCD outgoing message when the value for the specified CCS concept is 0 .

Currency Code Configuration for Playing Announcements

Convergent Charging Controller feature nodes that play announcements, such as the Account Status feature node, derive the announcement to play using the system and user currency values stored in the `walletInfoSystemCurrency` and `walletInfoUserCurrency` CCS concepts. Because ECE does not return the currency code in CCA response messages, values for these CCS concepts are not populated by default and therefore the announcements cannot be played.

To enable Convergent Charging Controller feature nodes to play announcements following a CCA response message, you set the currency code for the system and user currencies to a specific value by adding the following AVP configuration for each type of CCA response message:

1. Add the Currency-Code AVP to the ORA-Balance-Element AVP for the CCA response message.
2. Configure the Currency-Code AVP to set the values for the `walletInfoSystemCurrency` and `walletInfoUserCurrency` CCS concepts to a specific currency.

The following example configuration shows the configuration for the ORA-Balance-Element AVP for balance top-ups. The Currency-Code-Topup AVP has been added to the list of mandatory contents, and its value has been set to 1 (one) in the Currency-Code-Topup AVP. This sets the system and user currency for balance top-ups to euros.

```
{avpCode = 243 name = "ORA-Balance-Element" type =
"Grouped" mandatoryContents = ["ORA-Balance-Element-Id" #
"ORA-Balance-Item" # Not currently supported in Convergent Charging
Controller ccsConcepts "Unit-Value" "Currency-Code-Topup" ] {avpCode =
425 name = "Currency-Code-Topup" ccsConcept =
"walletInfoUserCurrency,walletInfoSystemCurrency" type =
"Unsigned32" value = "1" # set User & System Currency to allow
announcements to be played for topups }
```

Mapping Convergent Charging Controller Voucher-Expiry Periods to ECE Values

Convergent Charging Controller and ECE use different values for voucher-expiry-period units. You configure how to map Convergent Charging Controller expiry units to ECE expiry units in ORA-Validity-Unit AVPs:

```
{avpCode = 219 name = "ORA-Validity-Unit" ccsConcept =
"voucherConcept" type = "Enumerated" conversion = [{esg = CCC_unit
vendor = ECE_unit } ] }
```

where:

- `voucherConcept` is the CCS concept for the voucher expiry data that will be mapped, such as the `voucherInfoBalanceExpiryExtensionType` or the `voucherInfoBalanceValidityType`
- `CCC_unit` is specified in the `esg` parameter and defines the Convergent Charging Controller expiry unit type to map; for example, 1 for months, 2 for hours
- `ECE_unit` is specified in the `vendor` parameter and defines the corresponding ECE expiry unit type; for example, 4 for months, 2 for hours

Specify a combination of the `esg` and `vendor` parameters for each expiry unit type you want to map. For example, the following ORA-Validity-Unit-End-Relative AVP configuration maps the expiry units used in Convergent Charging Controller for months and hours to the values used in ECE for months and hours:

```
{avpCode = 219name =
"ORA-Validity-Unit-End-Relative"ccsConcept =
"voucherInfoBalanceExpiryExtensionType"type =
"Enumerated"conversion = [{esg = 1 # CCC Expiry Unit Monthsvendor =
4 # ECE Expiry Unit Months}{esg = 0 # CCC Expiry Unit Hoursvendor =
2 # ECE Expiry Unit Hours}]}
```

Voucher Top-Up Configuration for Multiple Balance Types

You use the ORA-Account-Topup AVP to top up the balances for a voucher. If you are using voucher types for which multiple balance types have been configured, you can configure the ORA-Account-Topup AVP to top up all the balances at the same time.

To configure the ORA-Account-Topup AVP to apply balance top-ups to multiple balances:

1. Add the ORA-Balance AVP to the mandatory contents.
2. For each additional balance type, add an ORA-Balance AVP to the optional contents.

For example, the following ORA-Account-Topup AVP shows the configuration required to top up three different balances:

```
{avpCode = 206name = "ORA-Account-Topup"type =
"Grouped"mandatoryContents =
["ORA-Recharge-Reference" "ORA-Balance"]optionalContents =
["ORA-Balance" "ORA-Balance" ]}
```

If you do not add the extra optional components, only the first balance is sent in the request message.

