Oracle® Communications Convergent Charging Controller

Configuration User's Guide

Release 6.0

May 2016

ORACLE®

Copyright

Contents

About This Document

Scope

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.

Related Documents

The following documents are related to this document:

- *Installation Guide*
- *System Administrator's Guide*

Prerequisites

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.

Document Conventions

Typographical Conventions

The following terms and typographical conventions are used in the Oracle Communications Convergent Charging Controller documentation.

Formatting Convention	Type of Information
Special Bold	Items you must select, such as names of tabs. Names of database tables and fields.
Italics	Name of a document, chapter, topic or other publication. Emphasis within text.
Button	The name of a button to click or a key to press. **Example:** To close the window, either click **Close**, or press **Esc**.
Key+Key	Key combinations for which the user must press and hold down one key and then press another. Example: **Ctrl+P** or **Alt+F4**.
`Monospace`	Examples of code or standard output.
`Monospace Bold`	Text that you must enter.
variable	Used to indicate variables or text that should be replaced with an actual value.
menu option > menu option >	Used to indicate the cascading menu option to be selected. Example: **Operator Functions > Report Functions**
hypertext link	Used to indicate a hypertext link.

Specialized terms and acronyms are defined in the glossary at the end of this guide.

Chapter 1

Introduction to Convergent Charging Controller Configuration

Overview

Introduction

This guide is written to give the reader 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.

In this chapter

This chapter contains the following topics.

Service Management System

Accessing SMS

Follow these steps to launch Service Management System using Java Webstart. You can use this process to install a shortcut to the SMS on your desktop.

Note: To launch GUI applications using Java Webstart, you must ensure that the Web server supports the **jnlp** file type. For more information, see *Setting up the Screens* in *Service Management System Technical Guide*.

Step	Action
1	Using an Internet browser, open the SMS Webstart by using one of the following methods: a. Open the Service Management System default page on the *SMS_hostname*, then click the WebStart link. b. Open SMS Webstart directly. The address is in the format: http://*SMS_hostname*/**sms.jnlp** Where S*MS_hostname* is the hostname of the SMS or cluster which is running the SMS application. **Result:** You see the Opening **sms.jnlp** download screen.
2	Select **Open with** and click **OK**. **Result:** The following two windows open: a. The SMS - *SMS_hostname* window, for example:

Step	Action

b. The SMS Login window.

See *Logging on to SMS* (on page 2).

Note: When launching SMS for the first time using Webstart, a shortcut icon is downloaded and displayed on the Desktop.

This enables you to open the SMS GUI directly by double-clicking the shortcut icon. The icon is removed every time you clear the system cache and downloads again when launching SMS through Webstart.

Logging on to SMS

Follow these steps to log on to SMS.

Step	Action
1	On the SMS Login screen:

2 Enter the account details for the default super-user. The:

- User name is `su`.
- Password is usually `ssob` on freshly installed servers.

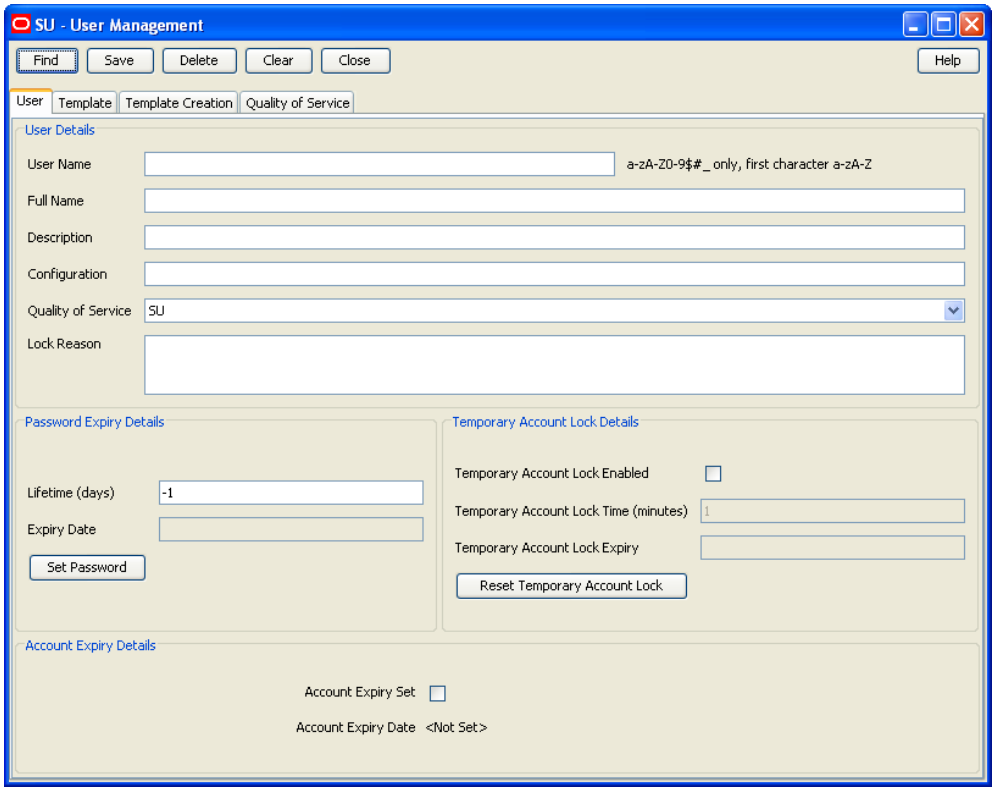
Note: Please take care entering these account details as three incorrect attempts will lock out all other users. You will use your own user account as soon as you have created one. If you manage to lock the su account, please see your trainer or technical mentor to show you how to unlock it.

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.

Follow these steps to add the SMS user.

Step	Action
1	Log in to the SMS server as `su`. See Accessing SMS.
2	Select **Operator Functions > User Management**. **Result:** The User Management screen, **User** tab appears.

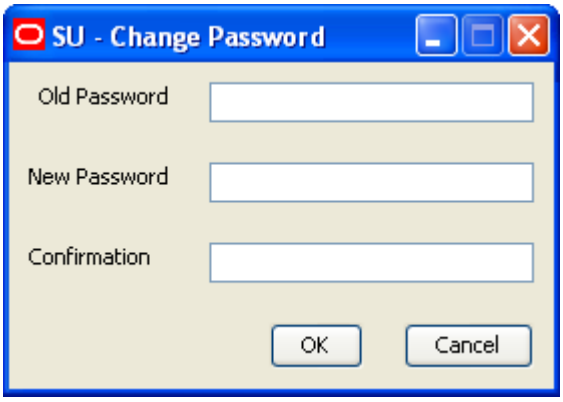


3 Enter your details.

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.

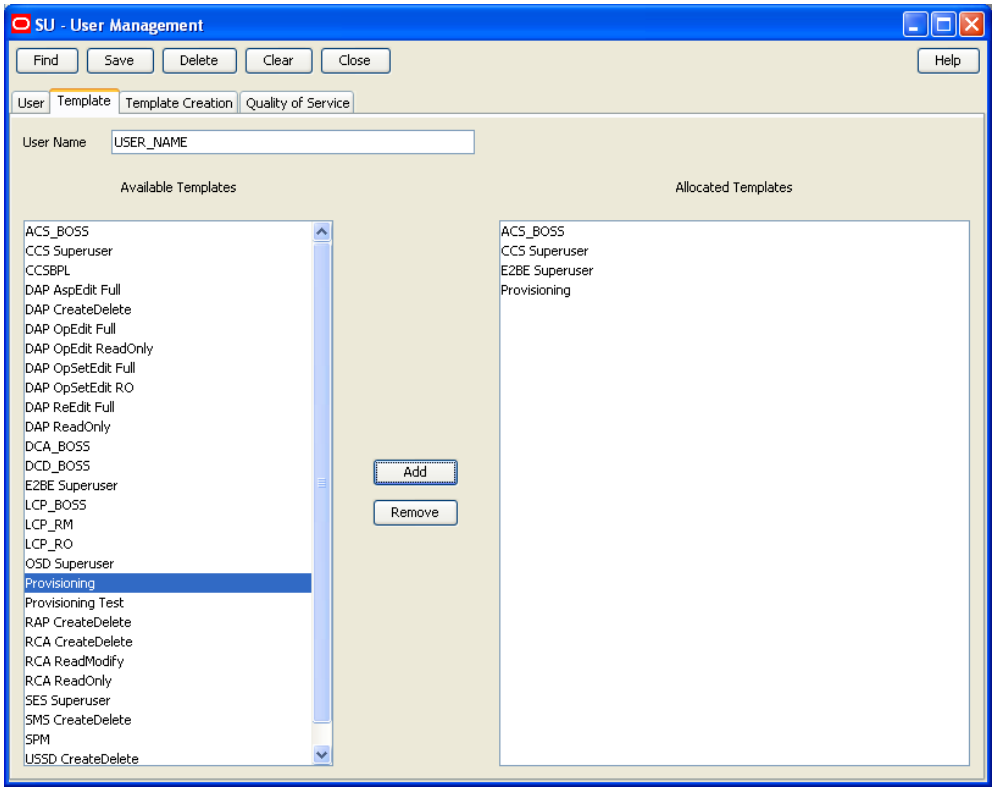
Step	Action
	
5	Set the user password. **Note:** Please do not change passwords for any users not created by you.
6	Click **OK**.
7	Remember to **Save** and then **Close**.
8	Close the browser to log out of SMS. When you logout of SMS, you will find you can no longer log in as another user without restarting 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

Follow these steps to assign services to your user.

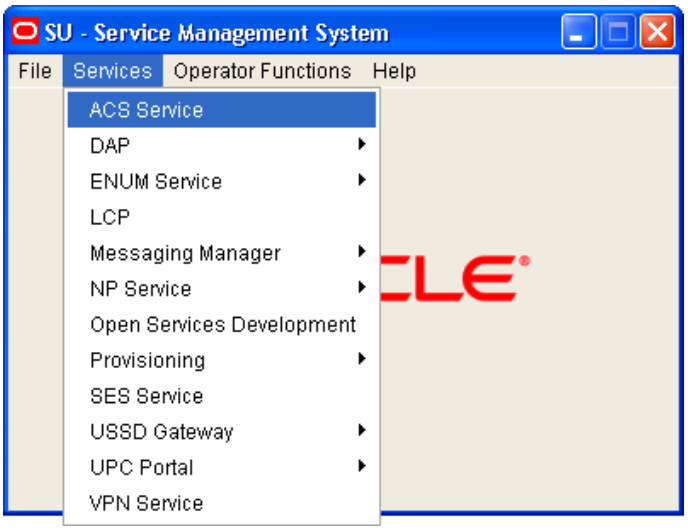
Step	Action
1	First log out of SMS, then log in again as the super user (`su`). Your user does not have the right to assign services.
2	Select **Operator Functions -> User Management**, then click **Find**. **Result:** You see the SU – Find User screen.
3	Type your user's name (in whole or part) into the **User Name** field and click **Search**. **Result:** The Search screen will display all users matching your search terms.

Step	Action
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.

Step	Action



Create a Control Plan

About control plans

Control plans define how calls are processed. The functionality for creating control plans is available in the ACS Control Plan Editor (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 on using the ACS Control Plan Editor, see *CPE User's Guide*.

Creating a Basic Control Plan

Follow these steps to create a basic control plan that contains only the Start and End feature nodes.

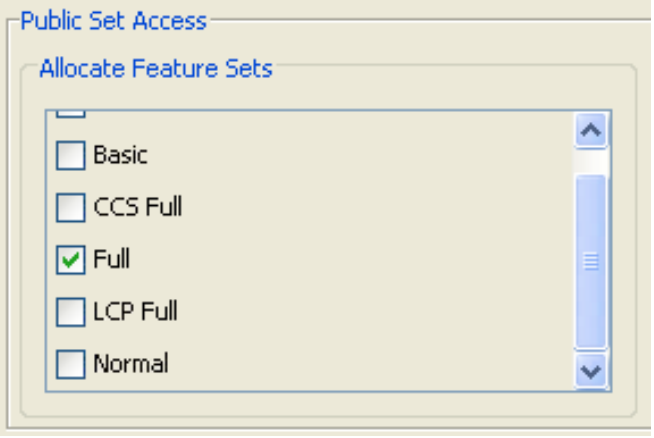
Step	Action
1	On the Service Management System screen, select **ACS Service** from the **Services** menu. The **Advanced Control Services** window appears.
2	Click **Control Plans** to open the CPE.
3	From the **File** menu in the Control Plan Editor window, select **New**. The Start feature node is 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.

Step	Action
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 **acs.jnlp**, please log out, then access the CPE by selecting **Service Management System->Services > ACS Service**.

Allocating Additional Feature Nodes

Follow these steps 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.

Step	Action
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.

Chapter 2

Service Provider

Overview

Introduction

This chapter explains how to configure a service provider.

In this chapter

This chapter contains the following topics.

Service Provider Creation

About service providers

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 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.

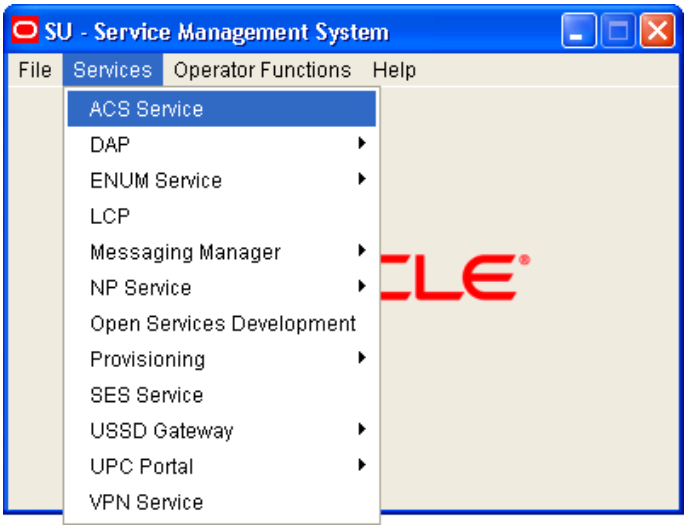
Follow these steps to allocate CCS permissions for the SMS user.

Step	Action
1	Connect to SMS by entering the following url: `http://SMShostname/sms.jnlp` Where *SMShostname* is the hostname of an SMS in the IN.
2	Log in to the SMS UI as the su user.
3	Create an SMS user, if you have not created one already. For more information, see *Adding the SMS user* (on page 3).
4	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* (on page 4).

Creating service provider

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

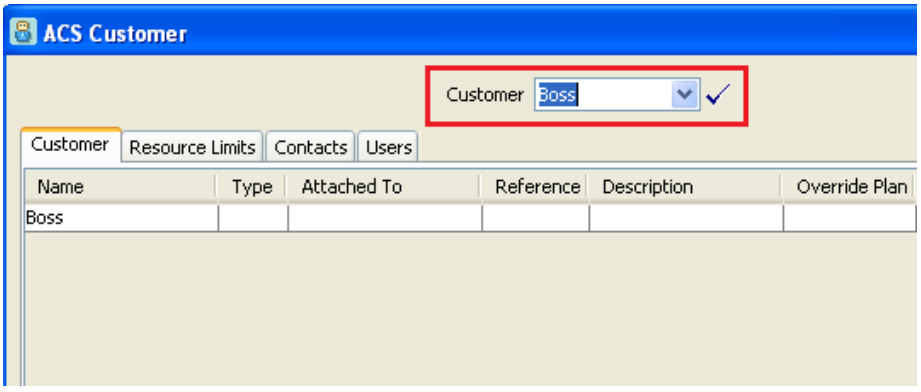
Follow these steps to create a new ACS customer that will become the service provider.

Step	Action
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.

Step	Action

2 Click **Customer**.

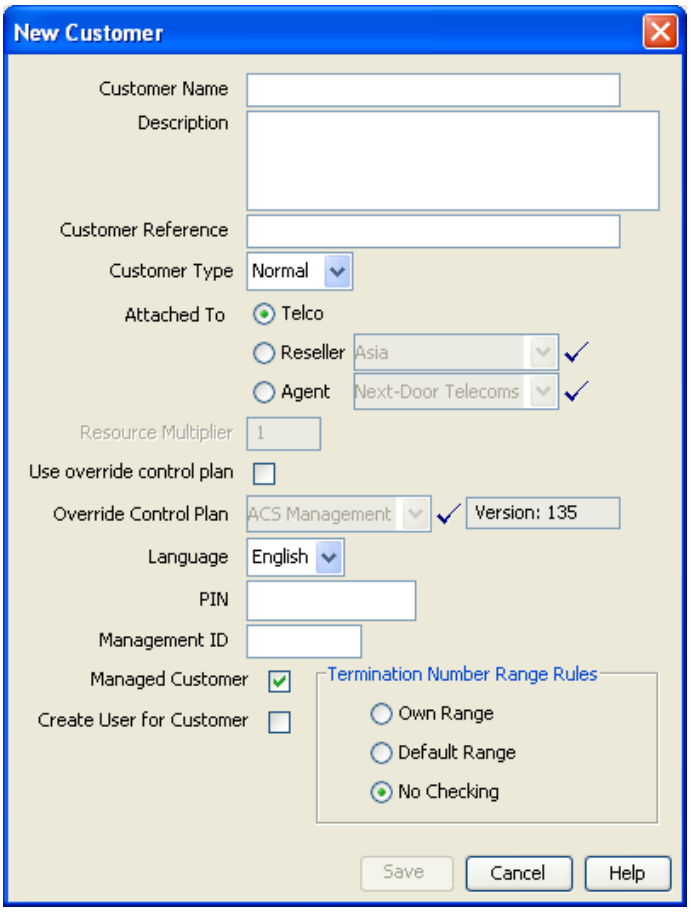
3 To see what customers are already on the system, clear the **Customer** drop down box and **Enter**.



