Oracle® Communications Convergent Charging Controller

CCS Provisioning Interface Commands

Release 6.0

May 2016

Copyright

Contents

Chapter 4

Chapter 5

Chapter 6

About This Document

Scope

The scope of this document includes all the information required to configure the Provisioning Interface commands.

Audience

The audience for this document includes system administrators responsible for the monitoring, maintenance, and configuration of the Oracle Communications Convergent Charging Controller IN applications.

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.

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

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

Related Documents

The following documents are related to this document:

- *Oracle Communications Convergent Charging Controller Provisioning Interface User's and Technical Guide*
- *Oracle Communications Convergent Charging Controller Virtual Private Network User's Guide*

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

PI Commands Overview

Overview

Introduction

The provisioning interface (PI) uses TCP/IP-based UNIX sockets to receive provisioning commands and parameters. These are translated into SQL commands that update prepaid application tables of the SMF and E2BE Oracle databases. This chapter defines the rules and packages required to translate the provisioning commands into SQL commands.

In this chapter

This chapter contains the following topics.

Command List

Command list

The following table lists commands and corresponding definitions. Each definition requires one or two of the optional packages to have been run at installation time.

Example: To install the commands to recharge accounts and buckets, install the `piWalletSms` and `piVoucherSms` packages.

Function	piSubscriberSms	piWalletSms	piVoucherSms
Add a subscriber	CCSCD1=ADD	CCSCD1=ADD	
Change subscriber details	CCSCD1=CHG	CCSCD1=CHG	
Delete a subscriber	CCSCD1=DEL	CCSCD1=DEL	
Query a subscriber	CCSCD1=QRY	CCSCD1=QRY	
Change call barring details	CCSCD4=CHG	CCSCD4=CHG	
Change credit card details	CCSCD5=CHG		
Change credit card status	CCSCD6=CHG		
Add an F&F/F&D number	CCSCD8=ADD	CCSCD8=ADD	
Change F&F/F&D numbers	CCSCD8=CHG	CCSCD8=CHG	
Delete an F&F/F&D number	CCSCD8=DEL		
Change subscriber profile	CCSCD9=CHG		
Query subscriber profile	CCSCD9=QRY		
Add a CLI to a closed user group	CCSCG1=ADD		

Delete a CLI from a closed user group	CCSCG1=DEL		
Add a closed user group	CCSCG2=ADD		
Change a closed user group	CCSCG2=CHG		
Delete a closed user group	CCSCG2=DEL		
Perform CCS business process logic	CCSBPL=EXE		
Change Subscriber Searchable Fields	CCSSF1=CHG		
Query Subscriber Searchable Fields	CCSSF1=QRY		
Merge subscriber wallets		CCSCD1=MSW	
Query wallet users		CCSCD2=QRY	
Recharge an account or bucket		CCSCD3=RCH	CCSCD3=RCH
Credit transfer		CCSCD3=CTR	
Query EDRs for an account		CCSCD7=QRY	
Change credit card recharge request status		CCSCC1=CHG	
Query credit card recharge requests		CCSCC1=QRY	
Add service charge		CCSSC1=ADD	
Query service charge		CCSSC1=QRY	
Add charge for call		CCSSC2=ADD	
Query subscribed periodic charges		CCSSC3=QRY	
Change voucher status			CCSVR1=CHG
Mark voucher frozen			CCSVR1=FRZ
Query a recharge voucher			CCSVR1=QRY

PIClientIF command list

The following table lists commands and corresponding definitions for the Social Networking Service Template service. Each definition requires the optional SNST package to have been run at installation time.

Function	Command
Add an SNST friend	CCSSR1=ADD
Remove an SNST friend	CCSSR1=DEL

About Service Provider Restrictions

For security reasons, the data that a PI user can query or modify is restricted by service provider. This means that you can run PI commands to query or modify the data only for a service provider that is associated with your PI user.

The system administrator specifies which service providers to associate with your PI user on the **Users** tab in the Administration screen in the PI UI. For more information, see *PI User's and Technical Guide*.

Attempts to run PI commands for a service provider that is not associated with your PI user will result in a negative acknowledgement (NACK) message being returned.

Parameter Formats

Formats

This table describes the format of each PI parameter.

Note: Lengths are largely determined by the size of fields in the database, with exceptions noted in the table.

Parameter	Format
ACCOUNT_NUMBER	20 digit number
ACS_CUST	50 character string
ADD	String: “CLI1|CLI2|…”
AMOUNT	Number (-2147483648 to 2147483647) (small currency) **Note:** Length determined by the length of a signed 32 bit number
AUTH_HASH_FN_NAME	50 character string
BALANCE	18 digit number (small currency) **Note:** Length determined by the length of a signed 32 bit number
BALANCE_EXPIRY	0 to 999 (days)
BALANCE_TYPE	50 character string
BALMODE	“ABSOLUTE” or “DELTA”
BE_ID	10 digit number (>0)
BPL	50 character string
CC_ALLOWED	“Y” or “N”
CC_EXPIRY_DATE	YYYYMMDD
CC_NUMBER	20 digit number
CC_PIN_RESET	“Y” or blank
CT_TYPE	50 character string
CHARGE (CCSSC1)	Number (-2147483648 to 2147483647) (small currency) **Note:** Length determined by the length of a signed 32 bit number
CHARGE (CCSSC3)	50 character string
CHARGING_DOMAIN	28 digit number
CLASS	50 character string
CLEAR	“Y” or “N”
CREATE_CUG	“Y” or “N”
CREDIT_LIMIT	Number (-2147483648 to 2147483647) (small currency) **Note:** Length determined by the length of a signed 32 bit number

Parameter	Format
CUG	50 character string
CURRENCY	3 character string
DATE	YYYYMMDDHHMMSS (24 hour clock)
DAYS	Integer >=0
DEL	String: “MSISDN1|MSMIDN2|…”
DEST_MSISDN	18 digit number (>0) **Note:** MSISDN is 2 digits less than ACCOUNT_NUMBER due to the 2 digit account prefix.
DEST_WALLET	50 character string
DURATION	Number of seconds
EDR_TYPE	Numbers separated by pipe symbols
END_DATE	YYYYMMDDHHMMSS (24 hour clock)
END_MSISDN	18 digit number (>0), same length and less than or equal to START_MSISDN **Note:** MSISDN is 2 digits less than ACCOUNT_NUMBER due to the 2 digit account prefix.
EVENT	50 character string
EXTn (n is 1 to 10)	160 character string
EXTRA_EDR	String of EDR fields to add: “<KEY1>=<VALUE1>|<KEY2>=<VALUE2>|...” **Note:** <KEY*n*> and <VALUE*n*> must not contain the comma (‘,’), bar (‘|’) or semi-colon (‘;’) characters. Additionally, <KEY*n*> must not contain the equals (‘=’) character. The length of the EXTRA_EDR data should not exceed 100 characters.
FDNUM	20 digit number
FFLIST	String: “FFNUM1|FFNUM2|FFNUM3…”
FFNUM	20 digit number
FRIEND	MSISDN - 18 digit number
FRIEND_LIST	String: "CLI1|CLI2|CLI3..."
FROM	20 digit number
IGNORE	“Y” or “N”
IMSI	15 digit number (>0)
LANGUAGE	50 character string
LIMIT_TYPE	“Debit”, “Single Use”, “Credit” or “Limited Credit”
LIST_TYPE	String (CCSCD1=QRY, CCSSR1=QRY) or “A” or “B” (CCSCD4=CHG)
LW_ACCOUNT_NUMBER	20 digit number
LW_MSISDN	18 digit number (>0) **Note:** MSISDN is 2 digits less than ACCOUNT_NUMBER due to the 2 digit account prefix.
MAX_CONCURRENT_ACCESS	Number (0 to 429467295) **Note:** Length determined by the length of an unsigned 32 bit number

Parameter	Format
MAX_RECORDS	Integer 1 to 99 (CCSCC1=QRY) Integer >1 (CCSCD7=QRY)
MODE	1,2,3 or 4
MSISDN	18 digit number (>0) **Note:** MSISDN is 2 digits less than ACCOUNT_NUMBER due to the 2 digit account prefix.
MSISDN1	18 digit number (>0) **Note:** MSISDN is 2 digits less than ACCOUNT_NUMBER due to the 2 digit account prefix.
MSISDN2	18 digit number (>0) **Note:** MSISDN is 2 digits less than ACCOUNT_NUMBER due to the 2 digit account prefix.
NEW_FFNUM	20 digit number
NEW_MSISDN	18 digit number (>0) **Note:** MSISDN is 2 digits less than ACCOUNT_NUMBER due to the 2 digit account prefix.
NICKNAME	50 character string
ORIG_MSISDN	18 digit number (>0) **Note:** MSISDN is 2 digits less than ACCOUNT_NUMBER due to the 2 digit account prefix.
ORIG_WALLET	50 character string
PIN	4 digit decimal number. It may start with 0, for example, 0123
PRODUCT	50 character string
PRODUCT_TYPE	50 character string
PROVIDER	50 character string
RECHARGE_TYPE	“Voucher”, “Credit” or “Custom” (not case sensitive)
REFERENCE	See CCSCD3=RCH / CCSCC1
REFUND	“1” or “0”
SCENARIO	“1” to “9”
START_DATE	YYYYMMDDHHMMSS (24 hour clock)
START_MSISDN	18 digit number (>0), same length and greater than or equal to END_MSISDN **Note:** MSISDN is 2 digits less than ACCOUNT_NUMBER due to the 2 digit account prefix.
STATUS	Single character
TAG	See CCSCD9=CHG or CCSCD9=QRY
TARIFF_PLAN	50 character string
TO	20 digit number
TRACKING_DOMAIN	28 digit number

Parameter	Format
TYPE	“F”, “D” or “N” (CCSCD8=CHG)
VALUE	See CCSCD9=CHG
VOUCHER	20 digit number
WALLET_EXPIRY	0 to 999 (days)
WALLET_EXPIRY_DATE	YYYYMMDDHHMMSS (24 hour clock)
WALLET_REFERENCE	50 character string
WALLET_TYPE	50 character string

Chapter 2

PI Subscriber Package

Overview

Introduction

This chapter describes the Oracle Communications Convergent Charging Controller provisioning interface (PI)I commands for provisioning subscriber information on the SMS.

These commands are added by the `piSubscriberSms` package.

In this chapter

This chapter contains the following topics.

Add a Subscriber

Description

Use the command CCSCD1 and action ADD to add a new subscriber or range of subscribers, and set whether to use the system language for new subscribers or the subscriber's language to the SMF database. It executes the functionality provided by the **Subscriber** tab on the Subscriber management screen.

Note: If the `piWalletSms` package has been installed, you can also use this command to add new wallets or links to existing wallets.

Required parameters

This table describes each required parameter.

Parameter	Description
PROVIDER	Service provider name
PRODUCT	Product type name

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
MSISDN	The CLI for the new subscriber
START_MSISDN	The starting CLI for a range of CLIs
END_MSISDN	The final CLI in the range.
CHARGING_DOMAIN	The reference id of the charging domain for this subscriber.
WALLET_REFERENCE	The external wallet reference (Third party domains only).

Optional parameters

CCSCD1=ADD accepts the following optional parameters.

`LANGUAGE`

Syntax: `LANGUAGE="`*`language_name`*`"`

Description: The language short name. The language must exist already in the language table.

Type: String

Optionality: Optional (default used if not set).

Allowed: Up to 50 characters

Default: Language of the logged PI user if this is defined. Otherwise the system default language is used.

Notes:

Example: `LANGUAGE="French"`

`BYPASS_NUMBER`

Syntax: `BYPASS_NUMBER=` *`number`*

Description: Used by the operator to link subscriber accounts

Type: Integer

Optionality: Optional.

Allowed:

Default:

Notes:

Example:

`PIN`

Syntax: `PIN=`*`number`*

Description: The subscriber's private PIN.