Balance Query Response Configuration for Multiple Balances

The request message for balance queries can return multiple balances. To enable DCD to process additional balances, you set the `repeating` parameter to **true** in the ORA-Balance AVP:

```
{avpCode = 208name = "ORA-Balance"type = "Grouped"repeating =
truemandatoryContents =
["ORA-Balance-Element-Id-Topup" "Unit-Value-Topup" "Service-Identifier" "Rating-
Group" ]}
```

Preventing Balance Queries from Returning Invalid Values

To enable the balance values returned in balance-query responses to be processed successfully, remove the Exponent AVP from the ORA-Credit-Floor and ORA-Credit-Ceiling AVPs.

If you include the Exponent AVP in ORA-Credit-Floor and ORA-Credit Ceiling AVPs, Diameter processing returns invalid balance values in balance-query response messages; for example, by creating extra balance types with the ID 0 .

Configuring AVPs to Support First Use

You can configure voucher types to use the following first-use expiry policies:

- First Use With Offset. The expiry date is based on the expiry offset value configured for the voucher type that is stored in the voucherInfoBalanceExpiryExtensionType CCS concept
- First Use – Account Cycle. The expiry date is based on the ECE Account Cycle value that is stored in the voucherInfoBalanceExpiryExtensionPolicy CCS concept
- First Use – Bill Cycle. The expiry date is based on the ECE Billing Cycle value that is stored in the voucherInfoBalanceExpiryExtensionPolicy CCS concept

See *Enabling First-Use Balance-Expiry Policies in Convergent Charging Controller* for more information about enabling first-use expiry policies for a voucher type.

To extract the correct first-use values for a voucher and pass them on to ECE, configure the following AVPs:

```
{name = "ORA-First-Usage-Validity"avpCode = 217type =
"Grouped"excludeWhenEmpty = truemandatoryContents =
[]optionalContents =
["ORA-Validity-First-Use-Cycle" "ORA-Validity-First-Use-With-Offset"]}##
Intentionally blank avpCodename =
"ORA-Validity-First-Use-With-Offset"mandatoryContents =
["ORA-Validity-Offset-First-Use" "ORA-Validity-Unit-First-Use"]{avpCode
= 218name = "ORA-Validity-Offset-First-Use"ccsConcept =
"voucherInfoBalanceFirstUse"type = "Unsigned32"excludeWhenIn =
"0"}{avpCode = 219name = "ORA-Validity-Unit-First-Use"ccsConcept =
"voucherInfoBalanceExpiryExtensionType"type = "Enumerated"}##
Intentionally blank avpCodename =
"ORA-Validity-First-Use-Cycle"mandatoryContents =
["ORA-Validity-Type-First-Use-Cycle"]{avpCode = 219name =
"ORA-Validity-Type-First-Use-Cycle"ccsConcept =
"voucherInfoBalanceExpiryExtensionPolicy"type = "Enumerated"# 6 -
firstAccountCycle# 7 - firstBillCycleincludeWhenIn =
"6,7"conversion = [{esg = 6 # CCC Expiry Policy - First Use Account
Cyclevendor = 5 # ECE - First Use Account Cycle}{esg = 7 # CCC
Expiry Policy - First Use Bill Cyclevendor = 6 # ECE - First Use
Bill Cycle}]}
```

New Bucket AVP Configuration

The concept of when a new bucket is created is reversed between Convergent Charging Controller and ECE. In Convergent Charging Controller, you select whether to create new buckets in the Edit Multi-balance screen for a voucher type. If you selected to create new

buckets, Convergent Charging Controller sets the voucherInfoNewBucket CCS concept value to 1, otherwise Convergent Charging Controller sets the value to 0.

DCD sends the voucherInfoNewBucket value to ECE in ORA-Extend-Bucket-Validity AVPs. Because ECE interprets the value 0 as "create new bucket" and the value 1 as "use existing bucket", you must configure to change the value sent in the AVP to the reversed value:

```
{avpCode = 228name = "ORA-Extend-Bucket-Validity"ccsConcept =  
"voucherInfoNewBucket"type = "Enumerated"conversion = [{esg =  
0vendor = 1}{esg = 1vendor = 0}]}
```

where:

- The `esg` parameter defines the Convergent Charging Controller value; 0 means use existing bucket, 1 means create new bucket
- The `vendor` parameter defines the ECE value; 1 means use existing bucket, 0 means create new bucket

Sample DCD Configuration File

The **eserv.config.dcd.ece.example** file contains an example DCD configuration for integrating Convergent Charging Controller with ECE. It includes the following sections of the **eserv.config** file:

- DCD
- DomainTypes
- Domains
- AVPs

The example configuration file is located in the following directory:

//IN/service_packages/DCD/etc

Use the **eserv.config.dcd.ece.example** file for reference when you update DCD configuration in the **eserv.config** file on SLC nodes.