4 Do one of the following:

- Click **New** to create a new ACS customer.
- Select an existing customer and click **Edit**.

Step	Action
5	Give your customer a unique name and reference number (**Customer Name** and **Customer Reference** fields).



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.

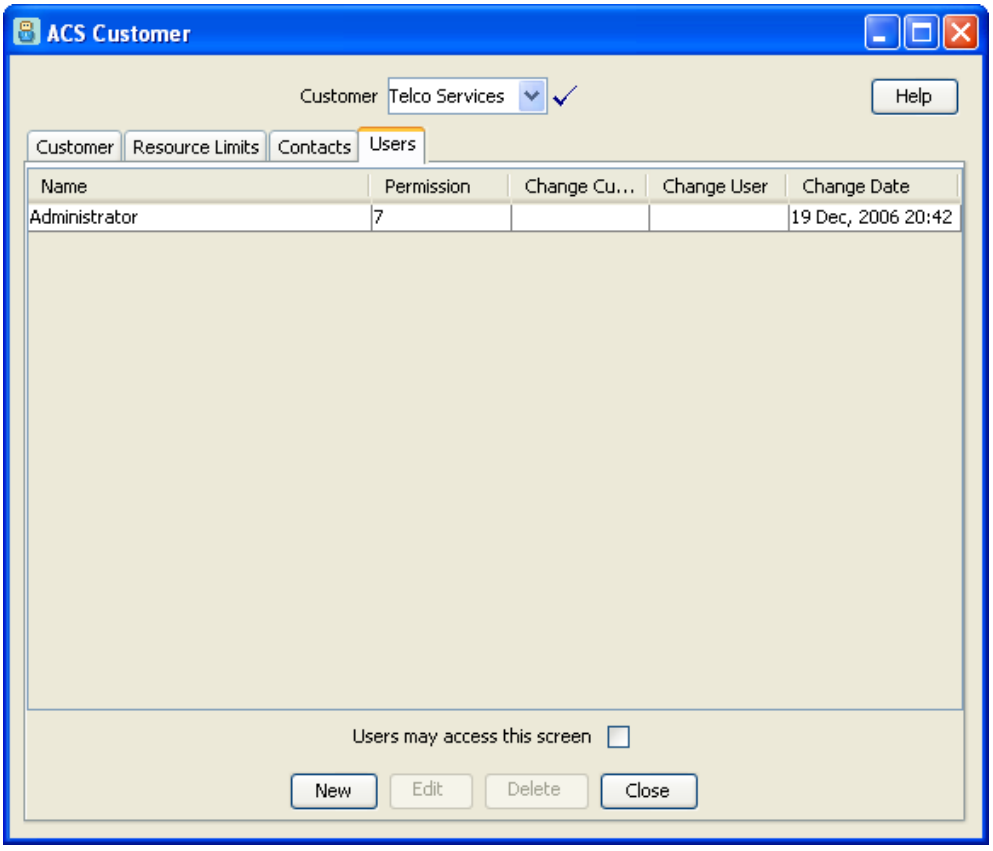
Note: You may need to clear the **Customer** drop down field (often defaulting to Boss) and **Enter** to see your new customer in the list.

Self managed customer

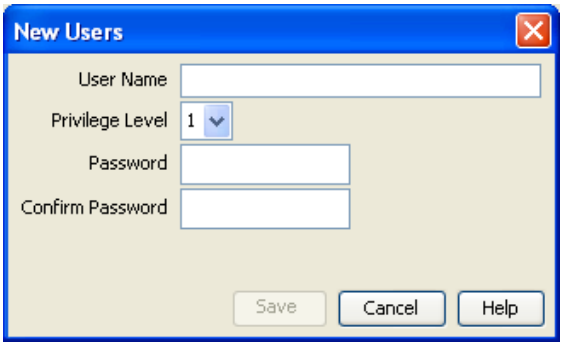
A self managed ACS customer is a customer who is managed directly in ACS, instead of being managed by the operator in SMS. You must manually create an ACS user for each self managed ACS customer, to enable them to log in to ACS directly.

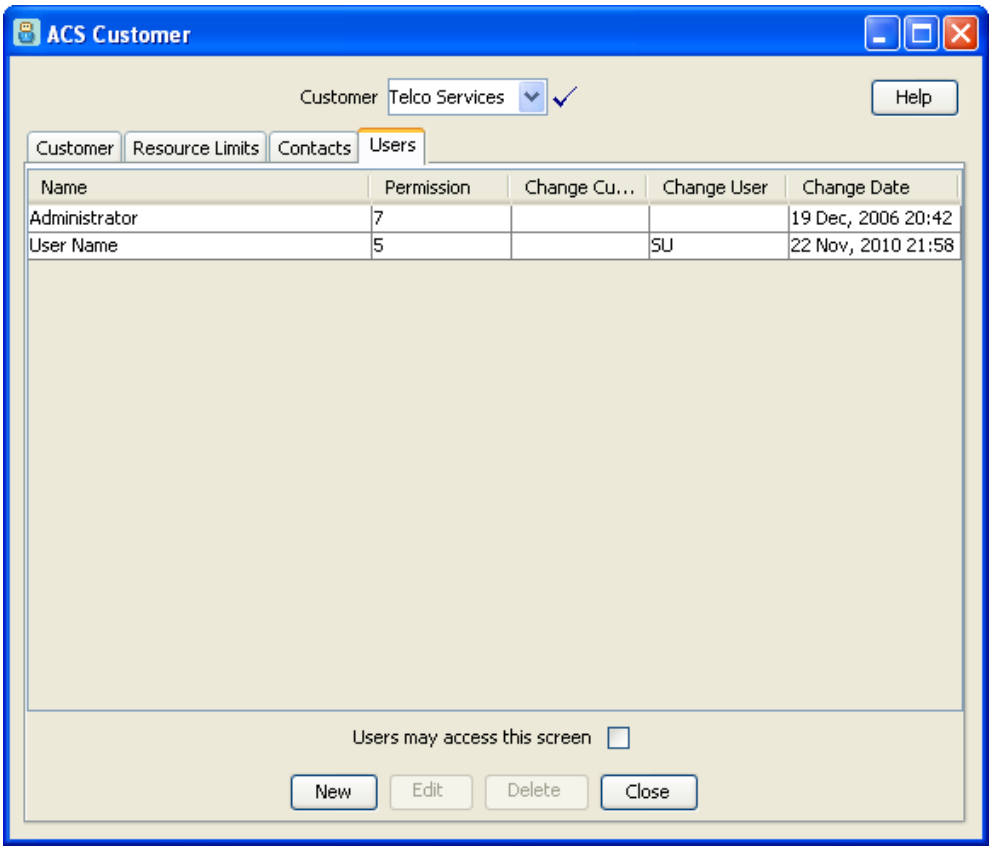
To create or edit a self managed customer user.

Step	Action
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.



Step	Action
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.



Step	Action
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 displays. 
6	Click **CLose**.

7	To test the ACS user for your ACS customer, log in to ACS directly as the ACS user, by entering the following url in an internet browser:
	`http://SMShostname/acs.jnlp`
	Where *SMShostname* is the hostname of the SMS on the IN.
	Note: If this is a managed customer, you will not be able to log in this way. Instead, the customer will be managed by the operator using SMS. You can access the SMS by entering the following url:
	`http://SMShostname/sms.jnlp`

Create / 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)

Follow these steps to create / allocate data sets to your service provider.

Step	Action
1	Select **Services > ACS Service -> 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

Introduction

To configure the ACs component of Convergent Charging Controller, select **Services > ACS Services > Configuration**.

Configure the:

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

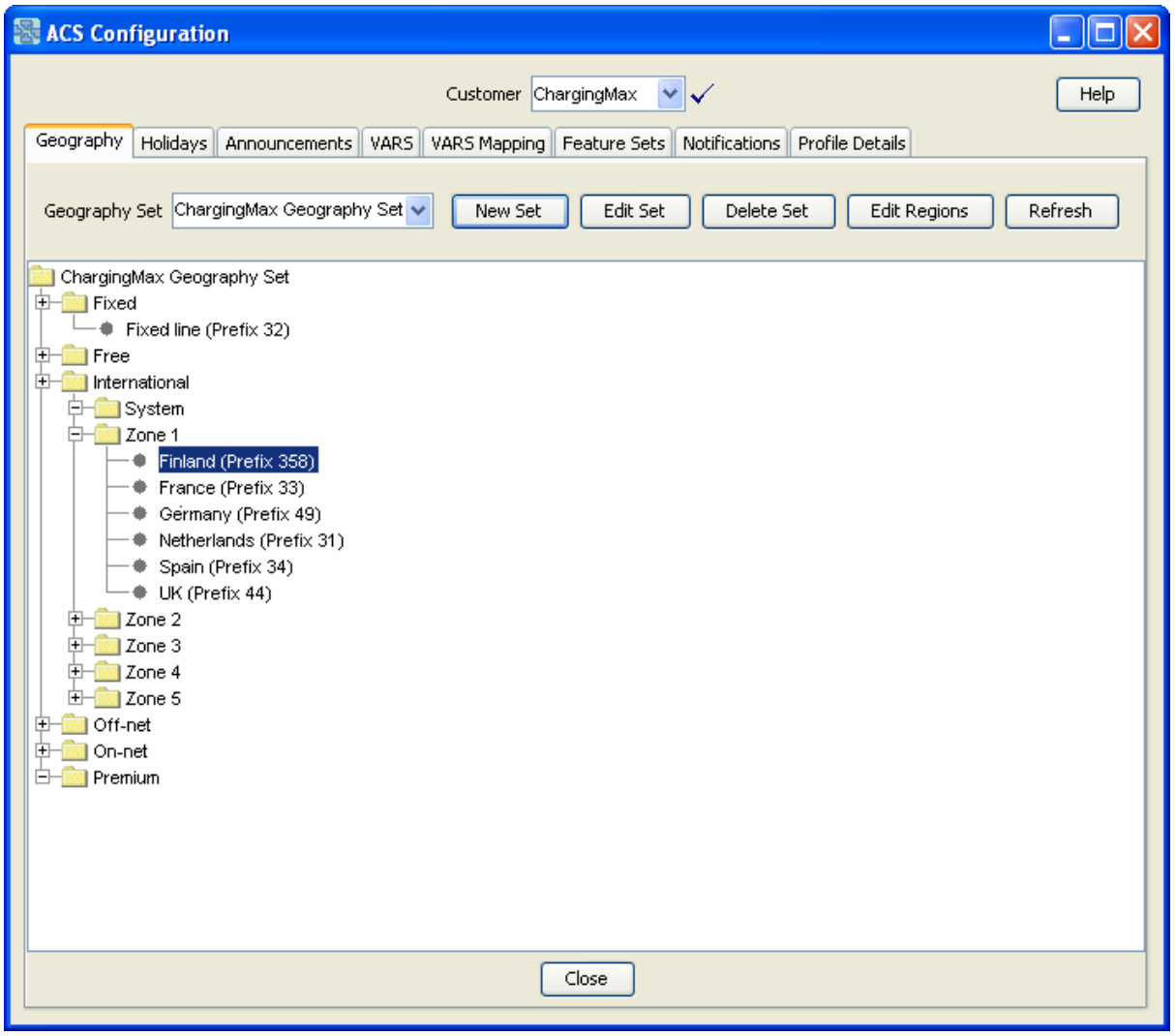
Geography set

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.

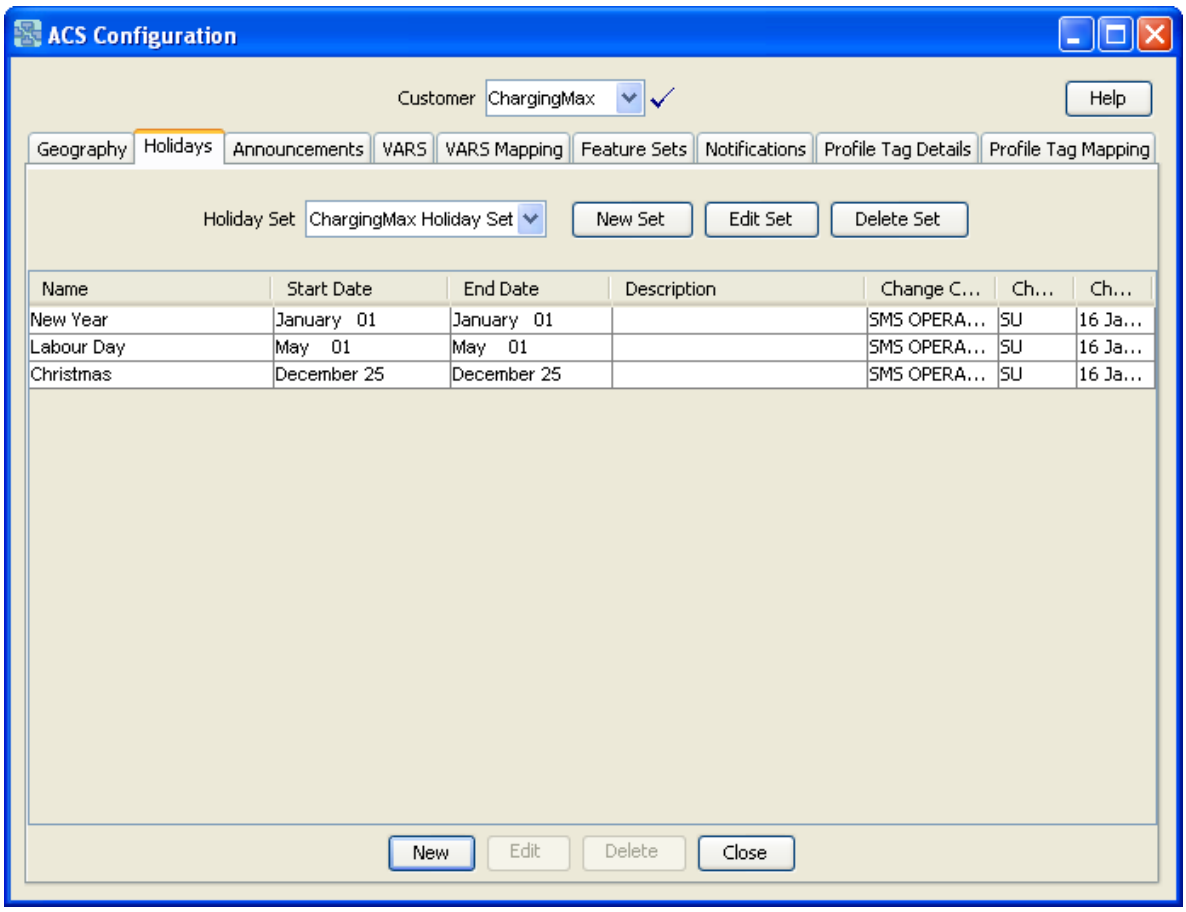
Here you can see the default Prepaid Charging geography set.