Type: Integer

Optionality: Optional

Allowed: Any decimal number between 4 and 10 digits long

Default:

Notes:

Example: `PIN=4812`

ACCOUNT_NUMBER

Syntax: `ACCOUNT_NUMBER=`*reference_number*

Description: The reference number (subscriber id) for the subscriber's account.

Type: Integer

Optionality: Optional.

Allowed: Any number up to 20 digits long

Default:

Notes:

Example: `ACCOUNT_NUMBER=1047768832`

TRACKING_DOMAIN

Syntax: `TRACKING_DOMAIN=`*domain_id*

Description: The reference id for the TRACKING_DOMAIN.

Type: Integer

Optionality: Optional

Allowed: A number up to 28 digits long. Must be the reference id for an existing tracking domain.

Default: None

Notes: For a subscriber to be associated with a tracking domain, this parameter must be present.

Example: `TRACKING_DOMAIN=5`

Logic and constraints

Here are the general rules that apply when using the CCSCD1=ADD command to create a new subscriber or a range of subscribers.

1. To create a single subscriber:
   - You must specify the MSISDN. This must not exist already.
   - You must not specify the START_MSISDN and END_MSISDN.
   - You can specify the ACCOUNT_NUMBER. The ACCOUNT_NUMBER (prepended with account prefix) must not exist already.
   - You must not specify LW_MSISDN and LW_ACCOUNT_NUMBER.
2. To create a range of subscribers:
   - You must specify START_MSISDN and END_MSISDN, but MSISDN must not be specified.
   - START_MSISDN must be a lower number than or equal to END_MSISDN and they must be the same length.
   - The MSISDNs within the range must not exist already.
   - If the system is unable to create any subscriber within the range, none of the subscribers in the range will be created.

- Each subscriber will have the same PROVIDER, PRODUCT, CHARGING_DOMAIN, LANGUAGE, BYPASS_NUMBER and PIN, if specified.
- The maximum size of a range is 1000 MSISDNs.

3 If specified, the PROVIDER must be a defined service provider and PRODUCT must be an existing product for the service provider.

4 If LANGUAGE is not specified, then if defined, the language of the logged in PI user will be is used, otherwise the system default language will be used.

5 If you do not specify the CHARGING_DOMAIN, the subscriber will be created on a VWS domain.

6 If you do not specify the TRACKING_DOMAIN, the subscriber will not be associated with a tracking domain.

7 If you do not specify the PIN, the last 4 digits of the generated account number will be used for the PIN.

VWS domain

Here are the additional rules that apply when creating subscribers on a VWS domain.

1 If you specify the CHARGING_DOMAIN, it must match a defined VWS domain.

2 The WALLET_REFERENCE parameter will be ignored.

Third party domain

Here are the additional rules that apply when creating subscribers on a third party domain.

1 You must specify a CHARGING_DOMAIN that matches a defined non-VWS domain.

2 If you do not specify the WALLET_REFERENCE, it is generated automatically. The WALLET_REFERENCE will not be validated.

3 To specify the WALLET_REFERENCE when creating a range of subscribers, specify either:

- a pipe ("|") separated list of references, one for each subscriber
- a single WALLET_REFERENCE

4 If only one WALLET_REFERENCE is supplied, each subscriber will get the same WALLET_REFERENCE.

5 If many WALLET_REFERENCEs are supplied, the number or references must match the number of CLIs in the range.

Tracking domain

Here are the additional rules that apply when creating subscribers and a tracking domain is required. If you:

1 Specified a VWS CHARGING_DOMAIN, then you must specify the same VWS ID for the TRACKING_DOMAIN.

2 Specified a third party CHARGING_DOMAIN, then you must specify a VWS ID for the TRACKING_DOMAIN.

3 Did not specify a CHARGING_DOMAIN, then the CHARGING_DOMAIN will be set to the VWS ID specified for the TRACKING_DOMAIN.

Error codes

1, 3, 6, 13, 65, 68, 69, 117, 118, 119, 120, 132, 133 and 134.

Refer to *PI Command Errors* (on page 82) for a description of each error.

Change Subscriber Details

Description

Use the command CCSCD1 and action CHG to change the details associated with an existing subscriber record in the SMF database.

Note: If the piWalletSms package has been installed then you can also use this command to modify wallet and balance details.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account that will be changed.
LANGUAGE	The language short name of up to 50 characters in length. The language must exist already in the language table.
WALLET_REFERENCE	The external wallet reference (Third party domains only). This can be up to 50 characters in length.
PIN	The PIN for the subscriber. This is a four digit number.

NEW_MSISDN

Description:	A new CLI for the subscriber
Type:	Integer
Allowed:	An 18 digit number which must be > 0
Default:	None
Notes:	Must not exist already.

Optional parameters

No optional parameters are supported.

Logic and constraints

The following rules apply when using the CCSCD1=CHG command:

1. You must specify an existing MSISDN.
2. You must specify at least one constraint required parameter in addition to the MSISDN.
3. For VWS domains, the WALLET_REFERENCE is not required and if supplied, it will be ignored.
4. For third party domains, the WALLET_REFERENCE is not validated.

Note: You cannot delete a WALLET_REFERENCE.

Delete a Subscriber

Description

Use the command CCSCD1 and action DEL to delete an existing subscriber record from the SMF database.

Note: If the `piWalletSms` package has been installed, you can also use this command to delete wallet and balance details for a subscriber.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

No optional parameters are supported.

Logic and constraints

The following rule applies when using the CCSCD1=DEL command:

1. You must specify an existing MSISDN.

Query a Subscriber

Description

Use the command CCSCD1 and action QRY to query the SMF database and for details on a specified subscriber.

Note: If the `piWalletSms` package has been installed, you can also use this command to return details on the wallets and balances for the specified subscriber.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Parameter	Description
LIST_TYPE	Pipe ("|") separated list of items to return. Valid values are: • BALANCE • BALANCE_TYPE • BYPASS_NUMBER • CHARGING_DOMAIN • CREATION_DATE • CUG • FDN • FFD • FFN • FREE_SWAPS_REMAINING • LANGUAGE • LAST_SWAP_RESET_DATE • PRODUCT • SERVICE_PROVIDER • TRACKING_DOMAIN • UNRESERVED_BALANCE • WALLET_REFERENCE • WALLET_TYPE

Logic and constraints

The following rules apply when using the CCSCD1=QRY command:

1. You must specify an existing MSISDN.
2. If a parameter value is null, the returned field will be empty.
3. For VWS charging domains, if LIST_TYPE is not specified then all values except the unreserved balance will be returned.
4. For third party charging domains, if LIST_TYPE is not specified then only the MSISDN and the ACCOUNT_NUMBER will be returned.
5. All dates are returned in the 24 hour format YYYYMMDDHHMMSS if set, or blank if not set.

Change Call Barring Details

Description

Use the command CCSCD4 and action CHG to change the allowed or barred numbers for an existing subscriber record in the SMF database.

Note: If the `piWalletSms` package has been installed, you can also use this command to create EDRs for a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. You must specify an existing MSISDN.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
LIST_TYPE	Sets whether this is a list of allowed or barred numbers. Valid values are: • "A" - Allowed • "B" - Barred
IGNORE	Sets whether or not the call barring list should be ignored. Valid values are: • "Y" - True • "N" - False
ADD	The list of MSISDNs to be added to the existing call barring list. This will be a pipe ("|") separated list of up to 50 MSISDNs.
DEL	The list of MSISDNs to be deleted from the existing call barring list. This will be a pipe ("|") separated list of up to 50 MSISDNs.
CLEAR	Either "Y" or "N". Defaults to "N". When set to "Y" the existing call barring list is cleared and then any MSISDNs in the ADD list are added. The DEL parameter is ignored.

Optional parameters

No optional parameters are supported.

Logic and constraints

The following rules apply when using the CCSCD4=CHG command:

1. You must specify an existing MSISDN.
2. You must specify at least one constraint required parameter.
3. Prior to this command, the call barring list must not include any MSISDNs from the ADD list.
4. Prior to this command, the call barring list must include all the MSISDNs in the DEL list.

Change Credit Card details

Description

Use the command CCSCD5 and action CHG to change the credit card details for an existing subscriber record in the SMF database. You use it to specify a new credit card number and expiry date for the subscriber.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. You must specify an existing MSISDN.
CC_NUMBER	The new credit card number. This can be up to 20 digits long.
CC_EXPIRY_DATE	The new credit card expiry date. This must be a valid date in the format: YYYYMMDD.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

No optional parameters are supported.

Logic and constraints

To successfully execute the CCSCD5=CHG command, you must supply valid values for all the required parameters.

Change Credit Card Status

Description

Use the command CCSCD6 and action CHG to change the credit card status for a subscriber. You can set the credit card recharge allowed flag, and you can reset the recharge authorization PIN.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
CC_ALLOWED	Either "Y" or "N". Determines whether or not to allow credit card recharges.
CC_PIN_RESET	When set to "Y", the users recharge pin is set to NULL and the CC start process indicator is set to "N". Valid value: "Y".

Optional parameters

No optional parameters are supported.

Logic and constraints

The following rules apply when using the CCSCD6=CHG command:

1. You must specify an existing MSISDN.
2. You must specify a valid value for either the CC_ALLOWED or the CC_PIN_RESET parameters.

Add an F&F/F&D Number

Description

Use the command CCSCD8 and action ADD to add a new Friends & Family or Friends & Destination number to an existing subscriber record in the SMF database.

Note: If the `piWalletSms` package has been installed, you can also use this command to create EDRs for a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
FFNUM	The F&F number to be added. This can be up to 20 digits long. **Note:** The number specified must not be on the F&F numbers barred list.
FDNUM	The F&D number to be added. This can be up to 20 digits long. **Note:** The number specified must not be on the F&D numbers barred list.

Optional parameters

No optional parameters are supported.

Logic and constraints

The following rules apply when using the CCSCD8=ADD command:

1. You must specify an existing MSISDN.
2. You must specify either the FFNUM or the FDNUM parameter (not both).
3. If the F&F numbers white list is not set to be ignored and you are adding an F&F number, then the value specified for FFNUM must match a prefix in the F&F white list.

4 When adding an F&D number, the value specified for FDNUM must exist already in the list of prefixes defined in the Prepaid Charging F&D prefix map.

5 Once specified, an F&D number cannot be re-specified.

Note: Each time an F&F number or an F&D number is added, the change count is automatically increased by one and the change date/time is set to "Now".

Change F&F/F&D Numbers

Description

Use the command CCSCD8 and action CHG to change the Friends & Family or Friends & Destination details for a subscriber.

Note: If the `piWalletSms` package has been installed, you can also use this command to create EDRs for a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
FFNUM	The F&F number to change.
NEW_FFNUM	The new F&F number. This can be up to 20 digits long. **Note:** The number specified must not be on the F&F numbers barred list.
TYPE	Sets the type of service. Valid values are: • "F" – Friends & Family • "D" – Friends & Destination • "N" – None
FDNUM	The F&D number to set the Active Discount Prefix. This can be up to 20 digits long. **Note:** The number specified must not be on the F&D numbers barred list.
FFLIST	Pipe ("|") separated list of F&F numbers. Use to set the list exactly.

Optional parameters

No optional parameters are supported.

Logic and constraints

The following rules apply when using the CCSCD8=CHG command:

1 You must specify an existing MSISDN.
2 You must specify at least one constraint required parameter.
3 To change a single F&F number, specify the FFNUM and MSISDN parameters only.
4 To set the entire F&F number list, specify the FFLIST and MSISDN parameters only.
5 When setting F&F numbers, if the F&F numbers white list is not set to be ignored, then new F&F number(s) must match a prefix in the F&F white list.
6 To change the service type, specify the MSISDN and Type parameters only.
7 To set the F&D number, specify the MSISDN and FDNUM parameters only.
8 When setting the F&D number, the value specified for FDNUM must exist already in the list of prefixes defined in the Prepaid Charging F&D prefix map.
9 Once specified, an F&D number cannot be re-specified.

Note: Each time an F&F number or an F&D number is added, the change count is automatically increased by one and the change date/time is set to "Now".

Delete an F&F/F&D Number

Description

Use the command CCSCD8 and action DEL to delete a Friends & Family or Friends & Destination number for a subscriber.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
FFNUM	The F&F number to delete.
FDNUM	The F&D number to delete.

Optional parameters

No optional parameters are supported.

Logic and constraints

The following rules apply when using the CCSCD8=DEL command:

1 You must specify an existing MSISDN.
2 You must specify either the FFNUM or the FDNUM parameter (not both).
3 When deleting an F&D number for a subscriber, the Discount Prefix is set to the default value of "N/A".

Note: Deleting an F&F number or an F&D number does not affect the change count as these changes are always free of charge.

Change Subscriber Profile

Description

Use the command CCSCD9 and action CHG to change entries in the subscriber profile.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.
TAG	This is a predefined profile tag name corresponding to a PROFILE_TAG_NAME in the ACS_PROFILE_DETAILS table. More than one tag may be specified. They should be separated by the pipe ("|") symbol.
VALUE	The value for the tag. This must match the tag data type. If more than one tag is specified, tag values must be pipe separated and listed in the same order as for TAG.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

CCSCD9=CHG command accepts the following optional parameters.

`PTMODE`

Syntax: `PTMODE=`*`ptupdatemode`*

Description: Specifies the type of operation to be performed on an existing prefix tree.

Type: String

Optionality: Optional.

Allowed:

ADD	Adds profile values to an existing prefix tree. Duplicate entries are ignored.
DEL	Removes the profile values from an existing prefix tree. Missing entries are ignored.

Default:

Notes: Updates the prefix tree specified in the TAG field with the profile values specified in the VALUE field.

Example: `PTMODE=ADD`

Profile tag types

The profile tag names for profile fields are defined on the **Profile Details** tab in the ACS Configuration screen.

This table describes the profile tag data types supported by the PI.

Tag Type	Description
BOOLEAN	A single value: • "T" - True • "F" - False
BYTE	A single byte decimal integer (signed) in the range -128 to 127.
DATE	A date in the format: "YYYYMMDDHHMMSS".
INTEGER	A decimal integer value (signed). Maximum 4 bytes.
LNSTRING	A hexadecimal string. Valid characters are 0-9, A-F.
LOPREFIX	A limited ordered prefix tree. This is made up of a limited list of numbers.
NSTRING	A hexadecimal string. Valid characters are 0-9, A-F.
OPREFIX	An ordered prefix tree comprising a list of numbers.
PREFIX	A prefix tree comprising a list of numbers.
STRING	A non limited ASCII string.
ZONE	A set of shape definitions for the area covered by the zone. Shapes may be circular or rectangular and are defined by their coordinates.

Note: The limits for LNSTRING and LOPREFIX tag values may be set for the Product Type in the Subscriber Management screen, or in Resource Limits in the Service Management screen.

Logic and constraints

The following rules apply when using the CCSCD9=CHG command:

1. You must specify an existing MSISDN.
2. The TAG and VALUE parameters must have the same number of items.
3. If more than one tag is specified, an empty value is indicated by putting no data between the pipe separators.
4. If an empty value is specified for a tag, the tag will be deleted.
5. If TAG is set to PREFIX, OPREFIX or LOPREFIX, PTMODE is used.

Note: Deleting an F&F number or an F&D number does not affect the change count as these changes are always free of charge.

Query Subscriber Profile

Description

Use the command CCSCD9 and action QRY to query entries in the subscriber profile and returns values for the specified tags.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Parameter	Description
TAG	This is a predefined profile tag name corresponding to a PROFILE_TAG_NAME in the ACS_PROFILE_DETAILS table. More than one tag may be specified. Multiple tags must be separated by the pipe ("|") symbol. The specified value is limited to 2000 characters, regardless of the number of tags specified.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

No optional parameters are supported.

Add a CLI to a Closed User Group

Description

Use the command CCSCG1 and action ADD to add one or more CLI identifiers to an existing closed user group. You can also add one or more virtual members to a CUG using this command. Virtual members are defined as single numbers or number ranges, by providing either a start and end number or a prefix and a number length.

You can also create a new closed user group and add one or more CLI identifiers to it.

When using CCSCG1 to add more than one CLI or virtual members to a CUG, a commit occurs only if the entire command completes successfully. If any part of the command fails, no updates are made to the CUG.

Required parameters

This table describes each required parameter.

Parameter	Description
CUG	The closed user group name.

Parameter	Description

MSISDN	The CLI to add to the closed user group. This must be a valid defined CLI.
	Specify multiple MSISDN addresses to add using the following format:
	`<pi:MSISDN>123456789|987654321|123123123</pi:MSISDN>`
	Specify virtual member ranges to add using the following format:
	`<pi:VM_RANGE>44-55|777|888|900-903</pi:VM_RANGE`
	Note: When adding virtual member ranges there is an issue with identifying the CUG and tariff plan if the command is issued without also creating an MSISDN at the same time. This is because the PI relies on the product and customer information associated with the incoming MSISDN to get those details.
	Alternately, ACS_CUSTOMER can be specified, which frees the dependency on MSISDN and allows the CCSCG1_ADD command and CCSCG1_DEL commands to process virtual member ranges alone.
	Therefore, when adding virtual member ranges, the parameter MSISDN has become optional and is now only required if ACS_CUSTOMER is absent.
	However, a new CUG cannot be created, even if the customer is specified, with a default tariff plan name because that information is derived from the product type of MSISDN. So when creating a new CUG and adding virtual member ranges with no MSISDNs, a tariff plan name must be included.
	As per current functionality, the CREATE_CUG option (Y|N) can be used to indicate whether or not the CUG container should be created if it does not already exist.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Parameter	Description
CREATE_CUG	Determines whether or not a new CUG is created. Valid values are: • "Y" - create a new CUG • "N" - (default) do not create a new CUG
TARIFF_PLAN	For new CUGs only. The name of the associated tariff plan. This may be up to 50 characters long.
MSISDN	Only required if ACS_CUSTOMER is absent.
ACS_CUSTOMER	When specified, frees the dependency on MSISDN and allows the CCSCG1_ADD command and CCSCG1_DEL commands to process virtual member ranges alone.

Logic and constraints

The following rules apply when using the CCSCG1=ADD command:

1 The MSISDN must be a valid defined CLI.

2 If the closed user group already exists, the MSISDN must not be included in the group already. The CREATE_CUG and TARIFF_PLAN parameters will be ignored.

3 If the closed user group does not exist already, CREATE_CUG must be set to "Y". The TARIFF_PLAN, if specified, must be a valid defined tariff plan. If not specified, the tariff plan for the Product Type associated with the MSISDN is used. Otherwise an error will be returned.

4 MSISDN becomes optional when including ACS_CUSTOMER. This is useful when adding virtual member ranges to a CUG.

Delete a CLI from a Closed User Group

Description

Use the command CCSCG1 and action DEL to delete a CLI or one or more virtual member ranges from a closed user group.

Required parameters

This table describes each required parameter.

Parameter	Description
CUG	The closed user group name.
MSISDN	The CLI to delete from the closed user group. This must be a valid defined CLI.
VM_RANGE	The virtual member ranges. Required when using this command to delete virtual ranges from a CUG.
ACS_CUSTOMER	Service provider name. Required when deleting virtual member ranges. This must be the name of an existing service provider. Defaults to "Boss" if not specified.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

No optional parameters are supported.

Logic and constraints

The following rules apply when using the CCSCG1=DEL command:

1 The MSISDN must be a valid defined CLI.

2 The CUG must be an existing closed user group and the MSISDN must exist in it.

3 The MSISDN and/or (VM_RANGE and ACS_CUSTOMER) must be specified for the CCSCG1 ADD and DEL commands to succeed

Create a New Closed User Group

Description

Use the command CCSCG2 and action ADD to create a new closed user group.

Required parameters

This table describes each required parameter.

Parameter	Description
NAME	The closed user group name. This can be up to 50 characters long.
TARIFF_PLAN	The name of the associated tariff plan. This can be up to 50 characters long.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Parameter	Description
ACS_CUSTOMER	Service provider name. This must be the name of an existing service provider. Defaults to "Boss" if not specified.

Logic and constraints

The following rules apply when using the CCSCG2=ADD command:

1. The closed user group name must not already exist for this service provider.
2. You must specify an existing tariff plan for this service provider.
3. When creating a CUG and adding virtual member ranges at the same time, the new CUG cannot be created, even if the customer is specified, with a default tariff plan name because that information is derived from the product type of MSISDN. So when creating a new CUG and adding virtual member ranges with no MSISDNs, a tariff plan name must be included.

Modify a Closed User Group

Description

Use the command CCSCG2 and action CHG to change the tariff plan associated with the closed user group to a different tariff plan.

Required parameters

This table describes each required parameter.

Parameter	Description
NAME	The closed user group name. This can be up to 50 characters in length.
TARIFF_PLAN	Name of the new tariff plan to associate with the closed user group. This can be up to 50 characters in length.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Parameter	Description
ACS_CUSTOMER	Service provider name. This must be the name of an existing service provider. Defaults to "Boss" if not specified.

Logic and constraints

The following rules apply when using the CCSCG2=CHG command:

1 You must specify an existing closed user group for this service provider.
2 You must specify an existing tariff plan for this service provider.
3 If specified, the ACS_CUSTOMER must be an existing service provider.

Delete a Closed User Group

Description

Use the command CCSCG2 and action DEL to delete a closed user group for a service provider.

Required parameters

This table describes each required parameter.

Parameter	Description
NAME	The closed user group name. This can be up to 50 characters in length.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Parameter	Description
ACS_CUSTOMER	Service provider name. This must be the name of an existing service provider. Defaults to "Boss" if not specified.

Logic and constraints

The following rule applies when using the CCSCG2=DEL command:

1 You must specify an existing closed user group for this service provider.

Query a Closed User Group

Description

Use the command CCSCG1_QRY to query a closed user group for a service provider.

Required parameters

This table describes each required parameter.

Parameter	Description
CUG	The closed user group name. This can be up to 50 characters in length.
ACS_CUSTOMER	This is a predefined profile tag name corresponding to a PROFILE_TAG_NAME in the ACS_PROFILE_DETAILS table. More than one tag may be specified. Multiple tags must be separated by the pipe ("|") symbol. The specified value is limited to 2000 characters, regardless of the number of tags specified.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Parameter	Description
ACS_CUSTOMER	Service provider name. This must be the name of an existing service provider. Defaults to "Boss" if not specified.

Logic and constraints

The following rule applies when using the CCSCG1_QRY command:

1 You must specify an existing closed user group for this service provider.

Perform a BPL

Description

Use the command CCSBPL and action EXE to execute a defined business process logic for a specified subscriber.

Note: The BPL will only be executed if the security level of the logged in PI user is greater or equal to the PI security level assigned to the BPL itself.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Parameter	Description
BPL	The short name of the business process logic to execute. This can be up to 50 characters in length.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
EXT1	The IDP extension. This can be up to 160 characters long.
EXT2	The IDP extension. This can be up to 160 characters long.
EXT3	The IDP extension. This can be up to 160 characters long.
EXT4	The IDP extension. This can be up to 160 characters long.
EXT5	The IDP extension. This can be up to 160 characters long.

Note: The IDP extension parameters are defined in the BPL record and correspond to the extension parameters (4 to 8) defined in the **acs.conf** configuration file.

Optional parameters

This table describes each optional parameter.

Parameter	Description
EXT6	IDP extension parameter. This can be up to 160 characters long.
PROVIDER	Name of the service provider.
WALLET_TYPE	The applicable wallet type. Values are: • "Primary" • "Secondary"

Note: The IDP extension parameter is defined in the BPL record and corresponds to the extension parameter 9 defined in the **acs.conf** configuration file.