Holiday set

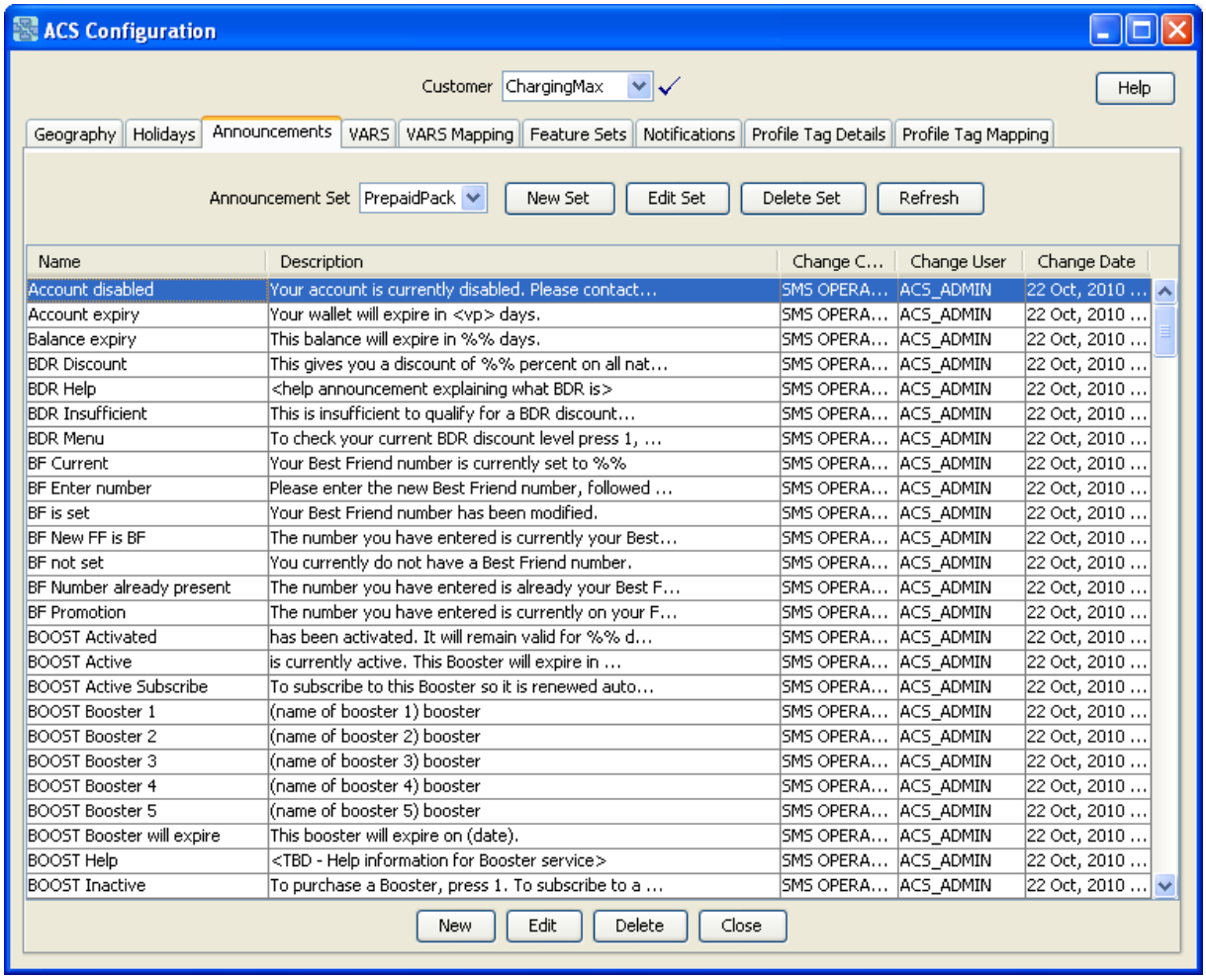
This is where holidays are defined (for example, Christmas and New Years.)

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

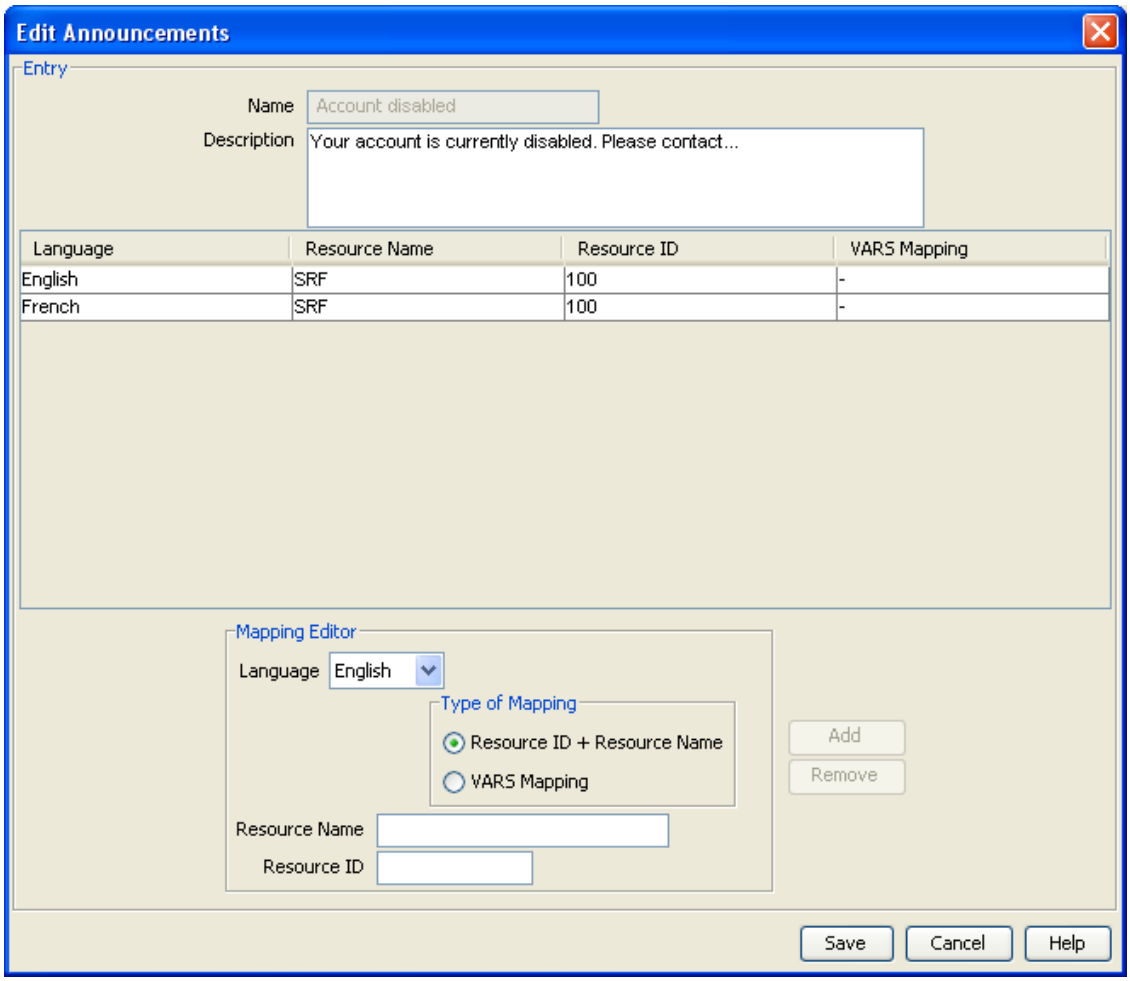


Announcement set

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

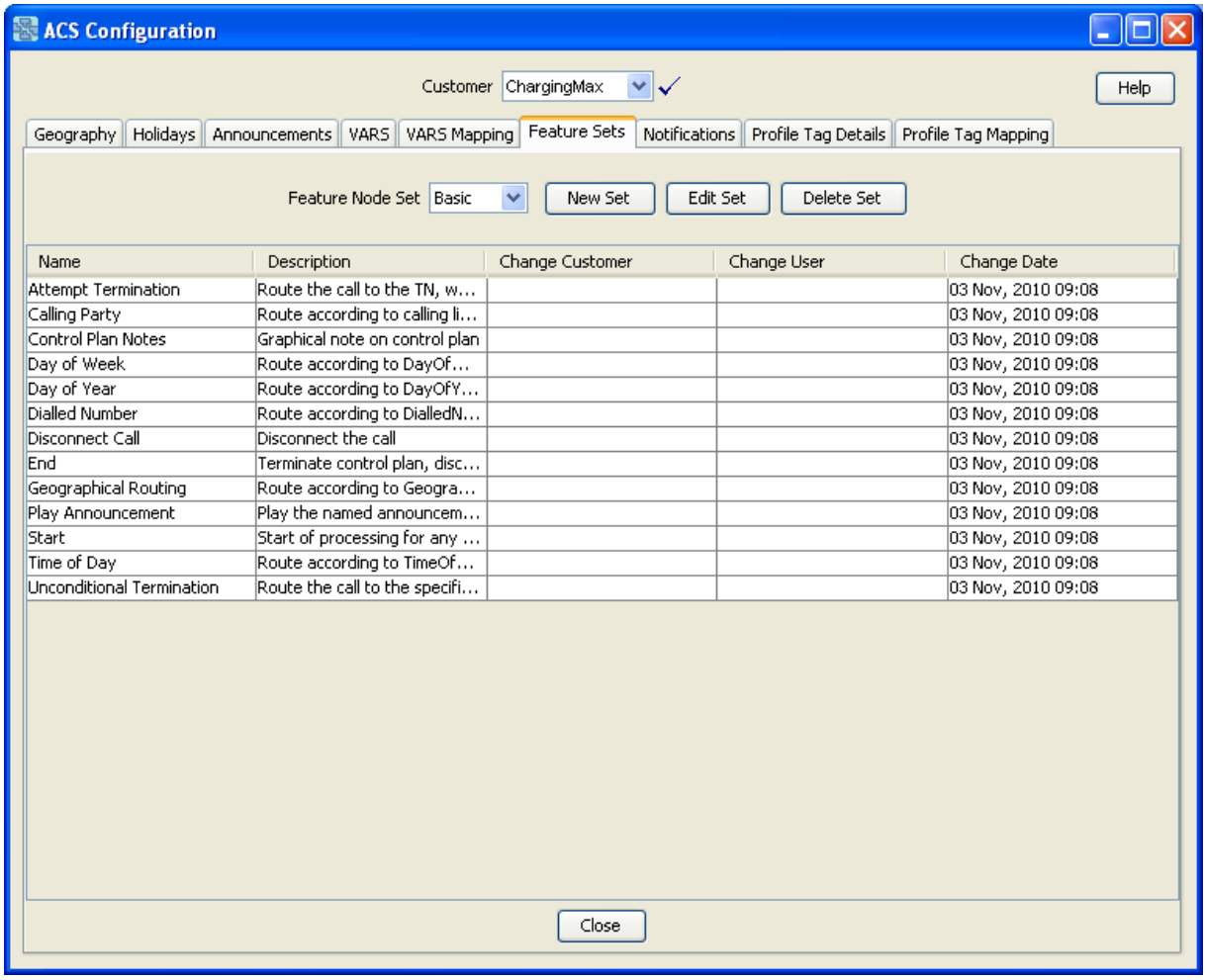


In that announcement set, you have a collection of announcements which are linked to a resource ID (to be played on the IVR) and a corresponding language.



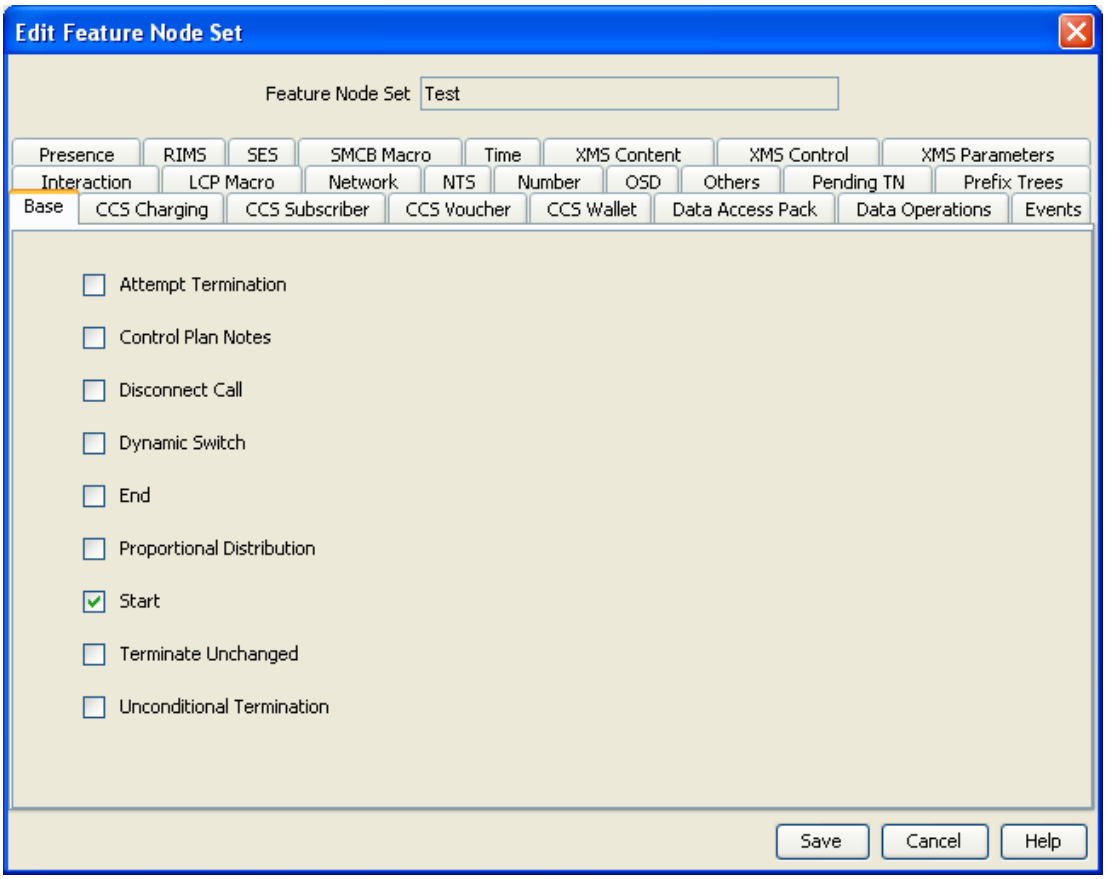
Feature Node set

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

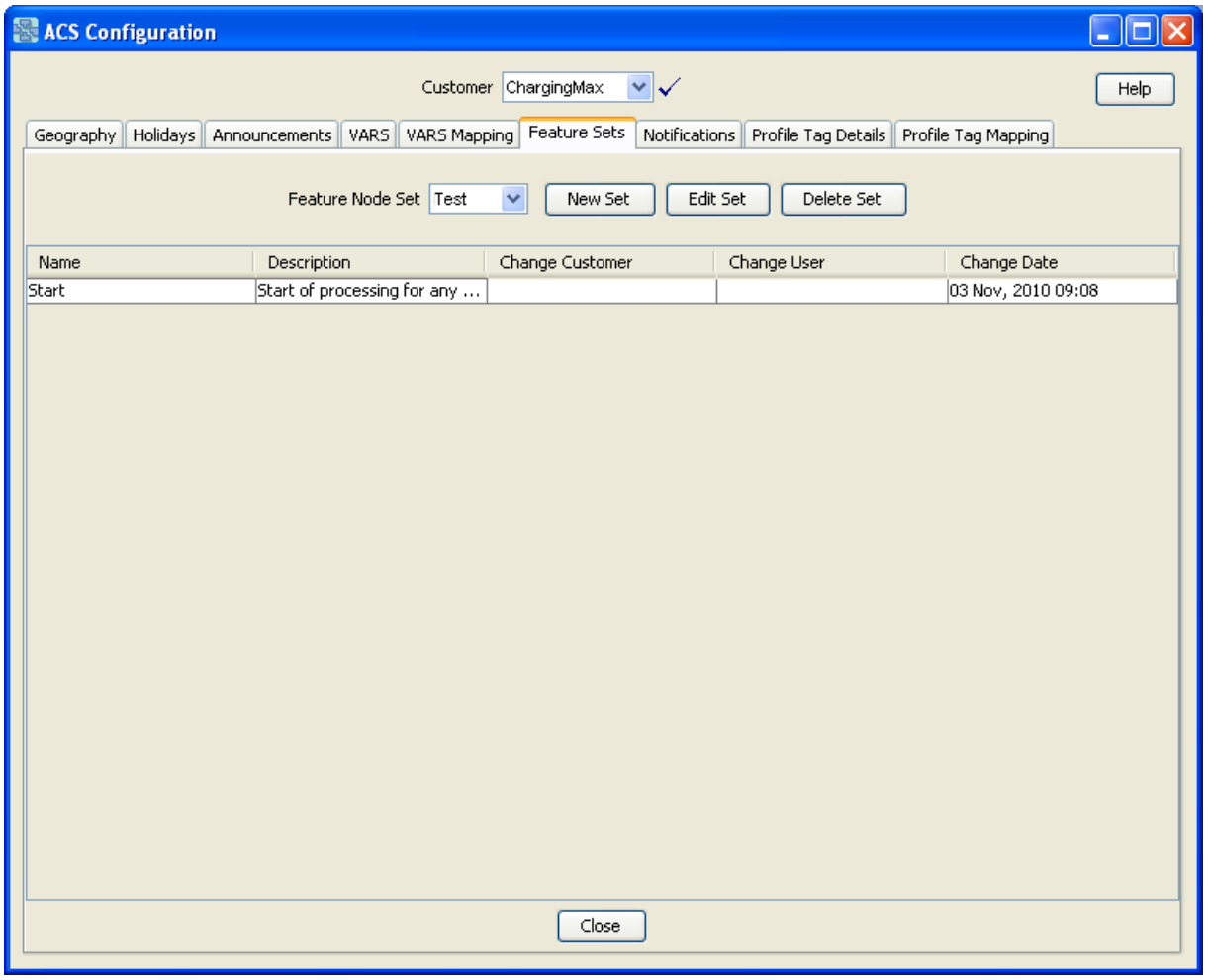


You can create a specific set to restrict rights for your users.

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



The feature set will look like this:



Service Provider Limits

Introduction

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

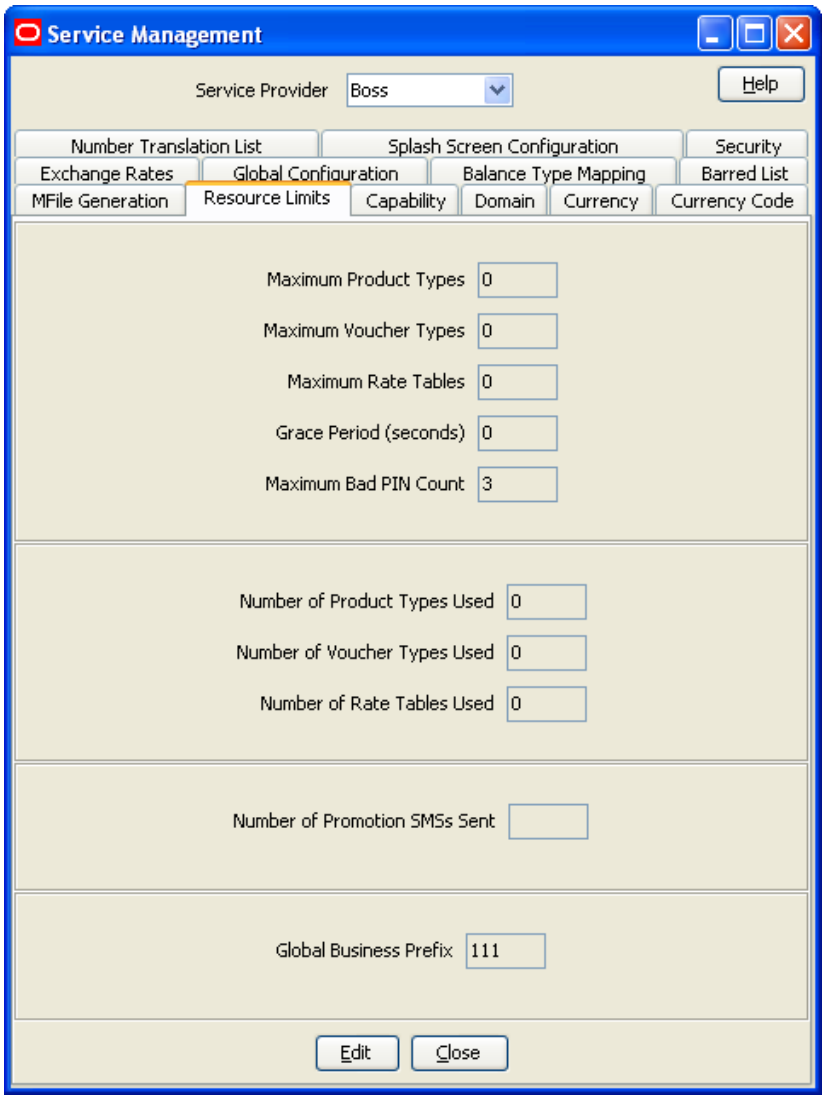
- Limits
- Business Prefix
- Sets
- Call barring
- Number translations

Setting limits

Follow these steps to set resource limits.

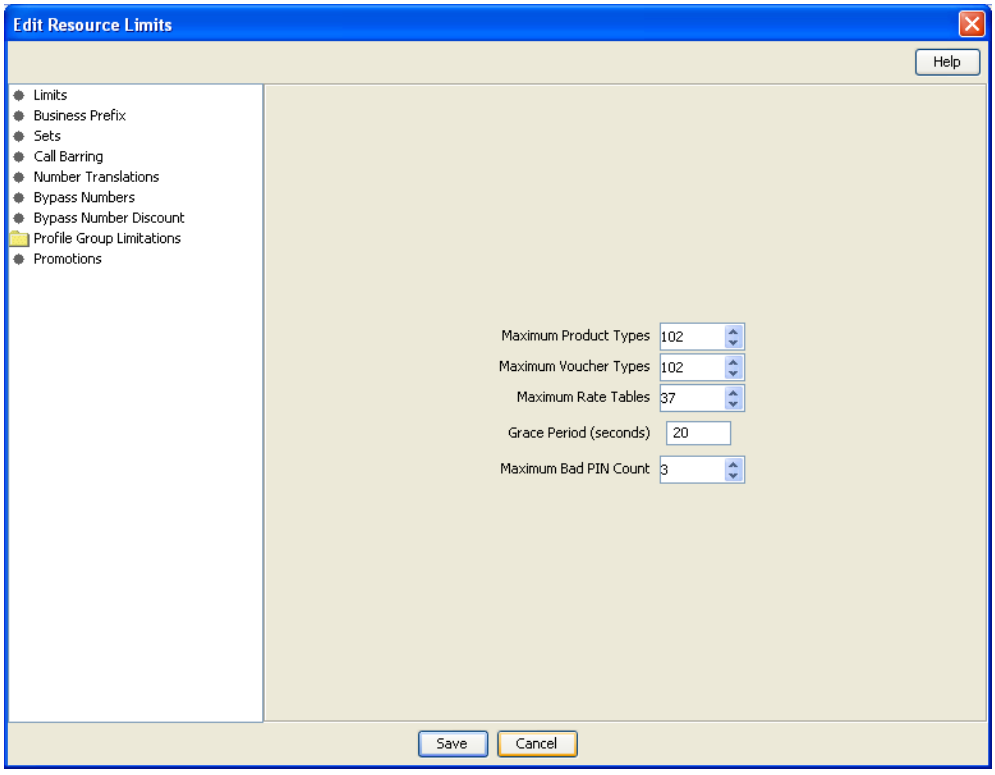
Step	Action
1	Select **Services > Prepaid Charging -> Service Management**.
2	Select the **Resource Limits** tab.

Step	Action



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

Step	Action



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 dependant 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**.

Chapter 3

Product Type

Overview

Introduction

This chapter explains the product types.

In this chapter

This chapter contains the following topics.

Product Types

About product types

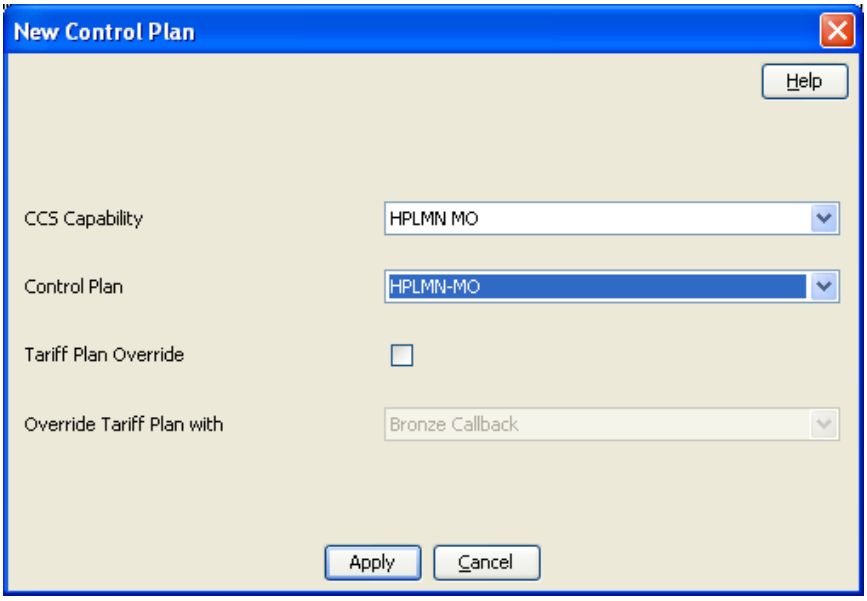
A product type is a collection of services that are provided to the group of subscribers who use that product type.

Creating product types

Follow these steps to create a product type.

Step	Action
1	You create a product type with: • A default tariff plan. See *Creating a Tariff* (on page 33) to create one. • A control plan associated with it. See *Create a Control Plan* (on page 6) to create one. • Some announcements. See *Announcement set* (on page 18) 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* (on page 50). Select a CCS Capability from the drop down menu (defined in the **acs.conf** file).

Step	Action
5	Select the control plans option, then **New**.



6	Select the control plan created earlier.
7	Select the Announcements option and select the Announcement set created earlier.

Chapter 4

Subscribers

Overview

Introduction

This chapter explains how to create subscribers.

In this chapter

This chapter contains the following topics.

Subscriber Creation

Introduction

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 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 on the wallet it is accessing.

Viewing existing subscribers

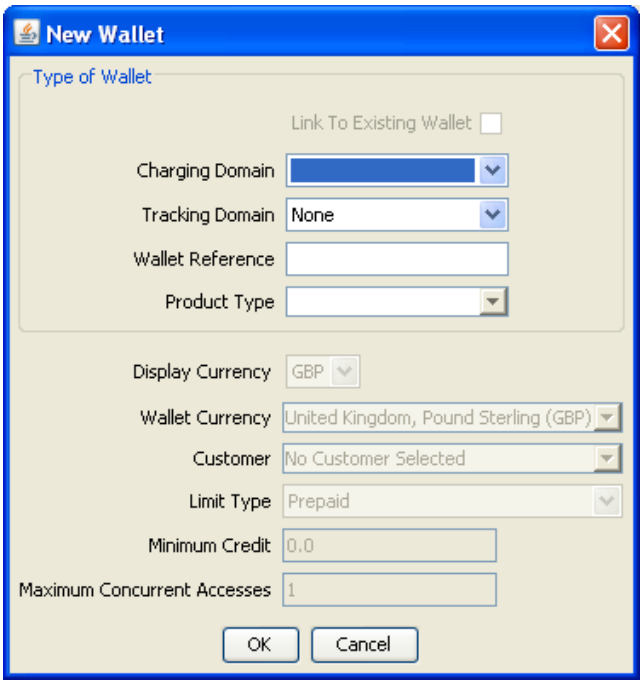
Follow these steps to find all existing subscribers.

Step	Action
1	Select **Services > Prepaid Charging > 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

Follow these steps to create a new subscriber account.

Step	Action
1	Select **Services > Prepaid Charging > 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.

Step	Action
	
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 > Wallets option, change the **Wallet State** to frozen.
- **Apply** and **Save** the changes.
- Run a Slpit script to emulate the above 'frozen' subscriber trying to make a call.

Unblock a subscriber

Follow these steps to unblock the previously blocked subscriber account.

Step	Action
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.

Chapter 5

Rating and Tariffs

Overview

Introduction

This chapter explains rating and tariffs.

In this chapter

This chapter contains the following topics.

Tariff Creation

Introduction

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

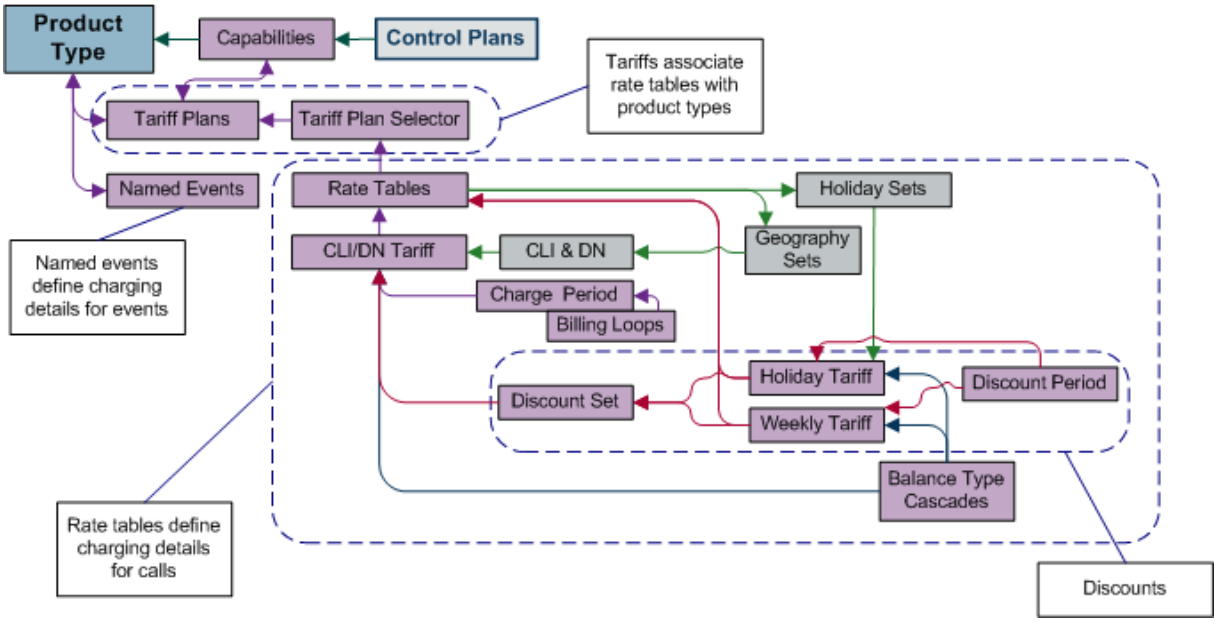
Tariff plans enable you to define which rate table applies to a product type at a particular time.

Rate tables define how Prepaid Charging charges for calls.

An ACScustomer 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.



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

Introduction

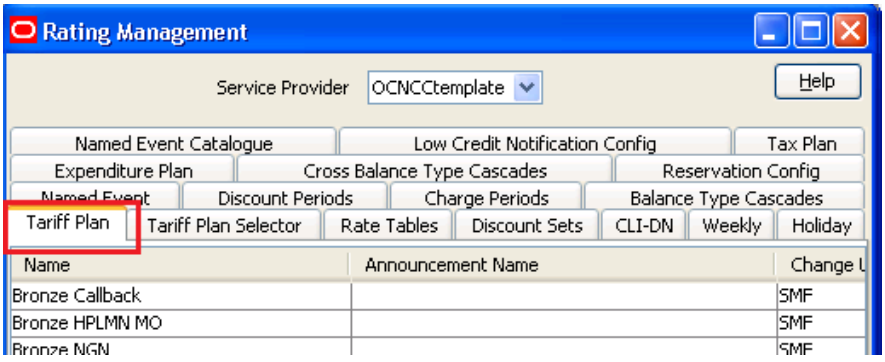
In order to create a proper tariff, you'll need to create the following, in this order.

- *Geography set* (on page 15)
- *Holiday set* (on page 16)
- *Tariff plan* (on page 33)
- *Rate table* (on page 34)
- *Tariff plan selector* (on page 35)
- *Balance type cascade* (on page 35)
- *Charge period* (on page 36)
- *Discount set* (on page 37)
- *Discount period* (on page 38)
- *CLI-DN* (on page 39)
- *Weekly tariff* (on page 40)
- *Holiday tariff* (on page 41)
- *MFile* (on page 42)

Tariff plan

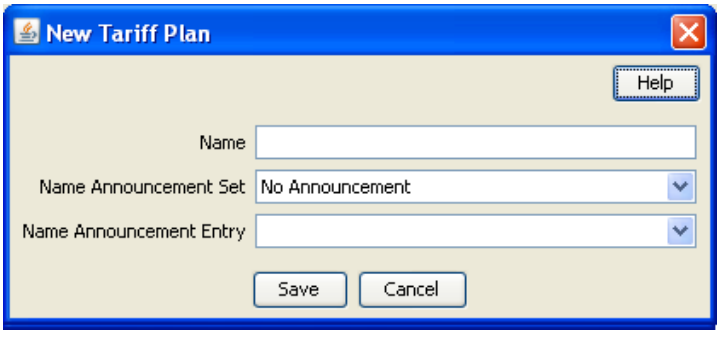
Follow these steps to create a tariff plan.

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



2	Click **New**.

Result: The New Tariff Plan screen is displayed.



2	Give your tariff plan a name.

Step	Action
3	Click **Save**.

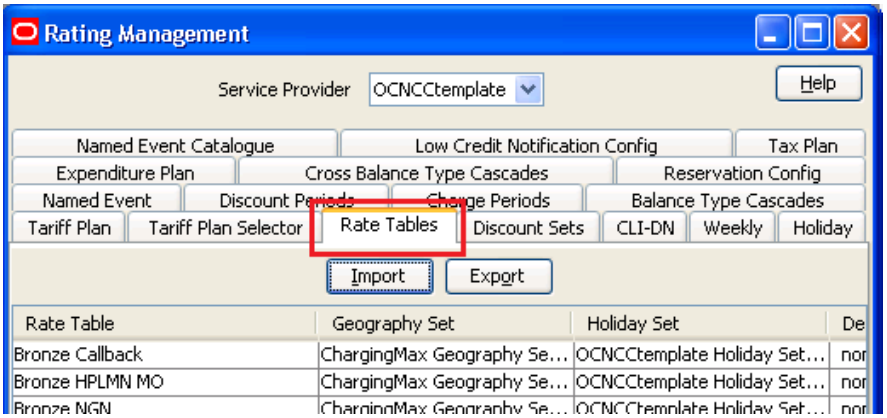
Rate table

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.