Logic and constraints

The following rules apply when using the CCSBPL=EXE command:

1. You must specify an existing MSISDN.
2. You must specify an existing BPL name.
3. If any of the parameters in the range EXT2 to EXT6 are specified, all the preceding EXT parameters must also be specified. That is, If EXT5 is specified, EXT4, EXT3, EXT2, and EXT1 must also be specified.
4. The PI does not check that the MSISDN has the specified WALLET_TYPE. This is verified by the BPL subsystem.

Change Subscriber Searchable Fields

Description

Use the command CCSSF1 and action CHG to change the detail for the fields which can be used to search for a subscriber on the Subscriber Profile Management screens.

For more information about the Subscriber Profile Management screens, see *Subscriber Profile Manager User's Guide*.

Required parameters

CCSSF1 requires the following parameters.

`MSISDN`

Syntax: `MSISDN=`*`CLI`*

Description: The CLI for the subscriber account this command applies to.

Type: Integer

Optionality: Mandatory

Allowed: The CLI must have already been set up as a valid subscriber in the CCS_ACCT_REFERENCE table.

Default: None

Notes:

Example: `MSISDN=0123456789`

`TAG`

Syntax: `TAG=`*`name`*`|...`

Description: A predefined profile tag name corresponding to a DISPLAY_NAME in the CCS_SEARCH_KEYS table.

Type: String

Optionality: Mandatory

Allowed: The TAG should be a valid key display name as defined in the CCS_SEARCH_KEYS table.

Default: None

Notes: More than one tag may be specified. They should be separated by the pipe (`|`) symbol.

Example: `TAG=adContactNumber|adPostcode`

`VALUE`

Syntax: `VALUE=`*`value`*

Description: The value for the corresponding tag.

Type: Plain text (however, must match the tag data type of the destination field).

Optionality: Mandatory

Allowed: The VALUE is plain text, but the characters listed below need to be escaped:

- To indicate a comma (`,`), use backslash-c (`\c`)
- To indicate a semi-colon (`;`), use backslash-s (`\s`)
- To indicate an equals sign (`=`), use backslash-e (`\e`)
- To indicate a pipe (`|`), use backslash-pipe (`\|`)
- To indicate a new-line, use backslash-n (`\n`)
- To indicate a backslash (`\`), use double backslash (`\\`)
  A single backslash will be interpreted as a syntax error.

The VALUE is split into fields at every new-line indicator (\n)

Default: None

Notes: If more than one tag is specified, then tag values must be pipe separated and listed in the same order as for TAG.

Example: `VALUE=+440123456789|AB123`

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

No optional parameters are supported.

Logic and constraints

The following rules apply when using the CCSSF1=CHG command:

1. If TAG is already defined, its value will be changed to the new value.
2. If VALUE is empty and the TAG is already defined, the entry will be deleted.
3. The number of values must match the number of tags:

   Example: If "Telephone Number" is "+440123456789" and "Postcode" is "AB123".

   `TAG=adContactNumber|adPostcode,VALUE=+440123456789|AB123`

Query Subscriber Searchable Fields

Description

Use the command CCSSF1 and action QRY to query the search fields which can be used to search for a subscriber on the Subscriber Profile Management screens.

For more information about the Subscriber Profile Management screens, see *Subscriber Profile Manager User's Guide*.

Required parameters

CCSSF1 requires the following parameters.

`MSISDN`

Syntax: `MSISDN=`*`CLI`*

Description: The CLI for the subscriber account this command applies to.

Type: Integer

Optionality: Mandatory

Allowed: The CLI must have already been set up as a valid subscriber in the CCS_ACCT_REFERENCE table.

Default: None

Notes:

Example: `MSISDN=0123456789`

`TAG`

Syntax: `TAG=`*`name`*`|...`

Description: A predefined profile tag name corresponding to a DISPLAY_NAME in the CCS_SEARCH_KEYS table.

Type: String

Optionality: Mandatory

Allowed: The TAG should be a valid key display name as defined in the CCS_SEARCH_KEYS table.

Default: None

Notes: More than one tag may be specified. They should be separated by the pipe (|) symbol.

Example: `TAG=adContactNumber|adPostcode`

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

No optional parameters are supported.

Logic and constraints

The following rules apply when using the CCSSF1=QRY command:

1 The MSISDN should be a valid MSISDN defined in the CCS_ACCT_REFERENCE table.

2 The TAG should be a valid key display name defined in the CCS_SEARCH_KEYS table.

3 To specify more than one tag to query, the tag parameter should contain the names of the tags separated by pipe symbols.

Chapter 3

PI Wallet Package

Overview

Introduction

This chapter describes the Oracle Communications Convergent Charging Controller provisioning interface (PI) commands for provisioning wallets on a VWS domain.

These commands are installed by the `piWalletSms` package. This extends the `piSubscriberSms` package by providing enhancements to existing commands and adding new commands.

In this chapter

This chapter contains the following topics.

Add/Link a Wallet

Description

Use the enhanced CCSCD1 command and action ADD to add a:

- New subscriber or range of subscribers
- New wallet or range of wallets
- Link to an existing wallet

To successfully execute this command, the subscriber account(s) must be using a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
PRODUCT	Product type name

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
MSISDN	The CLI for the new subscriber. This must not exist already in the database.
START_MSISDN	The starting CLI for a range of CLIs. This must be lower than or equal to END_MSISDN.
END_MSISDN	The final CLI in the range. This must be greater than or equal to START_MSISDN. **Note:** START_MSISDN and END_MSISDN must be the same length.
ACCOUNT_NUMBER	The account reference for this subscriber. This is a 20 digit number that is created automatically when a new subscriber is added.
PROVIDER	The service provider name. The service provider must exist already.

Optional parameters

This table describes each optional parameter.

Parameter	Description
BYPASS_NUMBER	A reference number for linking subscriber accounts.
CHARGING_DOMAIN	The reference ID of the charging domain for this subscriber.
CREDIT_LIMIT	Sets the credit limit (in minor currency unit) for Limited Credit subscriber accounts.
CURRENCY	Sets the currency code. This must be an existing currency code.
EXTRA_EDR	List of EDR fields to add using the following format: "*KEY1=VALUE1|KEY2=VALUE2*" This can be up to 100 characters long.
INITIAL_STATE	Sets the initial state for new wallets. This parameter will be ignored for linked wallets.
LANGUAGE	The language short name. The language must exist already in the language table. Defaults to the language of the logged PI user if this is defined. Otherwise the system default language is used.
LIMIT_TYPE	Sets the limit type for the subscriber. Valid values are: • "Debit", • "Single", • "Use", • "Credit" • "Limited Credit"

Parameter	Description
LW_ACCOUNT_NUMBER	The subscriber account reference number for a linked wallet.
LW_MSISDN	The CLI for the linked wallet.
MAX_CONCURRENT_ACCESS	Sets the maximum concurrent accesses for the wallet.
PIN	The subscriber's PIN. The PIN is only used when creating new accounts. It is a 4 digit decimal number. If not specified, the last 4 digits of the generated account number are used.
TRACKING_DOMAIN	The reference id of the tracking domain for this subscriber. Must be a VWS domain ID.
WALLET_TYPE	The wallet type name. Values are: • "Primary" • "Secondary"

Logic and constraints

Here are the general rules that apply when using the CCSCD1=ADD command to create subscribers and wallets.

1. You must either specify the MSISDN to create a single new subscriber and wallet, or the START_MSISDN and END_MSISDN to create a range of new subscribers and wallets. The maximum size for a range of subscriber MSISDNs is 1000.
2. You must specify the PRODUCT and the PROVIDER to create new subscribers (single or range). This must be a valid combination. That is the PROVIDER must be the ACS_CUST_ID for the account type of the specified PRODUCT. They should be the names of the product or provider, rather than the internal ID.
3. You can optionally specify the ACCOUNT_NUMBER when creating a single new subscriber and wallet. You must not specify the ACCOUNT_NUMBER when creating a range of new subscribers and wallets.
4. When creating a new account to link to an existing wallet, MSISDN and PRODUCT must be specified. PROVIDER will be ignored as the service provider of the existing wallet will be used. You can optionally specify the ACCOUNT_NUMBER, but the START_MSISDN and END_MSISDN must not be specified.
5. When creating a new range of accounts to link to an existing wallet, START_MSISDN, END_MSISDN and PRODUCT must be specified. PROVIDER will be ignored as the service provider of the existing wallet will be used. You must not specify the ACCOUNT_NUMBER or MSISDN.
6. When creating a range of subscribers and /or wallets:
   - If WALLET_TYPE is not specified or “Primary”, then START_MSISDN, END_MSISDN and any MSISDN within the range must not already exist in the database.
   - If WALLET_TYPE is “Secondary”, then START_MSISDN, END_MSISDN and any MSISDN within the range must already exist in the database and not already have a “Secondary” wallet.
   - Failure to create any subscriber and/or wallet within the range will result in no subscribers and/or wallets within the range being created.
   - Each subscriber will have the same PROVIDER, PRODUCT, CHARGING_DOMAIN, TRACKING_DOMAIN, LANGUAGE, CURRENCY, MAX_CONCURRENT_ACCESS, LIMIT_TYPE, CREDIT_LIMIT, BYPASS_NUMBER and PIN, if specified.
7. When adding a new wallet to an existing single account, ACCOUNT_NUMBER and PRODUCT must be specified. MSISDN must not be specified.
8. If WALLET_TYPE is not specified, then the default wallet type (“Primary”) is used.

9 If WALLET_TYPE is not “Primary”, then the account must already have a “Primary” wallet.

10 A subscriber may have only one wallet of each type.

11 If MSISDN is not specified, ACCOUNT_NUMBER references an existing account to associate the wallet to. The new wallet will be linked to the same account reference as the specified ACCOUNT_NUMBER.

12 If MSISDN is specified and ACCOUNT_NUMBER is not specified, the ACCOUNT_NUMBER is created automatically when creating the new account.

13 If both MSISDN and ACCOUNT_NUMBER are specified, the ACCOUNT_NUMBER is created as the service provider prefix plus the specified ACCOUNT_NUMBER. For example, if the service provider prefix is '10' and the specified ACCOUNT_NUMBER is '12345678', then the subscriber is created with account number '1012345678'.

14 The PIN is ignored when adding a secondary wallet.

Logic and constraints for creating a new wallet

Here are the rules that apply when using the CCSCD1=ADD command to create a new wallet.

1 Both LW_MSISDN and LW_ACCOUNT_NUMBER should not be specified when creating a new wallet.

2 Currency is ignored when adding a new card to an existing subscriber account.

3 All balance types defined for this account type will be created in the wallet, except for those with the Exclude From Creation field set to 'Y'.

4 If CHARGING_DOMAIN is specified then the subscriber account is created on that specific BE pair. If it is not specified then the standard internal Prepaid Charging functionality will be used.

5 The CHARGING_DOMAIN must be a reference to a valid VWS domain.

6 If no TRACKING_DOMAIN is specified, the subscriber(s) will not be associated with a tracking domain.

7 MAX_CONCURRENT_ACCESS must be positive and if not specified will default to 1.

8 LIMIT_TYPE must be one of "Debit", "Single Use", "Credit" or "Limited Credit", or not specified. If not specified the default type "Debit" is used.

9 CREDIT_LIMIT is ignored if the LIMIT_TYPE is not “Limited Credit”.

10 CREDIT_LIMIT is specified in small currency units if the default balance type does not have a Time unit type.

11 CREDIT_LIMIT is specified in seconds if the default balance type has a Time unit type.

12 If INITIAL_STATE is not specified, then the wallet state will be determined either by the `pi.CCSCD1.ADD.initialState` configuration parameter, or it will be set to the default value ‘P’ (Pre-use).

13 If specified, the INITIAL_STATE and the `pi.CCSCD1.ADD.initialState` configuration parameter must be valid wallet states. These are:

- A = Active
- D = Dormant
- F = Frozen
- P = Pre-Use
- S = Suspended
- T = Terminated

Creating a link to an existing wallet

Here are the rules that apply when using the CCSCD1=ADD command to create a link to an existing wallet.

1 To link to an existing wallet, either LW_MSISDN or LW_ACCOUNT_NUMBER must be specified.

2 If both LW_MSISDN and LW_ACCOUNT_NUMBER are specified, LW_ACCOUNT_NUMBER is used and LW_MSISDN is ignored.

3 If LW_MSISDN is specified, a subscriber account with that CLI must exist in the database already.

4 If LW_ACCOUNT_NUMBER is specified, a subscriber account with that account number must exist in the database already.

5 The specified PRODUCT must be a product type of the existing wallet's service provider.

6 The wallet type to link to is determined by the WALLET_TYPE parameter.

7 When linking to an existing wallet, the following parameters are ignored and therefore the values of the existing wallet are not affected:

- PROVIDER
- PROVIDER_ID
- CURRENCY
- CHARGING_DOMAIN
- TRACKING_DOMAIN
- MAX_CONCURRENT_ACCESS
- LIMIT_TYPE
- CREDIT_LIMIT
- INITIAL_STATE

Change Wallet Details

Description

Use the enhanced CCSCD1 command and action CHG to change the details for:

- An existing subscriber
- The subscriber's wallets and balances

To successfully execute this command the subscriber must be using a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
BALANCE	The value of the new balance in small currency
BALANCE_EXPIRY	The number of days from today's date before the balance will expire.
BALANCE_EXPIRY_DATE	Sets the balance expiry date using this format: YYYYMMDDHHMMSS.
CREDIT_LIMIT	Sets the credit limit (in small currency) for Limited Credit subscriber accounts.
CURRENCY	Sets the currency code. This must be an existing currency code.
LANGUAGE	The language short name. The language must exist already in the language table. Defaults to the language of the logged PI user if this is defined. Otherwise the system default language is used.

Parameter	Description
NEW_MSISDN	The new CLI for the subscriber. This must not exist already.
PIN	Sets the subscriber's PIN (a 4 digit decimal number).
PRODUCT	The product type name. Must be valid for the service provider.
STATUS	Sets the status of the account. Valid values are: • A = Active • D = Dormant • F = Frozen • P = Pre-Use • S = Suspended • T = Terminated
WALLET_EXPIRY	The number of days from today's date before the wallet will expire.
WALLET_EXPIRY_DATE	Sets the wallet expiry date using this format: YYYYMMDDHHMMSS. If set to blank then the wallet will not expire. **Note:** Only current or future dates are valid.

Optional parameters

This table describes each optional parameter.

Parameter	Description
BALMODE	Determines how the BALANCE value is treated. Valid values are: • ABSOLUTE • DELTA (default)
BALANCE_TYPE	Sets the balance type for the BALANCE, BALANCE_EXPIRY and WALLET_EXPIRY values.
EXPMODE	Sets whether or not the expiry date is always used.
EXTRA_EDR	List of EDR fields to add using the following format: "*KEY1=VALUE1|KEY2=VALUE2*" This can be up to 100 characters long.
WALLET_TYPE	The name of the wallet type for the BALANCE, BALANCE_EXPIRY and WALLET_EXPIRY values. Values are: • Primary • Secondary

Logic and constraints

Here are the rules that apply when using the CCSCD1=CHG command to modify subscribers and wallets.

1. You must specify an existing MSISDN and at least one constraint required parameter.
2. The NEW_MSISDN, if specified, should not exist already.
3. If WALLET_EXPIRY_DATE is specified then the wallet expiry date will be set to that date, even if the wallet is currently set to expire at a later date or it is set not to expire.
4. The BALANCE value must be specified in small currency units unless the unit type for the balance type is Time. For Time balances, the balance value must be specified in seconds.

5 Negative BALANCE amounts will be credited to the account. Positive BALANCE amounts will be debited from the account.

6 If the WALLET_EXPIRY_DATE and WALLET_EXPIRY parameters are both specified, then the wallet expiry is set to the value specified for the WALLET_EXPIRY_DATE.

7 If WALLET_TYPE is not specified then the default wallet type is used.

8 If BALANCE_TYPE is not specified then the default balance type is used.

9 If BALANCE_TYPE is specified and the VWS has been configured as a tracking domain only, then the balance category for the specified balance type must be either Fraud or Expenditure.

10 If WALLET_TYPE or BALANCE_TYPE is specified, then at least one of BALANCE, BALANCE_EXPIRY or WALLET_EXPIRY must also be specified.

11 If BALMODE is specified then BALANCE must also be specified.

12 If BALANCE is specified but BALMODE is not, then the default of “DELTA” will be used for BALMODE.

13 If BALMODE is set to "ABSOLUTE", then the BALANCE amount will be applied to the bucket with the longest expiry date and all other buckets of that BALANCE_TYPE will be expired. This means that the bucket values will be set to zero and expiry will be set to “Now” for the defined BALANCE_TYPE.

14 If BALMODE is set to "DELTA", the bucket is chosen based on the following:

- The bucket which has the latest expiry date will be updated.
- If there is a bucket which never expires, the bucket which never expires will be updated.
- If there are buckets with the same latest expiry (or none at all), the bucket with the highest ID will be updated.

15 CREDIT_LIMIT must be specified in small currency units unless the unit type for the balance type is Time. For Time balances, the credit limit must be specified in seconds. This parameter will be ignored if the account is not a limited credit type.

16 If EXPMODE is specified then the BALANCE_EXPIRY_DATE and the WALLET_EXPIRY_DATE must be specified. The value specified for EXPMODE must be "ABSOLUTE".

17 If STATUS=A and EXPMODE=ABSOLUTE are specified then the dates specified in WALLET_EXPIRY_DATE and BALANCE_EXPIRY_DATE are used for the wallet expiry date and the balance expiry date (of the inserted bucket). This overrides the use of expiry dates set in the edit product type screen.

18 EXTRA_EDR will only be added to an EDR if this command causes a billing engine update. That is, if any of these parameters are also specified:

- BALANCE, where the balance change is not 0
- BALANCE_EXPIRY
- BALANCE_EXPIRY_DATE
- STATUS
- WALLET_EXPIRY
- WALLET_EXPIRY_DATE
- CREDIT_LIMIT, for a limited credit type account.

Delete a Wallet/Balance

Description

Use the enhanced CCSCD1 command and action DEL to delete a:

- Subscriber account, including any account history and recharge transactions
- Subscriber account and any linked accounts
- Single wallet
- Balance type from a wallet

To successfully execute this command the subscriber must be using a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Parameter	Description
BALANCE_TYPE	The name of the balance type to delete.
DELETE_LINKED_MSISDNS	Sets whether or not to delete any other subscriber accounts that share this wallet. Values are: • "Y" - Yes • "N" - No (default value)
EXTRA_EDR	List of EDR fields to add using the following format: "*KEY1=VALUE1*|*KEY2=VALUE2*" This can be up to 100 characters long.
WALLET_TYPE	The name of the wallet type of the balance to delete.

Logic and constraints

Here are the rules that apply when using the CCSCD1=DEL command to delete subscribers, wallets and balances.

1 You must specify an existing MSISDN.

2 If DELETE_LINKED_MSISDNS=N or is not specified, then:

- If WALLET_TYPE and BALANCE_TYPE are not specified, the whole account is deleted.
- If WALLET_TYPE is specified and BALANCE_TYPE is not specified, the wallet with that WALLET_TYPE is deleted from the account. The WALLET_TYPE must exist for the subscriber.
- If a wallet is linked to another account, the wallet is not deleted. Only the link to this account will be removed.
- A Primary wallet cannot be deleted if the account has a Secondary wallet.
- If WALLET_TYPE and BALANCE_TYPE are both specified, the balance with that BALANCE_TYPE is deleted from the wallet with that WALLET_TYPE. The account must have a wallet with that WALLET_TYPE and the wallet must have a balance with that BALANCE_TYPE.
- If WALLET_TYPE is not specified and BALANCE_TYPE is specified, the balance with that BALANCE_TYPE is deleted from the default wallet. The account must have a wallet with the default wallet type and the wallet must have a balance with that BALANCE_TYPE.

3 If DELETE_LINKED_MSISDNS=Y:

- – All wallets are deleted for the MSISDN, and the MSISDN is deleted.
- – WALLET_TYPE and BALANCE_TYPE are ignored.

- – If other MSISDNs share the wallets, those MSISDNs will be unlinked from the wallets.
- – Any MSISDN with no wallets left after unlinking will be deleted.
- – If an MSISDN is left with only a Secondary wallet, the wallet type is changed to Primary for that MSISDN.

Query Wallet Details

Description

Use the enhanced CCSCD1 command and action QRY to request details about subscribers and wallets.

To successfully execute this command the subscriber must be using a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Parameter	Description
BALANCE_TYPE	The name of the balance type to query for balance, balance expiry date and wallet expiry date details. Special balance type values of ALL return the values for all balance types in a parameter called BALANCES. Special balance type values of ALL_UNRESERVED returns the unreserved balance on each balance of the subscriber. The output contains the value in system currency then subscriber currency.
LIST_TYPE	Pipe ("|") separated list of items to return. Valid values are: • BALANCE • BALANCE_TYPE • BYPASS_NUMBER • CHARGING_DOMAIN • CREATION_DATE • CUG • CURRENCY • FDN • FFD • FFN • FIRST_ACTIVATION_DATE • FREE_SWAPS_REMAINING

Parameter	Description
	• INITIAL_BALANCE • LANGUAGE • LAST RECHARGE_AMOUNT • LAST_CC_RECHARGE_DATE • LAST_ESP_CREDIT • LAST_RECHARGE_DATE • LAST_STATE_CHANGE_DATE • LAST_STATE_CHANGE_REASON • LAST_SWAP_RESET_DATE • LAST_USE_DATE • PREV_BALANCE • PREV_BALANCE_EXPIRY_DATE • PREV_WALLET_EXPIRY_DATE • PRODUCT • SERVICE_PROVIDER • STATUS • TOTAL_EXP_DATE • TRACKING_DOMAIN • UNRESERVED_BALANCE • WALLET_EXPIRY_DATE • WALLET_LIFE_CYCLE_PERIOD • WALLET_REFERENCE • WALLET_TYPE
WALLET_TYPE	The name of the wallet type of the balance to query for balance, balance expiry date and wallet expiry date details.

Logic and constraints

Here are the rules that apply when using the CCSCD1=QRY command to query the database for details on subscribers, wallets and balances.

1. You must specify an existing MSISDN.
2. If a value is null then the returned field will be empty.
3. If WALLET_TYPE is not specified, then the default wallet type will be used.
4. If BALANCE_TYPE is not specified, then the default balance type will be used.
5. If BALANCE_TYPE=ALL is specified, but WALLET_TYPE is not specified, then the default wallet will be used.
6. If BALANCE_TYPE=ALL is specified then:
   - The values for all balance types will be returned in a parameter called BALANCES
   - The name of the balance type will be returned
   - LIST_TYPE will be ignored and the return text will automatically be set to: MSISDN, WALLET_EXPIRY_DATE, BALANCES (value,balance expiry)
7. If BALANCE_TYPE=ALL_UNRESERVED is specified, then:
   - The output is the unreserved balance on each balance of the subscriber
   - The output contains the value in system currency then subscriber currency
8. If the VWS has been configured as a tracking domain only and:
   - BALANCE_TYPE=ALL is specified then only balance types where the balance category is either Fraud or Expenditure will be returned

- BALANCE_TYPE is specified then the balance category for the specified balance type must be either Fraud or Expenditure

9 For balances other than Time balances, the amount will be in small currency units. For Time balances, the amount will be in seconds.