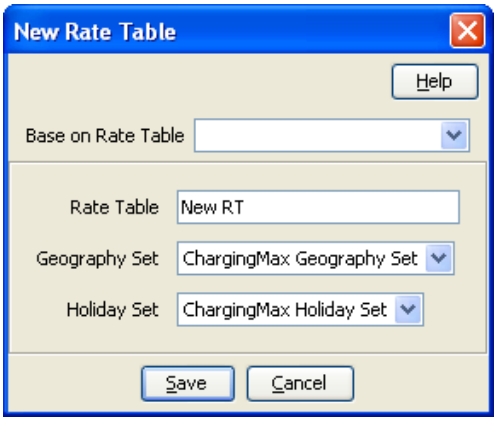
Follow these steps to create a rate table.

Step	Action
1	Select the **Rate Tables** tab.



2	Click **New**.

Result: The New Rate Table screen is displayed.

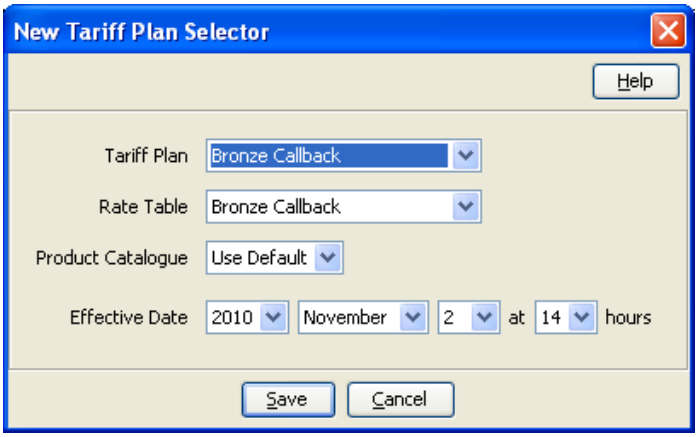


3	Give your rate table a name.
4	Select a Geography set.
5	Select a Holiday set.
6	Click **Save**.

Tariff plan selector

The tariff plan selector links the tariff plan and the Rate Table. Follow these steps to create a tariff plan selector.

Step	Action
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.

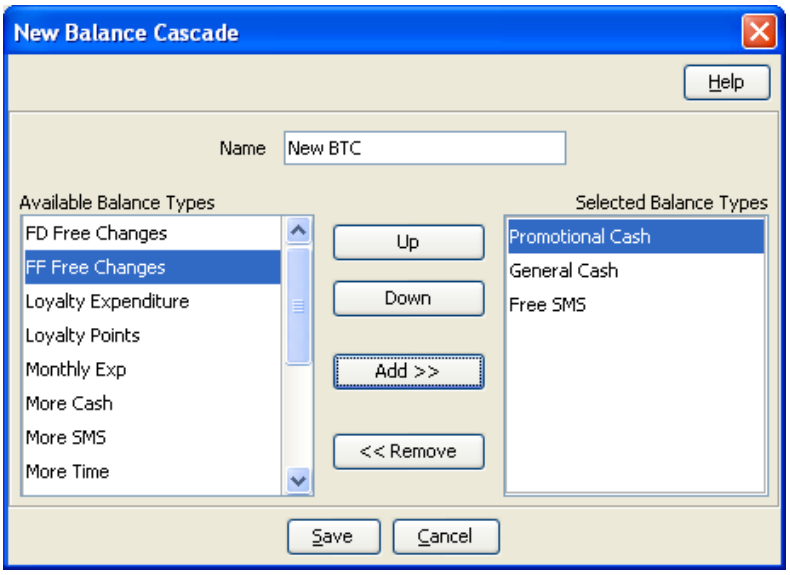
The Balance cascade is how the balance is going to be used, if we take the example bellow, we will first use the Promotional Cash, then the General Cash, then the Free SMS. For example:

1 Promotional Cash
2 General Cash
3 Free SMS

Follow these steps to create a balance cascade.

Step	Action
1	Select the **Balance Type Cascades** tab.
2	Click **New**. **Result:** The New Balance Cascade screen is displayed.

Step	Action



3 Type a Balance Type Cascade name

4 Select an available balance type and **Add**. Repeat until your have selected all balance types you require.

5 Use **Up** / **Down** to order the selected balance types.

6 Click **Save**.

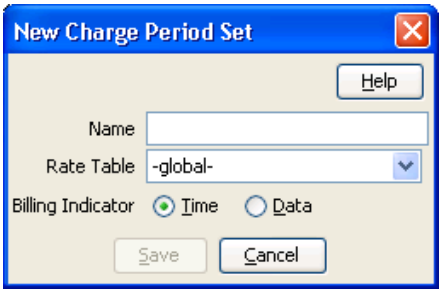
Charge period

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.

Follow these steps to create a charge period set with charge periods.

Step	Action
1	Select the **Charge Periods** tab.
2	Click **New Set**. **Result:** The New Charge Period Set screen is displayed.

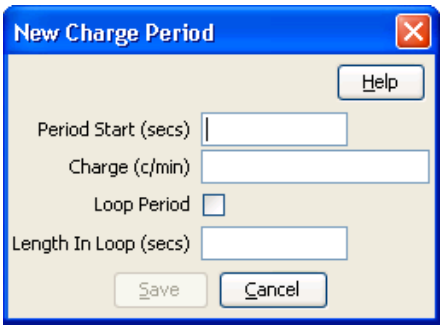


3 Type a name for the charge period set

4 Select the associated rate table.

5 Select the billing indicator.

6 Click **Save**.

Step	Action
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.

Follow these steps to create a new discount set and discount periods for the discount period set.

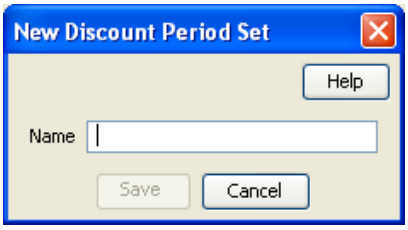
Step	Action
1	On the **Discount Sets** tab, select, from the **Rate Table** drop down box, the rate with which to associate a discount set.

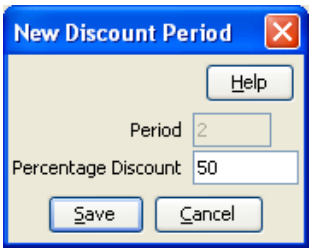
Step	Action
2	Click **New**. **Result:** You see the New Discount Set screen.  The **Rate Table** field displays the rate table selected.
3	Enter a name and description.
4	Click **Save**.

Discount period

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

Follow these steps to create a new discount set and discount periods for the discount period set.

Step	Action
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.

Step	Action
	 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.

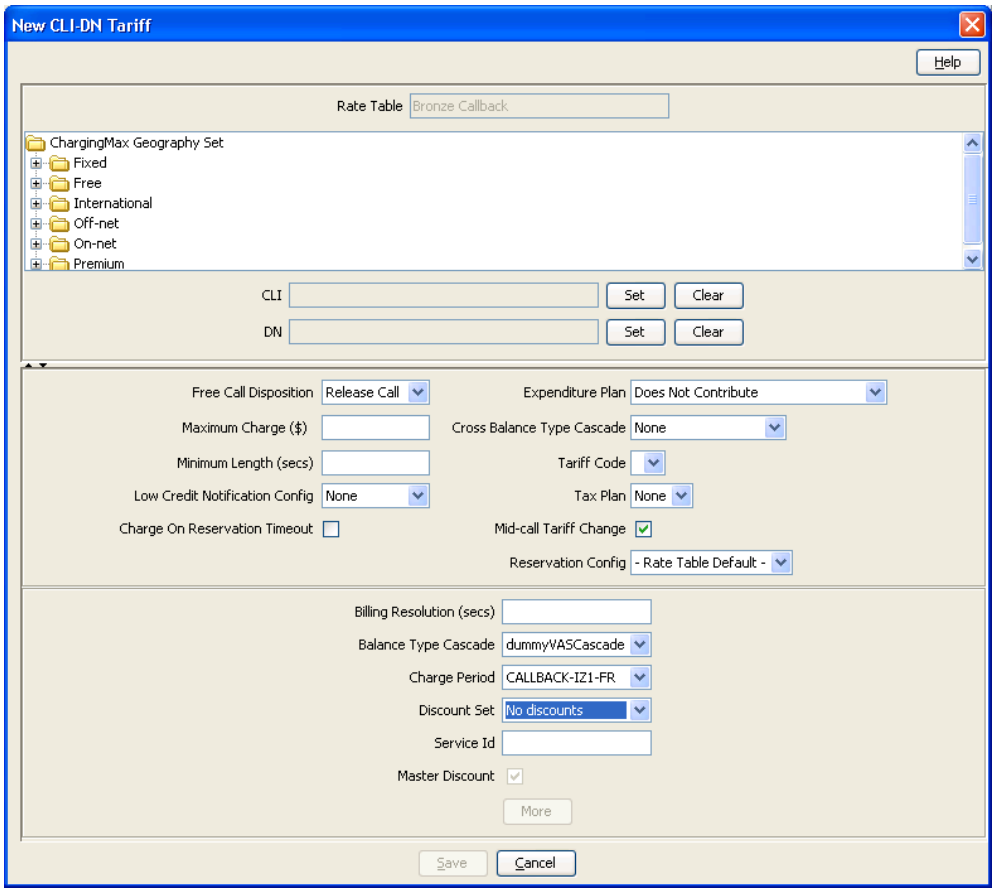
Use the records you have defined so, to configure a CLI-DN record applicable to the rate table you created. Each rate table includes rates for calls between two geographical areas.

The tariff code is a 2 digit hexadecimal code that is usually provided by the switch manufacturer. The tariff code is set in **ACS Services -> Resources -> 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.

Follow these steps to create a new CLI-DN.

Step	Action
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

Step	Action



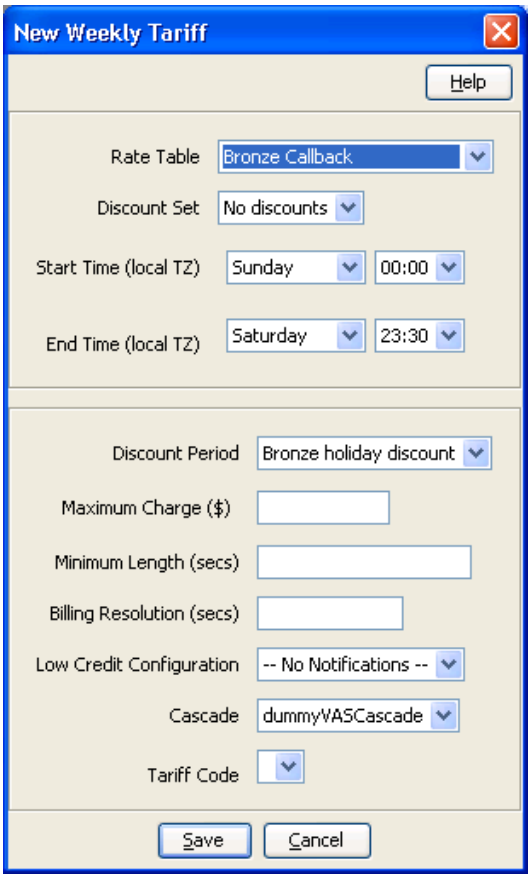
3 Click **Save**.

Weekly tariff

Weekly tariff are used to define specific discount during period of the week (off-peak for example). Follow these steps to create a weekly tariff.

Step	Action
1	Select the **Weekly** tab.
2	Click **New**. **Result:** The New Weekly Tariff screen appears.

Step	Action



3 Fill in the fields.
Refer to the *CCS User's Guide* for information on the fields.

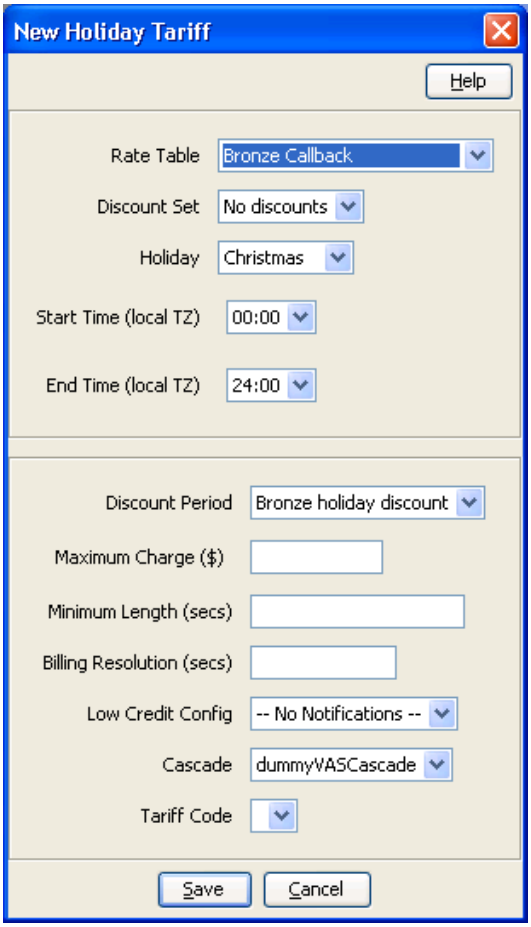
4 Click **Save**.

Holiday tariff

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

Step	Action
1	Select the **Holiday** tab.
2	Click **New**. **Result:** The New Holiday Tariff screen appears.

Step	Action



3 Fill in the fields.

Refer to the *CCS User's Guide* for information on these fields.

4 Click **Save**.

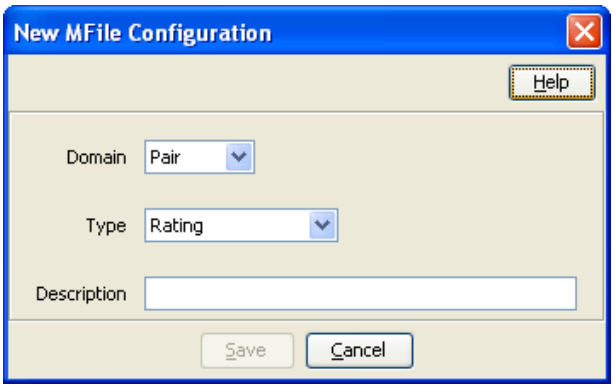
MFile

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.

Follow these steps to create an MFile.

Step	Action
1	Select **Prepaid Charging -> Service Management** and select the **MFile Generation** tab.
2	Click **New**. **Result:** The New MFile Configuration screen appears.

Step	Action



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.

Chapter 6

Adding a Service

Overview

Introduction

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

In this chapter

This chapter contains the following topics.

Service Creation

Introduction

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. Here is the step by step service creation.

Step	Action
1	Determine how your service will be triggered, let assume by SK (not an internal service).
2	Create the *Service Key* (on page 48) entry in **SLEE.cfg**.
3	Create the *Service* (on page 48) in **SLEE.cfg**.
4	Create the *Service Entry* (on page 48) in **acs.conf**.
5	Create a *Service Provider Creation* (on page 9).
6	*Create a control plan* (on page 6) that will define what the service will be doing.
7	Create a *capability* (on page 50) for your service use 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 to customize the action for range of subscriber as example, you'll need to create product types and add subscriber in those product types or have multiple capabilities based on CdPN. To create a product type, you will first need to *Rating and Tariffs* (on page 31).
8	Create a *product type* (on page 25).