10 The balance currency defaults to the wallet user currency. This can be changed to the system currency via the configuration parameter: `pi.CCSCD1.QRY.currencyType`. For more information, see *PI User's and Technical Guide*.

11 CURRENCY values will always be returned in the wallet currency, even if the balance currency has been set to the system currency.

12 Bucket values for balances other than Time balances will be in small currency units. For Time balances, bucket values will be in seconds.

13 If LIST_TYPE is specified it should contain at least one valid item to return. If it is not specified, all values are returned.

14 All dates are returned in the 24 hour format YYYYMMDDHHMMSS if set or blank if not set.

15 Balance information for periodic charge type balances will not be returned due to their specialist nature. For periodic charge information, see *Query Subscribed Periodic Charges* (on page 60) command.

Query Wallet Reservation

Description

Use the CCSRM1 command and action QRY to request details about wallet reservations.

To successfully execute this command the subscriber must be using a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the target wallet this command applies to. This must exist already.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Field	Description	Default
WALLET_TYPE	Primary or Secondary.	Primary

Logic and constraints

Here are the rules that apply when using the CCSRM1=QRY command on a VWS domain.

1 You must specify an existing MSISDN to base the query on.

2 You can specify the optional parameter if required. If not specified, the listed default value for the parameter will be used.

Delete Wallet Reservation

Description

Use the CCSRM1 command and action DEL to delete wallet reservations.

To successfully execute this command the subscriber must be using a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the target wallet this command applies to. This must exist already.
SCP_ID	First half of reservation key. Can be obtained from CCSRM1:QRY.
CALL_ID	Second half of reservation key. Can be obtained from CCSRM1:QRY.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Field	Description	Default
OPERATION	How to end the reservation. 0 = Apply tariff default operation (either commit or revoke) 1 = Commit 2 = Revoke	Let server decide between commit and revoke based on applicable **eserv.config**.
WALLET_TYPE	Primary or Secondary	Primary

Logic and constraints

Here are the rules that apply when using the CCSRM1=DEL command on a VWS domain.

1. You must specify an existing MSISDN to base the query on.
2. You must specify the SCP_ID from the reservation key.
3. You must specify the CALL_ID from the reservation key.
4. Optionally, you can specify either, none or both of the optional parameters.
5. Any optional parameter not specified uses the listed default value.

Change Barred Numbers with EDR

Description

Use the enhanced CCSCD4 command and action CHG to change a subscriber's allowed or barred numbers and generate a type 30 EDR showing the old and new values.

To successfully execute this command the subscriber must be on a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
LIST_TYPE	Sets whether this is a list of allowed or barred numbers. Valid values are: • "A" - Allowed • "B" - Barred
IGNORE	Sets whether or not the call barring list should be ignored. Valid values are: • "Y" - True • "N" - False
ADD	The list of MSISDNs to be added to the existing call barring list. This will be a pipe ("|") separated list of up to 50 MSISDNs.
DEL	The list of MSISDNs to be deleted from the existing call barring list. This will be a pipe ("|") separated list of up to 50 MSISDNs.
CLEAR	Either "Y" or "N". Defaults to "N". When set to "Y" the existing call barring list is cleared and then any MSISDNs in the ADD list are added. The DEL parameter is ignored.

Optional parameters

This table describes each optional parameter.

Parameter	Description
EXTRA_EDR	List of EDR fields to add using the following format: "*KEY1=VALUE1|KEY2=VALUE2*" This can be up to 100 characters long.

Logic and constraints

The following rules apply when using the CCSCD4=CHG command:

1. You must specify an MSISDN for an existing subscriber on a VWS domain.
2. You must specify at least one constraint required parameter.
3. Prior to this command, the call barring list must not include any MSISDNs from the ADD list.
4. Prior to this command, the call barring list must include all the MSISDNs in the DEL list.

Note: On successfully completing this command a type 30 BE EDR will be created to show the old and new values.

Add an F&F/F&D Number with EDR

Description

Use the enhanced CCSCD8 command and action ADD to add a new Friends and Family or Friends and Destination number to a subscriber.

To successfully execute this command the subscriber must be on a VWS domain. On completion a named event and a BE EDR will be generated.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
FFNUM	The F&F number to be added. This can be up to 20 digits long. **Note:** The number specified must not be on the F&F numbers barred list.
FDNUM	The F&D number to be added. This can be up to 20 digits long. **Note:** The number specified must not be on the F&D numbers barred list.

Optional parameters

This table describes each optional parameter.

Parameter	Description
EXTRA_EDR	List of EDR fields to add using the following format: "*KEY1=VALUE1|KEY2=VALUE2*" This can be up to 100 characters long.

Logic and constraints

Here are the rules that apply when using the CCSCD8=ADD command on a VWS domain.

1. You must specify an MSISDN for an existing subscriber on a VWS domain.
2. You must specify either the FFNUM or the FDNUM parameter (not both).
3. If the F&F numbers white list is not set to be ignored and you are adding an F&F number, then the value specified for FFNUM must match a prefix in the F&F white list.
4. When adding an F&F number, the maximum F&F numbers defined for the subscriber's product type must not be exceeded.
5. When adding an F&D number, the value specified for FDNUM must exist already in the list of prefixes defined in the Prepaid Charging F&D prefix map.
6. Once specified, an F&D number cannot be re-specified.

7 Each time an F&F number or an F&D number is added, the change count is automatically increased by one and the change date/time is set to "Now".

8 The change count for the F&F number is checked against the free count for the product type.
- If the new total change count is less than or equal to the free count, the Named Event Class=FnF FnD Events, Charge=FnF Config Change, with 100% discount will be sent to the BE to produce a BE CDR.
- If the new total change count is greater than the free count, the Named Event Class=FnF FnD Events, Charge=FnF Config Change, with 0% discount will be sent to the BE to produce a BE CDR, and the F&F event will be charged.

9 The change count for the F&D number is checked against the free count for the product type.
- If the new total change count is less than or equal to the free count, the Named Event Class=FnF FnD Events, Charge=FnD Config Change, with 100% discount will be sent to the BE to produce a BE CDR.
- If the new total change count is greater than the free count, the Named Event Class=FnF FnD Events, Charge=FnD Config Change, with 0% discount will be sent to the BE to produce a BE CDR, and the F&D event will be charged.

Change F&F/F&D Numbers with EDR

Description

Use the command CCSCD8 and action CHG to change the Friends & Family or Friends & Destination details for a subscriber.

To successfully execute this command the subscriber must be on a VWS domain. On completion a named event will be generated. A type 28 BE EDR will also be generated showing the old and new values for the F&F Numbers and the F&D number.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
FFNUM	The F&F number to change.
NEW_FFNUM	The new F&F number. This can be up to 20 digits long. **Note:** The number specified must not be on the F&F numbers barred list.
TYPE	Sets the type of service. Valid values are: • "F" – Friends & Family • "D" – Friends & Destination • "N" – None
FDNUM	The F&D number to set the Active Discount Prefix. This can be up to 20 digits long.

Parameter	Description
	Note: The number specified must not be on the F&D numbers barred list.
FFLIST	Pipe ("|") separated list of F&F numbers. Use to set the list exactly.

Optional parameters

This table describes each optional parameter.

Parameter	Description
EXTRA_EDR	List of EDR fields to add using the following format: "*KEY1=VALUE1|KEY2=VALUE2*" This can be up to 100 characters long.

Logic and constraints

Here are the rules that apply when using the CCSCD8=CHG command on a VWS domain.

1 You must specify an MSISDN for an existing subscriber on a VWS domain.

2 You must specify at least one constraint required parameter.

3 To change a single F&F number specify the FFNUM, NEW_FFNUM and MSISDN parameters. Do not specify the FFLIST parameter.

4 To set the entire F&F number list specify the FFLIST and MSISDN parameters. Do not specify the FFNUM and NEW_FFNUM parameters.

5 When setting F&F numbers, unless the F&F numbers white list is set to be ignored, the new F&F number(s) must match a prefix in the F&F white list.

6 To change the service type specify the MSISDN and Type parameters.

7 To set the F&D number specify the MSISDN and FDNUM parameters.

8 When setting the F&D number, the value specified for FDNUM must exist already in the list of prefixes defined in the Prepaid Charging F&D prefix map. The change count is automatically increased by one and the change date/time is set to "Now".

9 Once specified, an F&D number cannot be re-specified.

10 Each time a single F&F number is changed, the change count is automatically increased by one and the change date/time is set to "Now".

11 The change count for F&F will not be increased for the following changes which will be free:

- Numbers added using FFLIST, and
- TYPE changes.

12 When setting a single F&F number using NEW_FFNUM, the change count for the F&F number will be checked against the free count for the product type.

- If the new total change count is less than or equal to the free count, the Named Event Class=FnF FnD Events, Charge=FnF Config Change, with 100% discount will be sent to the BE to produce a BE EDR.
- If the new total change count is greater than the free count, the Named Event Class=FnF FnD Events, Charge=FnF Config Change, with 0% discount will be sent to the BE to produce a BE EDR, and the F&F event will be charged.

13 The change count for the F&D number will be checked against the free count for the product type.

- If the new total change count is less than or equal to the free count, the Named Event Class=FnF FnD Events, Charge=FnD Config Change, with 100% discount will be sent to the BE to produce a BE EDR.
- If the new total change count is greater than the free count, the Named Event Class=FnF FnD Events, Charge=FnD Config Change, with 0% discount will be sent to the BE to produce a BE EDR, and the F&D event will be charged.

Note: A BE EDR of Type 28 will be produced to show old and new values for F&F Numbers and the F&D number.

Merge Subscriber Wallets

Description

Use the command CCSCD1 and action MSW to merge the wallets of two subscribers.

To successfully execute this command the subscribers must be on a VWS domain. An EDR will be generated on completion.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN1	The CLI for the subscriber account whose wallet will be shared.
MSISDN2	The CLI for the subscriber account which will be updated to share the wallet of MSISDN1.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Parameter	Description
WALLET_TYPE1	The name of the wallet type for the MSISDN1 parameter. Values are: • Primary • Secondary
WALLET_TYPE2	The name of the wallet type for the MSISDN2 parameter. Values are: • Primary • Secondary
EXTRA_EDR	List of EDR fields to add using the following format: "*KEY1=VALUE1|KEY2=VALUE2*" This can be up to 100 characters long.

Logic and constraints

Here are the rules that apply when using the CCSCD1=MSW command on a VWS domain.

1. Both MSISDN1 and MSISDN2 must be existing subscribers on a VWS domain.
2. If you do not specify WALLET_TYPE1 or WALLET_TYPE2 then the default wallet type will be used.
3. If you specify WALLET_TYPE1 but the subscriber for MSISDN1 does not have this wallet type, then MSISDN1 will gain a wallet of this type after the merge.
4. If you specify WALLET_TYPE2 but the subscriber for MSISDN2 does not have this wallet type, then MSISDN2 will gain a wallet of this type after the merge.

5 The merge will be performed via a billing engine message using the following logic.

- The MSISDN2 account will be updated to share the wallet of the MSISDN1 subscriber account.
- Any balances of MSISDN2 will be merged into MSISDN1's wallet.
- The wallet of MSISDN2 will be removed.
- Any other accounts sharing MSISDN2's wallet will be updated to share MSISDN1's wallet.

Query Wallet Users

Description

Use the command CCSCD2 and action QRY to look for the MISISDNs of subscribers that share the same wallet as the specified ACCOUNT_NUMBER or MSISDN. A list of the MSISDNs that share the wallet will be returned. If no MSISDNs are returned, this means the wallet is not shared.

To successfully execute this command the subscribers must be on a VWS domain.

Required parameters

There are no required parameters for this command.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
MSISDN	The CLI to base the query on.
ACCOUNT_NUMBER	The account reference number to base the query on.

Optional parameters

This table describes each optional parameter.

Parameter	Description
WALLET_TYPE	The name of the wallet type to query.

Logic and constraints

Here are the rules that apply when using the CCSCD2=QRY command on a VWS domain.

1 You must specify either the MSISDN or the ACCOUNT_NUMBER to base the query on.

2 If you specify both MSISDN and ACCOUNT_NUMBER, then the query is based on the ACCOUNT_NUMBER.

3 If you do not specify WALLET_TYPE then the default wallet type is used.

Recharge an Account/Bucket

Description

Use the command CCSCD3 and action RCH to perform either a custom or a credit card recharge on a subscriber's account.

Performing a custom recharge will result in one of the following:

- A type 8 EDR being generated for positive amounts

- A type 2 EDR for negative amounts

Performing a credit card recharge will verify existing pending recharges in the recharge queue, but will not actually perform the recharge. Wallets and buckets will be adjusted accordingly and a type 9 EDR will be generated.

To successfully execute this command the subscribers must be on a VWS domain.

Note: If the piVoucherSms package has been installed then you can also use this command for voucher recharges.

Required parameters

This table describes each required parameter.

Parameter	Description
RECHARGE_TYPE	The type of recharge to perform. Values are: • Credit • Custom
REFERENCE	Holds either the voucher number or the recharge requests ID. For credit type recharges this will be the reference id of the pending recharges. For custom type recharges this will be a string matching a reload bonus entry.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber this command applies to.
ACCOUNT_NUMBER	The account reference number.
AMOUNT	The recharge amount. Use small currency units for all balances except Time balances. Specify the amount in seconds for Time balances.

Optional parameters

CCSCD3 accepts the following optional parameters.

`BALANCE_EXPIRY`

Syntax:	`BALANCE_EXPIRY=`*`days`*
Description:	The number of days from today's date before the balance expires.
Type:	Integer
Optionality:	Optional
Allowed:	
Default:	
Notes:	The expiry date for the **Free SMS** balance type cannot be set with the `CCSCD3=RCH` PI command.
Example:	`BALANCE_EXPIRY=7`

WALLET_EXPIRY

Syntax: `WALLET_EXPIRY=`*`days`*

Description: The number of days from today's date before the wallet expires.

Type: Integer

Optionality: Optional

Allowed:

Default:

Notes:

Example: `WALLET_EXPIRY=7`

MODE

Syntax: `MODE=`*`int`*

Description: Determines which recharge amounts are used.

Type: Integer

Optionality: Optional

Allowed:

1	Use provided values.
2	Use profile values.
3	Use whichever is greater, provided values or profile values.
4	Extend current values by specified amounts.

Default: 1

Notes:

Example: `MODE=3`

WALLET_TYPE

Syntax: `WALLET_TYPE=`*`str`*

Description: The name of the wallet which is the focus of this command.

Type: String

Optionality: Optional

Allowed:

Primary	Primary wallet
Secondary	Secondary wallet

Default:

Notes:

Example: `WALLET_TYPE=Primary`

BALANCE_TYPE

Syntax: `BALANCE_TYPE=`*`str`*

Description: The name of the balance to be recharged.

Type: String

Optionality: Optional

Allowed:

Default:

Notes:

Example: `BALANCE_TYPE=General Cash`

`LIMIT_TYPE`

Syntax: `LIMIT_TYPE=`*`str`*

Description: The limit type of a new balance for custom recharges.

Type: String

Optionality: Optional

Allowed:
Debit
Single Use
Credit
Limited Credit

Default: Debit

Notes: For custom recharges only.

Example: `LIMIT_TYPE=Credit`

`EXTRA_EDR`

Syntax: `EXTRA_EDR="`*`KEY1=VALUE1`*`|`*`KEY2=VALUE2`*`|...]"`

Description: A list of EDR fields to add.

Type: Array

Optionality: Optional

Allowed: List can be up to 100 characters long.

Default:

Notes:

Example: `EXTRA_EDR="TEST1=1|TEST2=2"`

`BALMODE`

Syntax: `BALMODE=`*`str`*

Description: Determines how the BALANCE value is treated.

Type: String

Optionality: Optional

Allowed:
ABSOLUTE
DELTA

Default: DELTA

Notes: This parameter is only valid for custom recharges and will be ignored for credit card recharges.

Example: `BALMODE=ABSOLUTE`

Logic and constraints

Here are the general rules that apply when using the CCSCD3=RCH command on a VWS domain.

1. You must specify either the MSISDN or the ACCOUNT_NUMBER. If you specify both, then the MSISDN valued will be ignored.
2. If you just specify the MSISDN, then the ACCOUNT_NUMBER will be derived.
3. The specified or derived ACCOUNT_NUMBER must exist in the database.
4. The subscriber account will only be recharged if it is currently in one of these states: Active, Dormant or Pre-Use.

5 If the new BALANCE_EXPIRY and WALLET_EXPIRY value means that the new expiration dates are sooner than the current expiration dates, the current dates will be used.

6 The AMOUNT may be modified by a bonus percentage specified in the Prepaid Charging bonus tables.

7 If WALLET_TYPE is not specified then the default wallet type will be used.

8 If BALANCE_TYPE is not specified then the default balance type will be used.

Custom recharges

Here are the additional rules that apply when using the CCSCD3=RCH command for custom recharges.

1 All values are taken from the input parameters, except BALANCE_EXPIRY and WALLET_EXPIRY where account default values will be used if not specified.

2 A type 8 EDR record will be generated by the billing engine for positive amounts and a type 2 for negative amounts.

3 For negative amounts, the last recharge date is not updated.

4 If no entry exists in the reload bonus table, a bonus of 0% is used (that is, no bonus).

5 If LIMIT_TYPE is not specified, the limit type defaults to "Debit" (that is, Prepaid).

6 If BALMODE is specified, AMOUNT must also be specified.

7 If BALMODE is “ABSOLUTE” then:

- If MODE is 1, and BALANCE_EXPIRY is not specified, the new balance will be set to not expire.
- If MODE is 2, and there is no default balance expiry extension, the new balance will be set to not expire.
- If MODE is 3, and BALANCE_EXPIRY is not specified and there is no default expiry date, the new balance will be set to not expire.
- If MODE is 4, the balance expiry date is treated as if MODE is 1, that is, the days from today, not days from the previous balance expiry date.

Credit card recharges

Here are the additional rules that apply when using the CCSCD3=RCH command for credit card recharges.

1 All values are taken from the input parameters.

2 The REFERENCE must already be in the pending queue.

3 The credit amount must be covered by an entry in the reload bonus table (bonus type/bonus value combination), or an error is returned. The entry in the reload bonus table may be zero (that is, no bonus).

4 A credit card recharge will not actually be carried out, but the existing pending recharges in the recharge queue will be verified and the wallet and buckets will be adjusted accordingly.

5 The BALANCE_EXPIRY and WALLET_EXPIRY periods placed into the transaction will be the new dates calculated from the current date.

6 A type 9 EDR record will be generated by the billing engine.

Credit Transfer

Description

Use the command CCSCD3 and action CTR to perform a credit transfer between two subscribe accounts.

To successfully execute this command the subscribers must be on a VWS domain.

The transfer is performed by triggering a call plan using the SMS trigger daemon. The default name for the call plan is "CREDIT TRANSFER". To use a different call plan, set the `pi.CCSCD3.CTR.creditTransferCP` parameter in **eserv.config**.

Required parameters

This table describes each required parameter.

Parameter	Description
ORIG_MSISDN	The CLI for the subscriber account initiating the transfer.
DEST_MSISDN	The CLI for the subscriber account receiving the transfer.
CT_TYPE	The short name of the credit transfer type.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Parameter	Description
ORIG_WALLET	The name of the wallet type to transfer credit from. Values are: • Primary • Secondary
DEST_WALLET	The name of the wallet type transfer credit to. Values are: • Primary • Secondary
PIN	The PIN number for the account initiating the transfer.

Logic and constraints

Here are the rules that apply when using the CCSCD3=CTR command on a VWS domain.

1. You must specify existing MSISDNs for the ORIG_MSISDN and DEST_MSISDN parameters.
2. The CT_TYPE must match the short name of a defined credit transfer type.
3. If ORIG_WALLET is specified, it should be a defined wallet type in the originating account.
4. If ORIG_WALLET is not specified, the default wallet type will be used.
5. If DEST_WALLET is specified, it should be a defined wallet type in the destination account.
6. If DEST_WALLET is not specified, the default wallet type will be used.
7. If PIN is specified it should match the originating account's PIN.
8. If PIN is not specified, a PIN-less transfer will be attempted. If the originating account has a PIN, the transfer will be rejected.

Query EDRs for Subscriber Account Details

Description

Use the command CCSCD7 and action QRY to query EDR records for subscriber account and wallet details. The most recent EDR records will be returned first.

To successfully execute this command the subscribers must be on a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>WALLET_TYPE</td><td>The name of the wallet to query. Values are:
• Primary
• Secondary</td></tr>
<tr><td>EDR_TYPE</td><td>Pip ("|") separated list of EDR types to return.</td></tr>
<tr><td>MAX_RECORDS</td><td>Sets the maximum number of EDRs to return (default = 5). If specified, this must be greater than 0.
<b>Note:</b> MAX_RECORDS refers to the size of the unfiltered resultset and not the size of the resultset after the filters have been applied.</td></tr>
<tr><td>DAYS</td><td>Sets the number of days data to return. If specified, this must be greater than or equal to 0. When set to 0 just today's data will be returned.</td></tr>
<tr><td>START_DATE</td><td>Sets the date for the earliest EDR to return.
This parameter will be ignored if DAYS is specified.</td></tr>
<tr><td>END_DATE</td><td>Sets the date for the latest EDR to return.
This parameter will be ignored if DAYS is specified.</td></tr>
<tr><td>TAG
VALUE</td><td>The parameters, TAG and VALUE work together.
When set, returns records containing only the values mentioned in the TAG and VALUE. You can have multiple entries for TAG and VALUE.
<b>Note:</b> The number of entries in TAG should be equal to the number of entries in the VALUE. If the number of entries are not equal, the command returns an error.
For example:
<code>CCSCD7=QRY:MSISDN=1234,TAG=EVENT_CLASS|EVENT_CLASS,VALUE=SMS |DATA, EDR_TYPE=5,MAX_RECORDS=100000</code>
<b>Result:</b> Displays the records containing both the <code>EVENT_CLASS=SMS</code> and <code>EVENT_CLASS=DATA</code></td></tr>
</table>

Logic and constraints

Here are the rules that apply when using the CCSCD7=QRY command on a UBE domain.

1. You must specify an existing MSISDN.
2. If you specify EDR_TYPE, then only EDRs of the specified types will be returned. For details on EDR types, see *Convergent Charging Controller Event Detail Record Reference Guide*.
3. If you specify WALLET_TYPE, only records for that wallet will be returned. Otherwise, records for the default wallet for the subscriber will be returned.

4 If you specify MAX_RECORDS, then only the first MAX_RECORDS records will be returned.

5 If you specify DAYS, then data for the last DAYS days is returned, starting from midnight (00:00:00) local time of the earliest day.

6 If you specify START_DATE, then no EDRs earlier than that date will be returned.

7 If you specify END_DATE, then no EDRs later than that date will be returned.

8 If both START_DATE and END_DATE are specified, END_DATE must be later than START_DATE.

9 START_DATE and END_DATE must be specified in local time.

10 RECORD_DATE will be returned in local time.

11 Any dates in EXTRA_INFORMATION will be returned unaltered from the database record and will be in GMT.

Change Credit Card Recharge Request Status

Description

Use the command CCSCC1 and action CHG to change the status of a specific credit card recharge request transaction.

To successfully execute this command the subscribers must be on a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
REFERENCE	The recharge reference ID.
STATUS	Sets the state of the recharge request. Must be set to one of the following: • "P" - pending • "Y" - verified • "N" - invalid.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

No optional parameters are supported.