Step	Action
9	*Creating a subscriber account* (on page 28) 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

Introduction

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`

EventTypeBCSM	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:

Field	Description
	• **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	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:
	`imsi "530240100000536"`
	`countryCode "530"`
	`networkCode "24"`

vlrNumber	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): `vlrNumber (1) "64241420003"`

CellGlobalIdOrServiceAreaIdFixedLength	The parameter in the snoop trace is broken into four parts in the SLPIT script. These parts are extracted following the rule below:
	`CellGlobalIdOrServiceAreaIdFixedLength ::= OCTET STRING (SIZE (7))`
	`- Refers to Cell Global Identification or Service Are 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`

Field	Description

	From the example snoop above, the CellGlobalIdOrServiceAreaIdFixedLength is 35F04200023C55, which gives the following fields in the SLPIT script:
	`countryCode "530"`
	`networkCode "24"`
	`locationAreaCode 0020`
	`cellID 50005`

callreference	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. `callreference "1E1B7CE329"`
mscAddr	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): `mscAddr (1) "64241420003"`
CalledPartyBCD Number	This specifies the B party number (the number dialed on the handset) and is defined in SLPIT in the following format: `calledPartyBCDNumber (8) "0241499216" numberPlan 1`

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_MO
SERVICEKEY=INTEGER 0x19200000015 CCS_HPLMN_MO
```

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

Service Entry

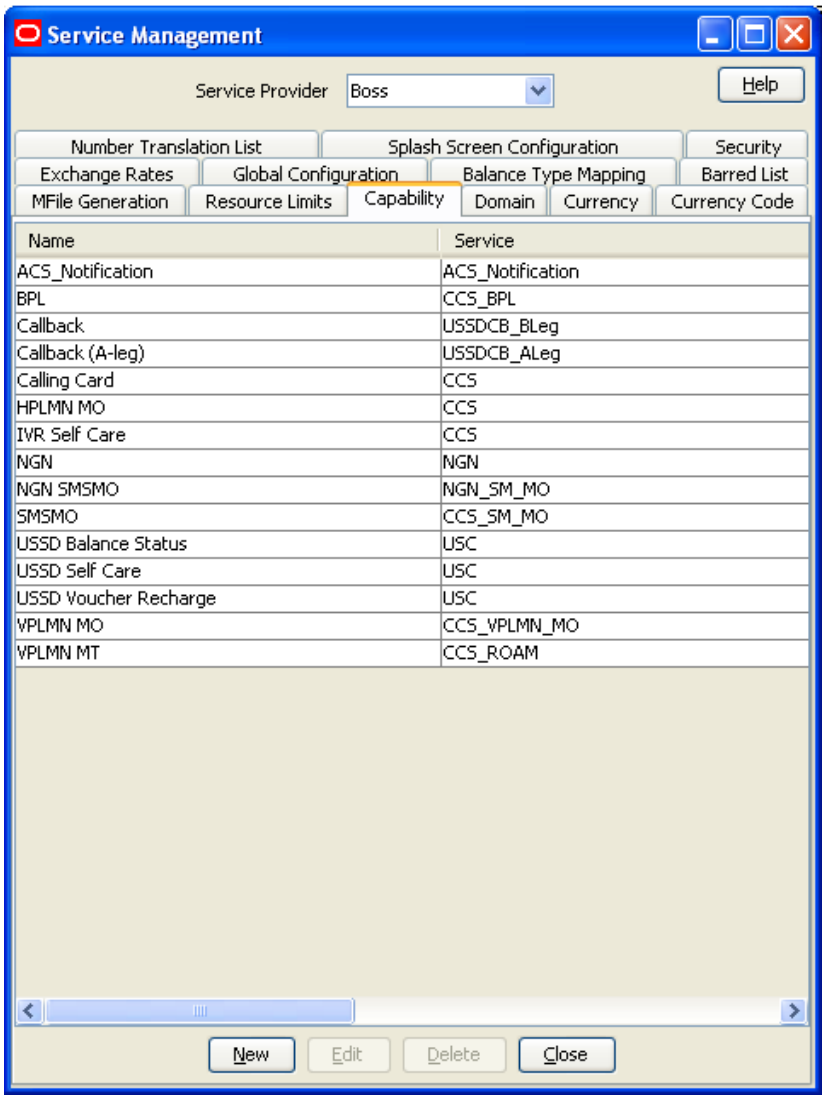
In this example, the CCS service is triggered for mobile originating voice calls as the CCS service library is used to load the service. The Service Entry in **acs.conf** appears as below:

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

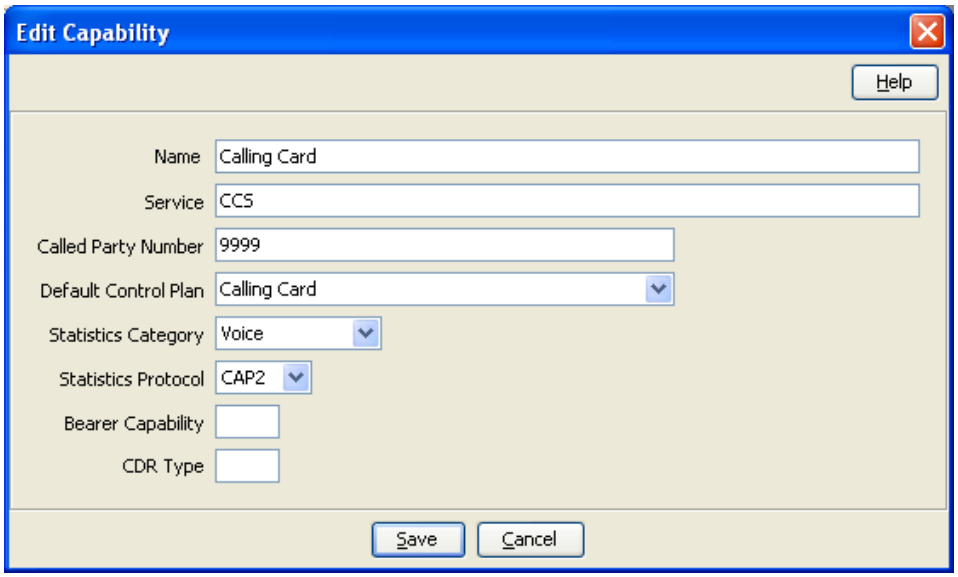
Refer to *ACS Technical Guide* for more information on Service Entry definitions.

CCS Capability

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



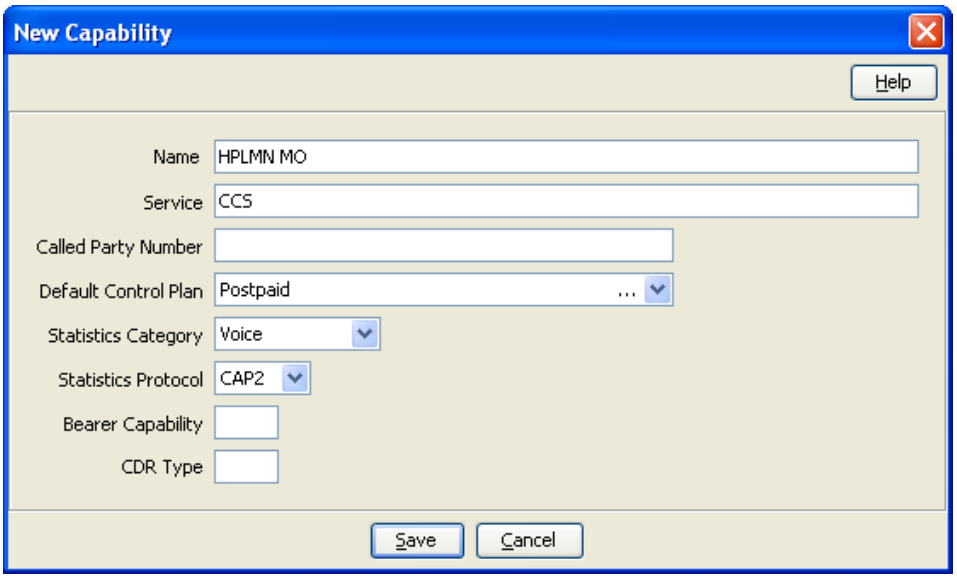
For example, **Prepaid Charging > Service Management > Capability**.



Creating a capability

Follow these steps to create a capability.

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

Step	Action
9	
10	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*.

Chapter 7

Tools

Overview

Introduction

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

In this chapter

This chapter contains the following topics.

Basic SLPIT

Introduction

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 language(s) spoken by the telephony network(s). 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 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 quirkiness 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.

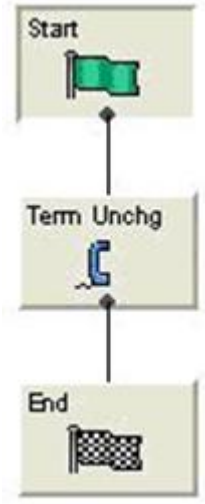
Basic SLPIT script

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.

Step	Action

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 number`
	`DN ?= "015111111"`
	`CLI ?= "0139411111"`
	`send {`
	`initialdp`
	`calledpartynumber DN`
	`callingpartynumber CLI`
	`callingpartyscategory 10`
	`locationnumber CLI`
	`eventtypebcsm 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).  **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.

Step	Action
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 basic`
	`Calls Run: 1`
	`Calls 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.`

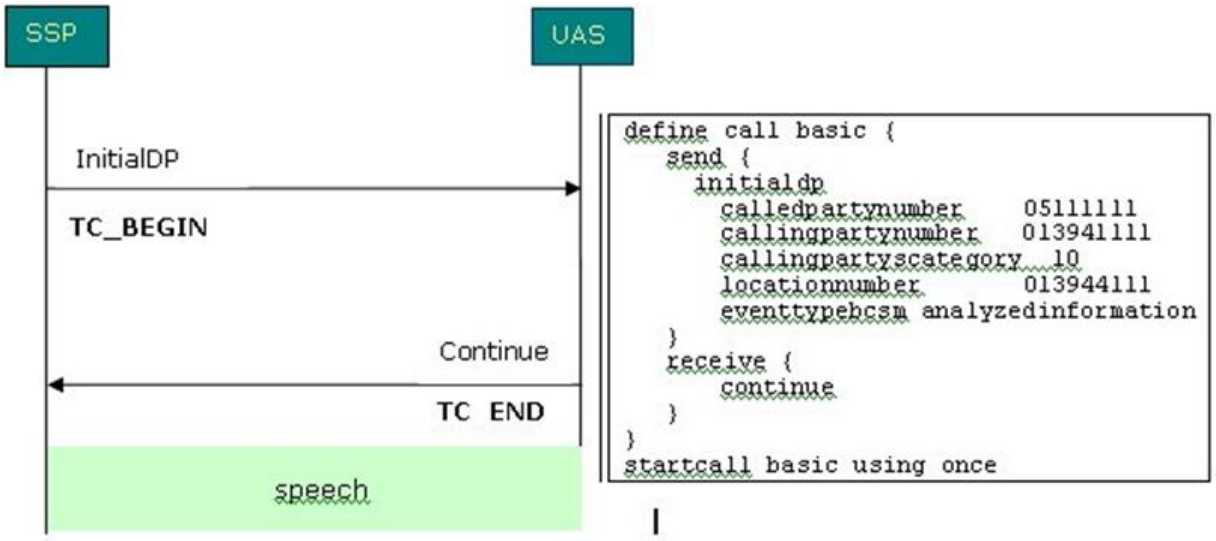
5	Try running SLPIT without `–v` and also with '> acs_basic.out', for example: `slpit –v –k 111 < acs_basic.slp > acs_basic.out`
	Note: You will be looking at the contents of a basic script in the next task.
	If the script failed: **1** Create the above control plan and assign it to your customer's service number. Then schedule the control plan to run against your customer's service number. **2** 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: **/volA | volB/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.



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:

```
…
initialdp
calledpartynumber 0151111111
callingpartynumber 0139411111
callingpartyscategory 10
locationnumber 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 = "0151222222" 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 {.
`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

Script	Description
	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 {
initialdp
appContext "0,4,0,0,1,0,50,1"
calledpartynumber "441394100005"
callingpartynumber "441394100009" // change this to your subscriber's number
callingpartyscategory 10
eventtypebcsm analyzedinformation
}
```

```
// The maxDuration parameter tells the switch the maximum time permitted before//
the next reservation request should be sent to the SCP
receive {
applycharging
maxDuration 2400
release 0
tone 0
requestreportbcsmevent
eventtypebcsm oCalledPartyBusy (2)
eventtypebcsm oNoAnswer (2)
dpspecificcriteria applicationtimer 10
eventtypebcsm oAbandon (1)
eventtypebcsm RouteSelectFailure
eventtypebcsm oDisconnect monitormode interrupted (2)
eventtypebcsm oDisconnect monitormode notifyAndContinue(1)
connect destroutingaddr "441394100005"
}

send {
eventreportbcsm
eventtypebcsm oDisconnect misccallinfo request (1)
// The timeNoTariffSwitch parameter tells the SLC how long to charge for in
deciseconds
applychargingreport
receivingSide 2
timeNoTariffSwitch 1200
callActive 1
}
}
startcall ccs_uatb using once
```

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

Provisioning Interface

Introduction

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.

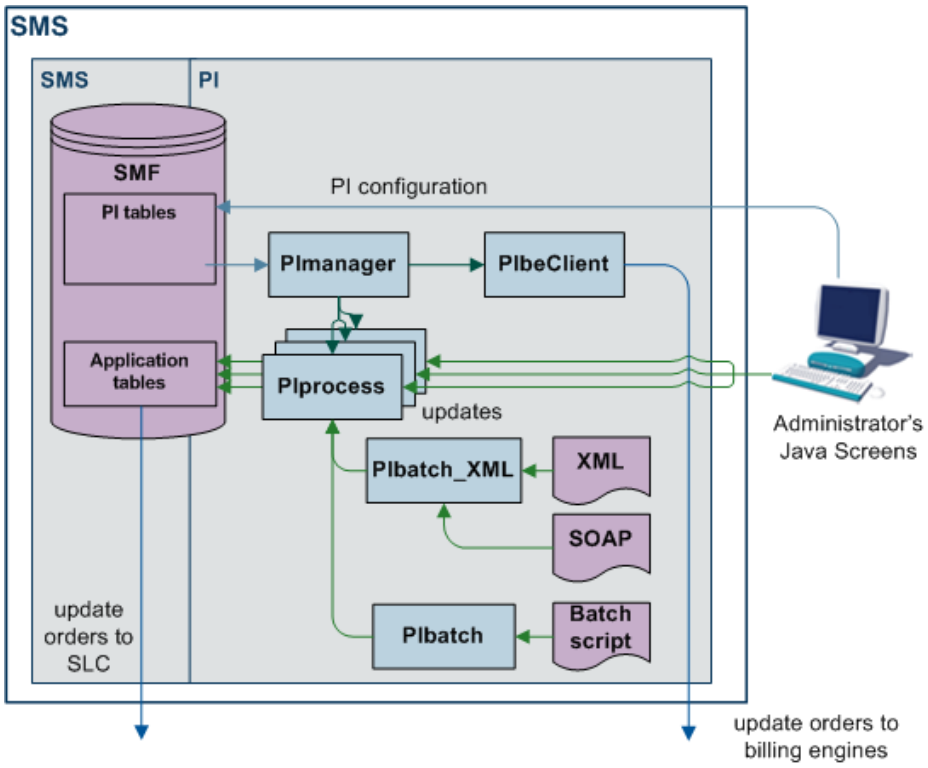
The PI can be used to provide access to the billing engine for customer care operators or by the operator to enable web based self-care for their customers.

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.



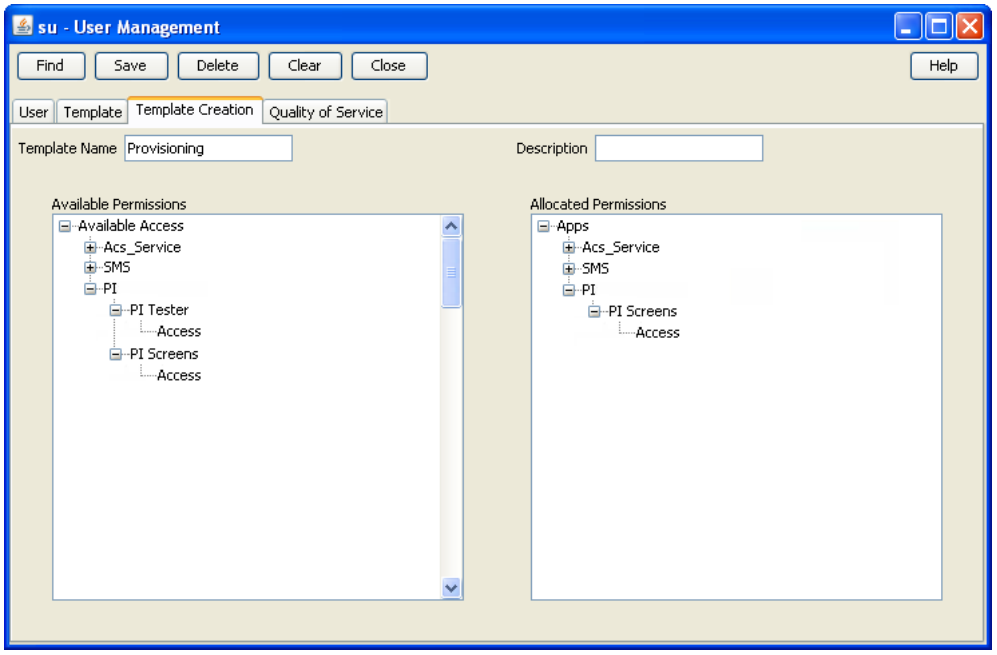
Process descriptions

This table describes the processes involved in the PI application.

Component	Description
PIprocess	Updates the SMF database and the SMS sends the data through replication to the other nodes in the network. PIprocess waits for TCP/IP connections and processes commands sent to it. Commands are processed through a shared library or by executing a PL/SQL function in the SMF database.
PIManager	Stops and starts the PIprocess. PImanager is started by init - you should find an entry in **/etc/inittab** called **pim1**. The log file for this command can be found at **/IN/service_packages/PI/tmp/PImanager.log** You can restart PI in two ways: • `/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 PIprocesses, which are then restarted by `/etc/inittab`.
PIbatch	Allows multiple PI commands to be sent to PIprocesses through a script file: **/IN/service_packages/PI/bin/PIbatch** *script server* Results are placed in a file of the same name but with a .result extension.

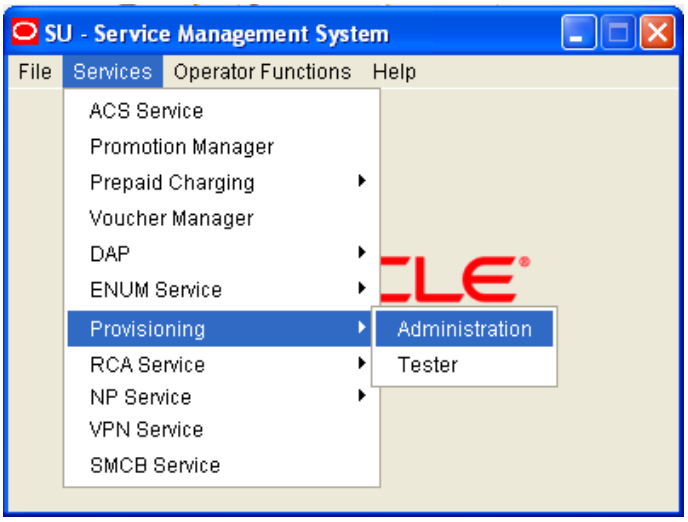
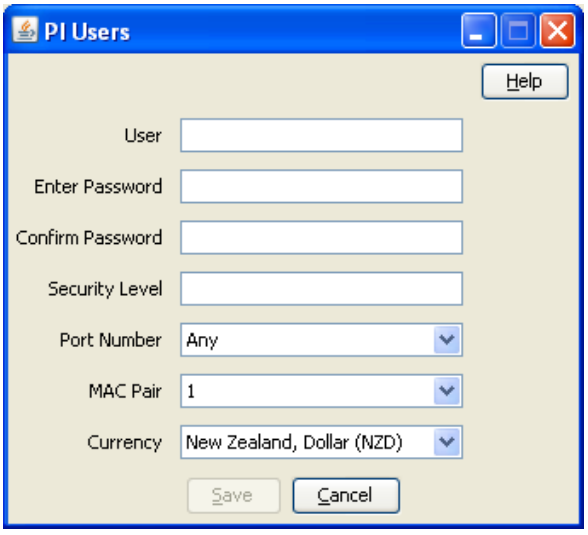
Configuring user access to PI screens

Follow these steps to configure access to PI screens for a specified user.

Step	Action
1	Open the SMS user interface (UI) from an internet browser by entering the following url: `http://SMShostname/sms.jnlp` Where *SMShostname* is the hostname of an SMS in the IN.
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

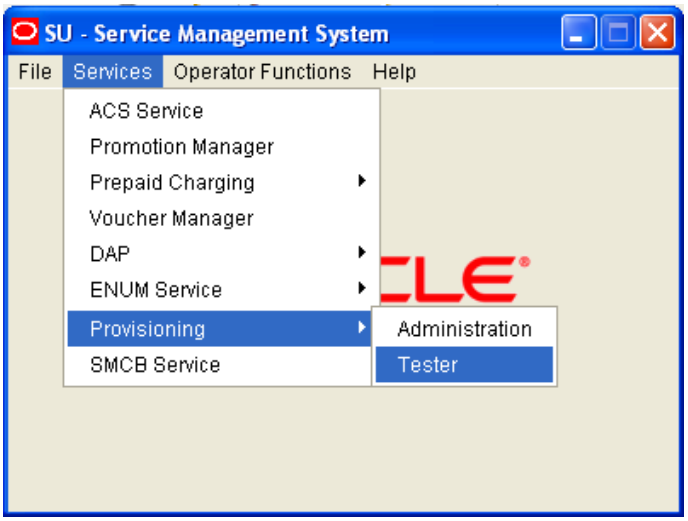
Follow these steps to configure personal computer (PC) connection details to the PI for a specified user.

Step	Action
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* (on page 60).
2	Select the **Administration** option from the SMS **Services, Provisioning menu.** 
3	On **Hosts** tab, click **New**. The PI Hosts window displays.
4	In the **IP Address** field, enter 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 if config from a UNIX terminal.
5	On **Users** tab, click **New**. The PI Users window displays. 

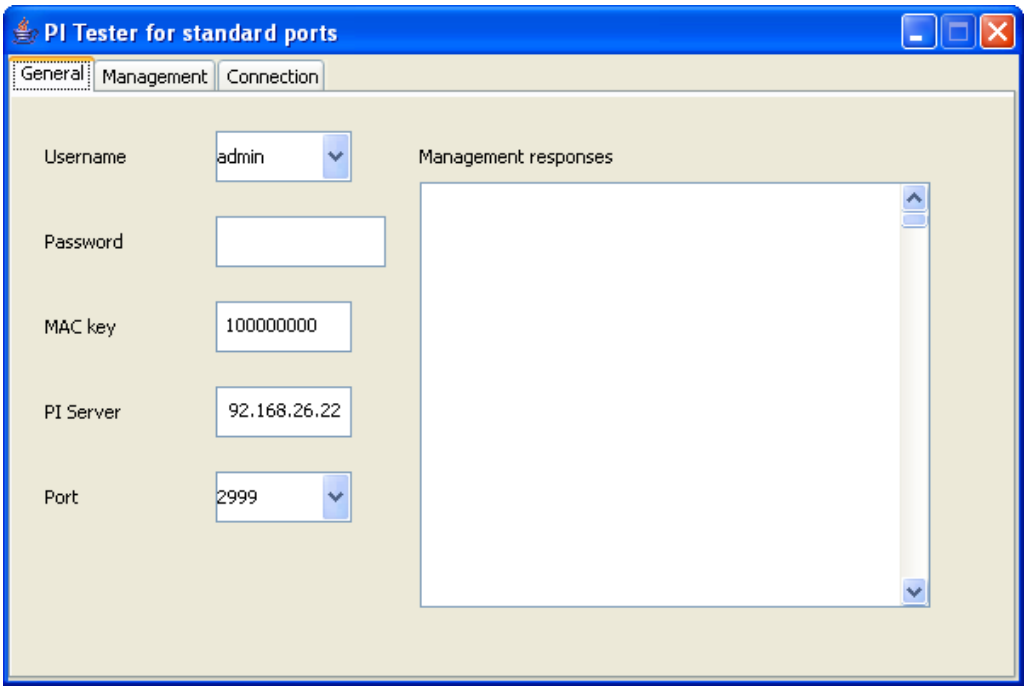
Step	Action
6	Enter the SMS user details for user who will be using the PC. You must allocate security level 99 to the user in the **Security Level** field.
7	Restart the PI to activate the changes by performing the following steps: a. Log in to the SMS as the smf_oper user. b. Go to the following directory: **/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: - `PIrestart.sh`

Testing your User and PC connection

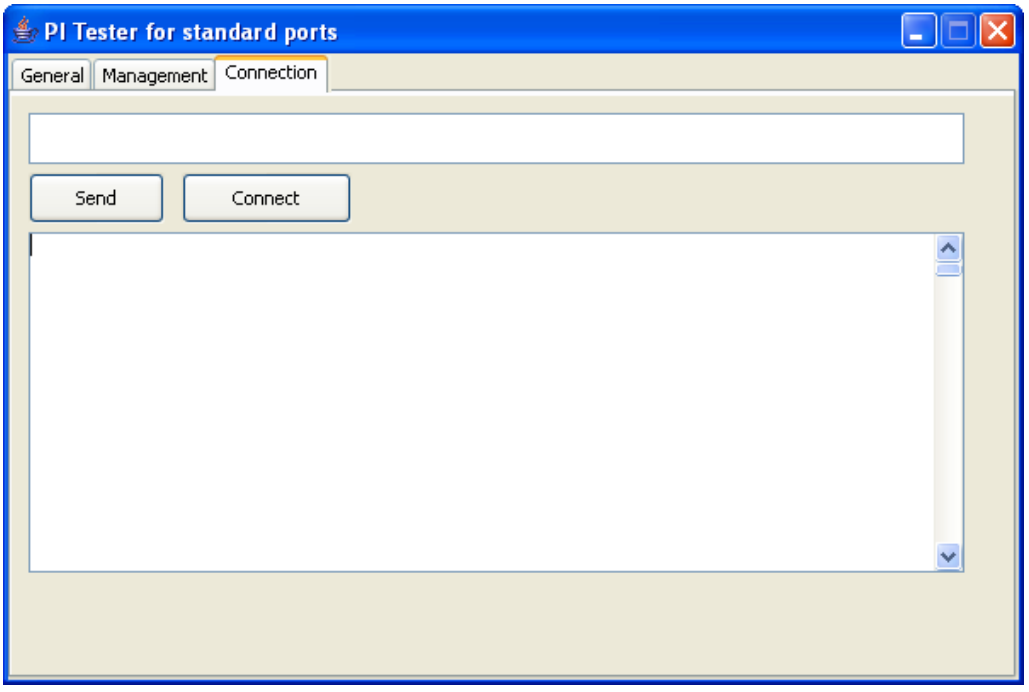
Follow these steps to test your user and PC connection

Step	Action
1	Give PI a moment to restart, then, through SMS: `http://`*`hostname of SMS`*`/`**`sms.jnlp`**, log in as your SMS user. Service Management System -> **Services -> Provisioning -> Tester**. 
2	On **General** tab, select your username from the drop-down list and enter the password.

Step	Action



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,********;
```

Step	Action
	<NACK,72-INVALIDLOGON - username,password;

ACS commands

Go to **Service Management System -> Provisioning -> 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 execute 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 that the previous timestamp recorded. Therefore, this SYNSTAMP value has to be manually incremented by '1' each time a new command is sent.

CSAdd CLI

ACSCLI=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:

```
ACSCLI=ADD:CLI=01206888888,CALLPLAN=callplan,ALLOWED=1|2|3|4|5,SYNSTAMP=200704110835
2387;
```

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 where those that belonged to the ACS customer 'Boss'.

ACS Query CLI (Calling Line Identifier)

ACSCLI=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:

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

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

```
<NACK,75-UNKNOWN COMMANDACSCLI=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_PROVID
ER=Boss,STATUS=A,CREATION_DATE=20070131175807,WALLET_EXPIRY_DATE=,BALANCE_EXPIRY_DAT
E=20070904165847,BALANCE=56815,INITIAL_BALANCE=100000,LANGUAGE=english,LAST_RECHARGE
_DATE=20070405150304,LAST_CC_RECHARGE_DATE=,LAST_USE_DATE=20070405150230,LAST_RECHAR
GE_AMOUNT=0,PREV_WALLET_EXPIRY_DATE=,PREV_BALANCE_EXPIRY_DATE=,PREV_BALANCE=0,LAST_E
XP_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=20
07041213581396;
```

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:

Go to **Subscriber Management -> 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.

Recharge Wallet

Use the CCSCD3=RCH - Recharge wallet command to change the value of a balance within the subscriber's wallet. This command has several mandatory parameters:

- MSISDN – Subscriber number (either MSISDN or Account must be entered)
- ACCOUNT – Subscriber account number (either MSISDN or Account must be entered)
- RECHARGE_TYPE – Credit or Custom
- REFERENCE – For Custom will be the 'Component' value defined in the Wallet Bonus Type screen
- AMOUNT – Recharge Amount (can be positive or negative)

The command also contains optional parameters which allow the operator to define how the wallet is recharged. Some of the variables are as follows:

- BALANCE_TYPE – For example General Cash, Promotional, SMS, etc
- WALLET_TYPE – Which wallet is recharged: Personal or Business
- BALMODE – Type of recharge – Delta or Absolute (default is delta)

If optional parameters are not defined the default values will be used for example default Wallet or default Balance Type as defined for subscriber.

See an example of the command entered below:

```
CCSCD3=RCH:MSISDN=44123457,BALANCE_TYPE=General
Cash,AMOUNT=1700,RECHARGE_TYPE=Custom,REFERENCE=10,MODE=2,
SYNSTAMP=2007041213581402;
```

If the wallet is recharged successfully the response should be similar to this:

```
<CCSCD3=RCH:ACK,SYNSTAMP=2007041213581402;
```

Change to the wallet balance can be verified on the **Service Management System -> Services -> Prepaid Charging -> Subscriber Management -> Edit Subscriber** -> Wallet screen. The CDR/EDR data can also be viewed by clicking on the **View CDRs** button. Different type of CDRs are created for positive (CDR8) and negative (CDR2) recharge values. More detailed information relating to the CDR can be seen in the **beServer.log** file.

Using the optional parameters attempt a number of recharges to your subscriber's wallet(s), applying the recharge to different balance types, positive and negative recharge values and using default and defined parameter values.

Recharge Bonus

A bonus set and a bonus type can be configured so that if a subscriber recharges their wallet by an amount that falls within the defined criteria they will receive a bonus percentage on top of their recharge amount.

Define a bonus value and then a bonus type using the tabs on the **Service Management System -> Services -> Prepaid Charging -> Wallet Management** screen. On the **Bonus Values** tab create a new bonus set. Define both the recharge range that will generate a bonus and the bonus percentage. Leave the **Voucher Type** field blank. Now create a new bonus type and link it to the bonus set, previously created. Define the balance type to which the bonus will be added. This can be a different balance to the one receiving the recharge. In the **Component** field enter a two character value, for example.12. This value must match the REFERENCE value sent in the PI Recharge command CCSCD3=RCH.

Send the PI command and verify that the recharge occurs successfully. Also verify that if recharge value falls within the defined bonus criteria the subscriber receives a bonus amount added to their defined balance type (view CDRs on Subscriber Management screens).

Chapter 8

Integration with ECE

Overview

Introduction

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

In this chapter

This chapter contains the following topics.

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 CCR messages.

Configuring the Set Active Domain Feature Node

To configure a Set Active Domain feature node:

Step	Action
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:

Step	Action
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.

Updating Convergent Charging Controller Balance Type Configuration for ECE

Perform the following updates to balance-type configuration 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* (on page 69).
- Update the subscriber-domain-type configuration in the **eserv.config** file. See *Subscriber-Domain-Type Configuration in eserv.config* (on page 71).
- Update the DCD-balance-query configuration in the **eserv.config** file. See *Balance-Query Configuration in eserv.config* (on page 71)
- Configure balance-type mapping between Convergent Charging Controller and ECE. See *Configuring Balance-Type Mapping Between Convergent Charging Controller and ECE* (on page 71).
- Configure balance-type announcements. See *Configuring Balance-Type Announcements* (on page 72).
- Configure balance-type translations for balance notifications. *Configuring Balance-Type Translations for Balance Notifications* (on page 73).
- Enable first-use policies for vouchers. See *Enabling First-Use Balance-Expiry Policies in Convergent Charging Controller* (on page 75).
- Define the conversion scale between Convergent Charging Controller currency values and ECE currency values. See *Defining the Conversion Scale Between Convergent Charging Controller and ECE Currency-Balance Values* (on page 75).

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

You configure balance-type mappings in Convergent Charging Controller so that balance values can be passed between Convergent Charging Controller and ECE. A balance-type mapping is used to map an Convergent Charging Controller balance type to the equivalent ECE balance type; for example, you would map the Convergent Charging Controller balance type for currency balances to the ECE balance ID **840**.

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

- The Convergent Charging Controller balance type that you want to map to an ECE balance ID
- The ECE balance ID
- The Diameter domain

You must configure a balance-type mapping for each ECE balance type that will be referenced in Diameter request and response messages that pass between Convergent Charging Controller and ECE.

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

Step	Action
1	In the SMS UI, open the Prepaid Charging, Service Management window.
2	From the **Service Provider** list, select the service provider you are using for external subscribers.
3	Select the **Balance Type Mappings** tab.
4	Click **New**. The **New Balance Type Mapping** dialog box opens.
5	From the **Balance Type** list, select the type of balance you want to map.
6	In the **Third-Party Resource** field, specify the ID of the ECE balance that you want to map to this Convergent Charging Controller balance type.
7	In the **Third-Party Resource Scale** field, enter `1`.
8	From the **Domain Type** list, select the Diameter domain that you configured previously. Select the `DIAMETER_Diameter` domConvergent Charging Controllerain type, for example.
9	Click **Save**.

See the discussion of service management in *CCS User's Guide* for more information about balance-type mapping configuration.

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.

Step	Action
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.

Step	Action
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:

Step	Action
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* (on page 73) for information about the available fields.
6	Click **Save**.

Balance type translations fields

This table describes the balance-type translation 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 *ACS User's Guide*.
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.
Dual Balance Currency Type	If required, you can select a different dual-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>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Translated Name</td><td>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.
Tip: The maximum number of characters allowed is 20.</td></tr>
<tr><td>Single Format for Balance</td><td>The structure of the data to send to the subscriber in short messages for a single (or first) currency.
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:
<code>%s %d %c %02d</code>
Where:<table><tr><th>Variable</th><th>Replaced by...</th><th>Example</th></tr><tr><td>%s</td><td>Currency Code</td><td>AUD</td></tr><tr><td>%d</td><td>Main unit of the currency</td><td>$</td></tr><tr><td>%c</td><td>Separator</td><td>.</td></tr><tr><td>%02d</td><td>Subunit of the currency</td><td>c</td></tr></table>
For balance types with non-cash units (for example, time balances are in seconds), you may only include the %d variable in the text in the Single Format for Balance field.<table><tr><th>For balance type unit</th><th>%d variable is replaced by</th></tr><tr><td>Time</td><td>Time balance in seconds</td></tr><tr><td>SMS</td><td>Number of named events</td></tr><tr><td>Bad PIN</td><td>Number of bad PIN attempts during the last 24 hours.</td></tr></table></td></tr>
<tr><td>Dual Format for Balance</td><td>The structure of the data to send to the subscriber in short messages for the second currency.
Note: This field is not available for non-cash balance types or balance types that have only one currency.</td></tr>
<tr><td>Separator</td><td>The character to use to separate the different balances when put together in an SMS Notification.
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:
<code>General Cash balance $20.00 , Promotional Cash balance $10.00 .</code></td></tr>
<tr><td>Balance Expiry</td><td>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.</td></tr>
<tr><td>No Balance Expiry</td><td>The text to output for balances that have no expiry date.</td></tr>
</table>

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 `ECEExtensions` Java applet parameter to **`true`** in the **sms.jnlp** file:

```
<param name="ECEExtensions" VALUE="true" />
```

Defining the Conversion Scale Between Convergent Charging Controller and ECE Currency-Balance Values

Convergent Charging Controller and ECE use different units to store currency-balance values:

- Convergent Charging Controller stores currency-balance values in small currency-units; for example, Convergent Charging Controller stores $1 as 100
- ECE stores currency-balance values in big currency-units; for example, ECE stores $1 as 1.0

DCD sends currency-balance values to ECE in CCR messages and receives currency-balance values from ECE in CCA messages. You define configuration to convert outgoing balances to the correct format for ECE and to convert incoming balances to the correct format for Convergent Charging Controller in the DCD section of the **eserv.config** file.

Defining the Conversion Scale for Balances in CCR Messages

DCD sends multiple balance values to ECE in CCR messages; for example, when sending voucher top-up requests. You configure DCD to convert the Convergent Charging Controller balance values to the format required by ECE by including the `conversionScale` parameter in the Value-Digits AVP:

```
conversionScale = scale_factor
```

where *scale_factor* is the integer value used to convert the Convergent Charging Controller balance value derived from the voucherInfoValue CCS concept into the format required by ECE. Specify:

- A negative value to divide the balance value by the *scale_factor* value
- A positive value to multiply the balance value by the *scale_factor* value

To convert Convergent Charging Controller balances (for example, dollar balances held in cents) to ECE balances (for example, dollar balances held in dollars), specify the following configuration:

```
conversionScale = -100
```

Note: DCD applies the value specified in the Value-Digits AVP for the `conversionScale` parameter to all the balance types sent in CCR messages. DCD makes no exception for balance types, such as SMS balances, that do not need converting because they do not use currency units.

Defining the Conversion Scale for Balances in CCA Messages

DCD receives multiple balance values from ECE in incoming CCA messages. You configure DCD to convert the ECE balance values to the format required by Convergent Charging Controller by including the `ccsBalanceScale` parameter in the DomainTypes, ccsBalanceScale section of the **eserv.config** file; for example:

```
DomainTypes = [
    ccsBalanceScale = [
        {
            conversionScale = scale_factor
            balance_type_id = balance_id
        }
    ]
]
```

where:

- `ccsBalanceScale` defines the scaling factor to apply to Convergent Charging Controller balance types in CCA messages. For each Convergent Charging Controller balance type, specify a combination of the `conversionScale` and `balance_type_id` parameters.
- *scale_factor* is an integer value that defines the scaling factor to apply to the balance value. Balances are multiplied by positive scaling factors and divided by negative scaling factors.
- *balance_id* is the ID of the Convergent Charging Controller balance type.

For example, the following configuration converts ECE dollar balance values to Convergent Charging Controller dollar balance values (in cents) for balance types with the ID 111, and does not apply any conversion to balance types with the ID 112:

```
DomainTypes = [
    ccsBalanceScale = [
        {
        # scale 1
            conversionScale = 100
            balance_type_id = 111
        }
        {
        # scale 2
            conversionScale = 1
            balance_type_id = 112
        }
    }
]
```

Note: DCD applies `ccsBalanceScale` configuration only to system and user balance values. Other balance-type values, such as balances impacted by ORA-Credit-Ceiling AVPs, are not converted.

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, AVPs section of the **eserv.config** file on SLC nodes. See the discussion of 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* (on page 77).
- Configure DCD to exclude ORA-Validity-End-Relative AVPs with the value 0 (zero). See *Excluding AVPs with Zero Values in Message Requests* (on page 77).
- 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* (on page 78).
- Configure mapping between Convergent Charging Controller voucher expiry periods and ECE expiry periods. See *Mapping Convergent Charging Controller Voucher-Expiry Periods to ECE Values* (on page 78).
- Configure ORA-Account-Topup AVPs to top up multiple balances. See *Voucher Top-Up Configuration for Multiple Balance Types* (on page 79).
- Configure ORA-Balance AVPs to handle balance queries that return multiple balances in CCR messages. See *Balance Query Response Configuration for Multiple Balances* (on page 80).
- Configure AVPs to prevent balance queries from returning invalid values. See *Preventing Balance Queries from Returning Invalid Values* (on page 80).
- Configure ORA-First-Usage-Validity AVPs. See *Configuring AVPs to Support First Use* (on page 80).
- Configure the ORA-Extend-Bucket-Validity AVP to send the correct new bucket information to ECE. See *New Bucket AVP Configuration* (on page 82).

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 = 12003
    ccsConcept = "voucherInfoValue"
    type = "Integer32"
    signInversion = true
    conversionScale = -100
}
```

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 = 218
    name = "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 = 219
    name = "ORA-Validity-Unit-End-Relative"
    ccsConcept = "voucherInfoBalanceExpiryExtensionType"
    type = "Enumerated"
    conversion = [
        {
            esg = 1          # CCC Expiry Unit Months
            vendor = 4       # ECE Expiry Unit Months
        }
        {
            esg = 0          # CCC Expiry Unit Hours
            vendor = 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 = 206
    name = "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 CCR message.

Balance Query Response Configuration for Multiple Balances

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

```
{
    avpCode = 208
    name = "ORA-Balance"
    type = "Grouped"
    repeating = true
    mandatoryContents = [
        "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* (on page 75) 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 = 217
    type = "Grouped"
    excludeWhenEmpty = true
    mandatoryContents = [
    ]
    optionalContents = [
        "ORA-Validity-First-Use-Cycle"
        "ORA-Validity-First-Use-With-Offset"
    ]
}

{
    ## Intentionally blank avpCode
    name = "ORA-Validity-First-Use-With-Offset"
    mandatoryContents = [
        "ORA-Validity-Offset-First-Use"
        "ORA-Validity-Unit-First-Use"
    ]
}

{
    avpCode = 218
    name = "ORA-Validity-Offset-First-Use"
    ccsConcept = "voucherInfoBalanceFirstUse"
    type = "Unsigned32"
    excludeWhenIn = "0"
}

{
    avpCode = 219
    name = "ORA-Validity-Unit-First-Use"
    ccsConcept = "voucherInfoBalanceExpiryExtensionType"
    type = "Enumerated"
}

{
    ## Intentionally blank avpCode
    name = "ORA-Validity-First-Use-Cycle"
    mandatoryContents = [
        "ORA-Validity-Type-First-Use-Cycle"
    ]
}

{
    avpCode = 219
    name = "ORA-Validity-Type-First-Use-Cycle"
    ccsConcept = "voucherInfoBalanceExpiryExtensionPolicy"
    type = "Enumerated"
    # 6 - firstAccountCycle
    # 7 - firstBillCycle
    includeWhenIn = "6,7"
    conversion = [
        {
            esg = 6     # CCC Expiry Policy - First Use Account Cycle
            vendor = 5  # ECE - First Use Account Cycle
        }
        {
            esg = 7     # CCC Expiry Policy - First Use Bill Cycle
            vendor = 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 = 228
    name = "ORA-Extend-Bucket-Validity"
    ccsConcept = "voucherInfoNewBucket"
    type = "Enumerated"
    conversion = [
        {
            esg = 0
            vendor = 1
        }
        {
            esg = 1
            vendor = 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.

Glossary of Terms

AAA

Authentication, Authorization, and Accounting. Specified in Diameter RFC 3588.

ACS

Advanced Control Services configuration platform.

ANI

Automatic Number Identification - Term used in the USA by long-distance carriers for CLI.

API

Application Programming Interface

AVP

Attribute Value Pair, used in Diameter to represent properties of a particular request or answer.

BCSM

Basic Call State Model - describes the basic processing steps that must be performed by a switch in order to establish and tear down a call.

C7

See SS7.

CAMEL

Customized Applications for Mobile network Enhanced Logic

This is a 3GPP (Third Generation Partnership Project) initiative to extend traditional IN services found in fixed networks into mobile networks. The architecture is similar to that of traditional IN, in that the control functions and switching functions are remote. Unlike the fixed IN environment, in mobile networks the subscriber may roam into another PLMN (Public Land Mobile Network), consequently the controlling function must interact with a switching function in a foreign network. CAMEL specifies the agreed information flows that may be passed between these networks.

CAP

CAMEL Application Part

CC

Country Code. Prefix identifying the country for a numeric international address.

CCA

Credit-Control-Answer, used in Diameter by the credit-control server to acknowledge a Credit-Control-Request (CCR) from the credit-control client.

CCR

Credit-Control-Request, used in Diameter by the credit-control client to request credit authorization from the credit-control server.

CCS

1) Charging Control Services (or Prepaid Charging) component.

2) Common Channel Signalling. A signalling system used in telephone networks that separates signalling information from user data.

CDR

Call Data Record

Note: The industry standard for CDR is EDR (Event Detail Record).

CLI

Calling Line Identification - the telephone number of the caller. Also referred to as ANI.

Connection

Transport level link between two peers, providing for multiple sessions.

Convergent

Also “convergent billing”. Describes the scenario where post-paid and pre-paid calls are handed by the same service platform and the same billing system. Under strict converged billing, post-paid subscribers are essentially treated as “limited credit pre-paid”.

CPE

Control Plan Editor (previously Call Plan Editor) - software used to define the logic and data associated with a call -for example, "if the subscriber calls 0800 *nnnnnn* from a phone at location *xxx* then put the call through to *bb bbb bbbb*".

CS1

ETSI INAP Capability Set 1. An ITU standard.

Diameter

A feature rich AAA protocol. Utilises SCTP and TCP transports.

DP

Detection Point

DTMF

Dual Tone Multi-Frequency - system used by touch tone telephones where one high and one low frequency, or tone, is assigned to each touch tone button on the phone.

ETSI

European Telecommunications Standards Institute

FCI

Furnish Charging Information. An INAP operation sent from ACS to the SSP to control the contents of EDRs produced by the SSP.

FDA

First Delivery Attempt - the delivery of a short message directly to the SME rather than relaying it through the MC.

GPRS

General Packet Radio Service - employed to connect mobile cellular users to PDN (Public Data Network- for example the Internet).

GSM

Global System for Mobile communication.

It is a second generation cellular telecommunication system. Unlike first generation systems, GSM is digital and thus introduced greater enhancements such as security, capacity, quality and the ability to support integrated services.

GT

Global Title.

The GT may be defined in any of the following formats:

- Type 1: String in the form "1,<noa>,<BCD address digits>"
- Type 2: String in the form "2,<trans type><BCD address digits>"
- Type 3: String in the form "3,<trans type>,<num plan>,<BCD address digits>"
- Type 4: String in the form "4,<trans type>,<num plan>,<noa>,<BCD address digits>"

The contents of the Global Title are defined in the Q713 specification, please refer to section 3.4.2.3 for further details on defining Global Title.

GUI

Graphical User Interface

HLR

The Home Location Register is a database within the HPLMN (Home Public Land Mobile Network). It provides routing information for MT calls and SMS. It is also responsible for the maintenance of user subscription information. This is distributed to the relevant VLR, or SGSN (Serving GPRS Support Node) through the attach process and mobility management procedures such as Location Area and Routing Area updates.

HPLMN

Home PLMN

IDP

INAP message: Initial DP (Initial Detection Point)

IMSI

International Mobile Subscriber Identifier. A unique identifier allocated to each mobile subscriber in a GSM and UMTS network. It consists of a MCC (Mobile Country Code), a MNC (Mobile Network Code) and a MSIN (Mobile Station Identification Number).

The IMSI is returned by the HLR query (SRI-SM) when doing FDA. This tells the MSC exactly who the subscriber is that the message is to be sent to.

IN

Intelligent Network

INAP

Intelligent Network Application Part - a protocol offering real time communication between IN elements.

Initial DP

Initial Detection Point - INAP Operation. This is the operation that is sent when the switch reaches a trigger detection point.

IP

1) Internet Protocol

2) Intelligent Peripheral - This is a node in an Intelligent Network containing a Specialized Resource Function (SRF).

IP address

Internet Protocol Address - network address of a card on a computer.

ISDN

Integrated Services Digital Network - set of protocols for connecting ISDN stations.

ITU

International Telecommunication Union

IVR

Interactive Voice Response - systems that provide information in the form of recorded messages over telephone lines in response to user input in the form of spoken words or, more commonly, DTMF signalling.

MAP

Mobile Application Part - a protocol which enables real time communication between nodes in a mobile cellular network. A typical usage of the protocol would be for the transfer of location information from the VLR to the HLR.

MC

Message Centre. Also known as SMSC.

MCC

Mobile Country Code. In the location information context, this is padded to three digits with leading zeros. Refer to ITU E.212 ("Land Mobile Numbering Plan") documentation for a list of codes.

MNC

Mobile Network Code. The part of an international address following the mobile country code (MCC), or at the start of a national format address. This specifies the mobile network code, that is, the operator owning the address. In the location information context, this is padded to two digits with a leading zero. Refer to ITU E.212 ("Land Mobile Numbering Plan") documentation for a list of codes.

MO

Mobile Originated

MS

Mobile Station

MSC

Mobile Switching Centre. Also known as a switch.

MSIN

Mobile Station Identification Number.

MSISDN

Mobile Station ISDN number. Uniquely defines the mobile station as an ISDN terminal. It consists of three parts; the country code (CC), the national destination code (NDC) and the subscriber number (SN).

MT

Mobile Terminated

NOA

Nature Of Address - a classification to determine in what realm (Local, National or International) a given phone number resides, for the purposes of routing and billing.

PC

Point Code. The Point Code is the address of a switching point.

PI

Provisioning Interface - used for bulk database updates/configuration instead of GUI based configuration.

PIN

Personal Identification Number

PL/SQL

Oracle's Procedural Language for stored procedures and packages.

PLMN

Public Land Mobile Network

SCCP

Signalling Connection Control Part (part of the SS7 protocol stack).

SCI

Send Charging Information. An INAP operation sent from ACS to the SSP to control real time charging by the SSP.

SCTP

Stream Control Transmission Protocol. A transport-layer protocol analogous to the TCP or User Datagram Protocol (UDP). SCTP provides some similar services as TCP (reliable, in-sequence transport of messages with congestion control) but adds high availability.

Service Provider

See Telco.

SGSN

Serving GPRS Support Node

SIM

Usually referred to as a SIM card, the Subscriber Identity Module is the user subscription to the mobile network. The SIM contains relevant information that enables access onto the subscripted operator's network.

SK

Service Key

SLC

Service Logic Controller (formerly UAS).

SLEE

Service Logic Execution Environment

SME

Short Message Entity - an entity which may send or receive Short Messages. It may be located in a fixed network, a mobile, or an SMSC.

SMS

Depending on context, can be:

- Service Management System hardware platform
- Short Message Service
- Service Management System platform
- Convergent Charging Controller Service Management System application

SN

Service Number

SQL

Structured Query Language - a database query language.

SRF

Specialized Resource Function – This is a node on an IN which can connect to both the SSP and the SLC and delivers additional special resources into the call, mostly related to voice data, for example play voice announcements or collect DTMF tones from the user. Can be present on an SSP or an Intelligent Peripheral (IP).

SRI

Send Routing Information - This process is used on a GSM network to interrogate the HLR for subscriber routing information.

SS7

A Common Channel Signalling system used in many modern telecoms networks that provides a suite of protocols which enables circuit and non circuit related information to be routed about and between networks. The main protocols include MTP, SCCP and ISUP.

SSF

Sub Service Field.

SSP

Service Switching Point

Switching Point

Anything that can send and receive C7 messages.

System Administrator

The person(s) responsible for the overall set-up and maintenance of the IN.

TCAP

Transaction Capabilities Application Part – layer in protocol stack, message protocol.

TCP

Transmission Control Protocol. This is a reliable octet streaming protocol used by the majority of applications on the Internet. It provides a connection-oriented, full-duplex, point to point service between hosts.

TDP

Trigger Detection Point.

Telco

Telecommunications Provider. This is the company that provides the telephone service to customers.

Telecommunications Provider

See Telco.

Termination Number

The final number that a call terminates to. Can be set in control plan nodes such as Attempt Termination and Unconditional Termination for re-routing numbers such as Toll Free or Follow Me numbers.

VLR

Visitor Location Register - contains all subscriber data required for call handling and mobility management for mobile subscribers currently located in the area controlled by the VLR.

VWS

Oracle Voucher and Wallet Server (formerly UBE).

Index

A

B

C

D

T

U

V

W