Logic and constraints

Here are the rules that apply when using the CCSCC1=CHG command on a VWS domain.

1 You must specify an existing REFERENCE for a credit card recharge request.

2 You must specify a valid STATUS.

Note: When STATUS is set to either "Y" or "N", the credit card details recharge pending flag is cleared.

Query Credit Card Recharge Requests

Description

Use the command CCSCC1 and action QRY to query credit card recharge records for details of all the transactions at a specified status. The following information will be returned:

- The MSISDN for the subscriber
- The credit card number
- The credit card expiry date
- The recharge reference id
- The amount to recharge
- The recharge date

To successfully execute this command the subscribers must be on a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
STATUS	The recharge request status to query on. Values are: • "P" – Pending • "Y" – Verified • "N" – Invalid
MAX_RECORDS	The maximum number of records to return. Must be between 1 and 100.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Parameter	Description
START_RECORD	Sets the record number of the first record to be returned. For example, set START_RECORD to 20 to begin returning records from the 20th record.

Logic and constraints

Here are the rules that apply when using the CCSCC1=QRY command on a VWS domain to query credit card recharge request records.

1. You must specify a number between 1 and 100 for MAX_RECORDS.
2. You must specify a valid STATUS.

Add Service Charge

Description

Use the command CCSSC1 and action ADD to add or refund a service charge to the subscriber's account.

To successfully execute this command the subscribers must be on a VWS domain.

Required parameters

There are no required parameters for this command.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber this command applies to.
ACCOUNT_NUMBER	The account reference number. When provided, this overrides any MSISDN value.
CLASS	Sets the event class name
EVENT	Sets the event charge name
CHARGE	Sets the charge amount. Use small currency units for all balances except Time balances. Specify the amount in seconds for Time balances.

Optional parameters

This table describes each optional parameter.

Parameter	Description
REFUND	Determines whether or not the calculated charge should be refunded. Values are: • 1 – Refund • 0 – Charge (default)
WALLET_TYPE	The name of the wallet to be charged. Values are: • Primary • Secondary
BALANCE_TYPE	The name of the balance to be charged.
EXTRA_EDR	List of EDR fields to add using the following format: "*KEY1=VALUE1|KEY2=VALUE2*" This can be up to 100 characters long.
DATE	The charge or refund is made at the specified date (so if the cost of an event changes on weekends for example, this allows charging/refunding of the event at the given date).

Logic and constraints

Here are the rules that apply when using the CCSSC1=ADD command on a VWS domain to add or refund a service charge.

1 You must specify either a CHARGE, or an EVENT and CLASS, or all three.

2 Either the MSISDN or the ACCOUNT_NUMBER must be specified. If both are specified the MSISDN is ignored. If only the MSISDN is specified, then the ACCOUNT_NUMBER is obtained from the account reference.

3 If CHARGE is specified, this will override the charge specified in the event.

4 A positive CHARGE is a positive amount for a debit and negative amount for a credit.

5 If REFUND=1 is specified, the calculated or specified charge is applied as a refund rather than a charge (that is, the amount is multiplied by –1). This allows the operator to refund for an event without having to query the database to find the charge for this event.

6 If WALLET_TYPE is not specified it will default to the default value defined in the database.

Query Service Charge

Description

Use the command CCSSC1 and action QRY to look for service charge details.

To successfully execute this command the subscriber must be on a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
CLASS	The event class name to look for.
EVENT	The event charge name to look for.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
PROVIDER	The service provider name.

Optional parameters

This table describes each optional parameter.

Parameter	Description
PRODUCT	The product type name. This must be a defined product type for the specified service provider.
DATE	The charge or refund made at the specified date.
MSISDN	The CLI for the subscriber this command applies to.

Logic and constraints

Here are the rules that apply when using the CCSSC1=QRY command on a VWS domain to look for service charge details.

1. To return default values for the class and event for the service provider, specify the CLASS, EVENT, and PROVIDER parameters.
2. To return the values possibly overridden by the product type for the class and event for the service provider, specify the CLASS, EVENT, PROVIDER, and PRODUCT parameters.
3. To return the values possibly overridden by the product type for the MSISDN, specify one of the following:
   - CLASS, EVENT, and MSISDN parameters
   - All the parameters
4. The specified CLASS and EVENT must be for a defined named event for PROVIDER or MSISDN.
5. The MSISDN, if specified, must be for an existing subscriber.
6. If you specify the PRODUCT and not the MSISDN then you must also specify PROVIDER.
7. If you specify the MSISDN, PRODUCT and PROVIDER parameters, then PRODUCT will override the current product for the MSISDN.
8. If you do not specify a PRODUCT or MSISDN, the information will be supplied from the SMF database as no overrides will be calculated.
9. If you supply the PRODUCT but no MSISDN, the VWS as configured by the defaultBEDomainID parameter will be queried.
10. If you supply a MSISDN, the VWS for that subscriber’s wallet will be queried.

Add Charge for Call

Description

Use the command CCSSC2 and action ADD to add or refund a charge as if a call had been made.

To successfully execute this command the subscribers must be on a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber account this command applies to. This must exist already.
FROM	The originating number.
TO	The destination number.
DURATION	The length of the call in seconds.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

Parameter	Description
COMMENT	User's remarks/description of up to 50 characters.
DATE	Sets the date for the charge. If not specified then the current date is used for calculating the charge.
REFUND	Determines whether or not the calculated charge should be charged or refunded. Valid values are: • 1 - Refund • 0 - Not a refund (default)
WALLET_TYPE	The name of the wallet to be charged. Either:
SMSNOT	The name of the Notification template to use for sending notifications.
EXTRA_EDR	List of EDR fields to add using the following format: "*KEY1=VALUE1|KEY2=VALUE2*" This can be up to 100 characters long.

Logic and constraints

Here are the rules that apply when using the CCSSC2=ADD command on a VWS domain to add or refund a service charge.

1. You must specify valid values for the MSISDN, FROM and TO parameters.
2. If you specify REFUND=1, then the calculated or specified charge will be applied as a refund rather than a charge.
3. If WALLET_TYPE is not specified it will default to the default value defined in the database.
4. If SMSNOT is specified, and the notification flag in the subscriber's profile is set to true, a notification will be sent to the subscriber through the `ccsSSMDispatcher` notification system on the SMS.

Query Subscribed Periodic Charges

Description

Use the command CCSSC3 and action QRY to query what a subscriber's subscription status is to a periodic charge.

The query returns:

- Current state
- Next state change date
- Number of charges remaining for each charge listed\

To successfully execute this command the subscribers must be on a VWS domain.

Required parameters

CCSSC3 requires the following parameters.

`MSISDN`

Syntax: `MSISDN=`*`CLI`*

Description: The CLI for the subscriber account this command applies to.

Type:	Integer
Optionality:	Mandatory
Allowed:	The CLI must have already been set up as a valid subscriber in the CCS_ACCT_REFERENCE table.
Default:	None
Notes:	
Example:	`MSISDN=0123456789`

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

CCSSC3 accepts the following optional parameters.

`CHARGE`

Syntax:	`CHARGE=`*`charge`*
Description:	The name of the periodic charge to query.
Type:	String
Optionality:	Optional (if not specified, all valid charges are returned).
Allowed:	
Default:	None
Notes:	If set, this must match an existing periodic charge.
Example:	`CHARGE=Weekly Charge`

`WALLET_TYPE`

Syntax:	`WALLET_TYPE=`*`str`*
Description:	The name of the wallet which is the focus of this command.
Type:	String
Optionality:	Optional
Allowed:	Primary — Primary wallet Secondary — Secondary wallet
Default:	
Notes:	
Example:	`WALLET_TYPE=Primary`

Logic and constraints

The following rules apply when using the CCSSD3=QRY command:

- The MSISDN must be a defined subscriber.
- The CHARGE must be a periodic charge that is valid for the MSISDN and WALLET_TYPE.
- If WALLET_TYPE is not specified, it will default to the default value defined in the database.

Example output

Here is an example of the ccssc3 QRY output.

```
>CCSSC3=QRY:MSISDN=999,SYNSTAMP=2009100718591095,MAC=100000;
<CCSSC3=QRY:ACK:CHARGE=CBA_PC_Test2,STATE=Active,SUBSCRIPTION_DATE=20091005112310,NE
XT_STATE_CHANGE_DATE=20091028000002,NEXT_CHARGE=20091028000002,CHARGES_REMAINING=12,
SYNSTAMP=2009100718591095;
```

Chapter 4

PI Voucher Package

Overview

Introduction

This chapter describes the Oracle Communications Convergent Charging Controller provisioning interface (PI) commands for managing vouchers on a VWS domain.

These commands are installed by the `piVoucherSms` package. This extends the `piVoucherSms` package by providing enhancements to existing commands and adding new commands.

In this chapter

This chapter contains the following topics.

Voucher Recharge

Description

Use the command CCSCD3 and action RCH to perform the following recharge functions on a subscriber's account:

- Voucher recharge
- Voucher type recharge
- Custom recharge
- Verify recharge (credit recharge)

To successfully execute this command the subscriber must be on a VWS domain.

Required parameters

This table describes each required parameter.

Parameter	Description
RECHARGE_TYPE	The type of recharge to perform. Valid types are: • Voucher • VoucherType • Credit • Custom

Parameter	Description
REFERENCE	Holds either the voucher number or the recharge requests id. For voucher recharges this will be the voucher number. For VoucherType recharges this will be the voucher type name. For credit recharges this will be the reference id of the pending recharges. For custom recharges this will be carried over in the EDR into the Reference EDR field.

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
MSISDN	The CLI for the subscriber this command applies to.
ACCOUNT_NUMBER	The account reference number.
AMOUNT	The recharge amount. Use small currency units for all balances except Time balances. Specify the amount in seconds for Time balances.

Optional parameters

CCSCD3 accepts the following optional parameters.

`BALANCE_EXPIRY`

Syntax: `BALANCE_EXPIRY=`*`days`*

Description: The number of days from today's date before the balance expires.

Type: Integer

Optionality: Optional

Allowed:

Default:

Notes:

Example: `BALANCE_EXPIRY=7`

`WALLET_EXPIRY`

Syntax: `WALLET_EXPIRY=`*`days`*

Description: The number of days from today's date before the wallet expires.

Type: Integer

Optionality: Optional

Allowed:

Default:

Notes:

Example: `WALLET_EXPIRY=7`

`MODE`

Syntax: `MODE=`*`int`*

Description: Determines which recharge amounts are used.

Type: Integer

Optionality: Optional

Allowed:	1 Use provided values. 2 Use profile values. 3 Use whichever is greater, provided values or profile values. 4 Extend current values by specified amounts.
Default:	1
Notes:	
Example:	`MODE=3`

WALLET_TYPE

Syntax:	`WALLET_TYPE=`*`str`*
Description:	The name of the wallet which is the focus of this command.
Type:	String
Optionality:	Optional
Allowed:	Primary Primary wallet Secondary Secondary wallet
Default:	
Notes:	
Example:	`WALLET_TYPE=Primary`

BALANCE_TYPE

Syntax:	`BALANCE_TYPE=`*`str`*
Description:	The name of the balance to be recharged.
Type:	String
Optionality:	Optional
Allowed:	
Default:	
Notes:	
Example:	`BALANCE_TYPE=General Cash`

LIMIT_TYPE

Syntax:	`LIMIT_TYPE=`*`str`*
Description:	The limit type of a new balance for custom recharges.
Type:	String
Optionality:	Optional
Allowed:	Debit Single Use Credit Limited Credit
Default:	Debit
Notes:	For custom recharges only.
Example:	`LIMIT_TYPE=Credit`

EXTRA_EDR

Syntax: `EXTRA_EDR="`*`KEY1=VALUE1`*`|`*`KEY2=VALUE2`*`|...]"`

Description: A list of EDR fields to add.

Type: Array

Optionality: Optional

Allowed: List can be up to 100 characters long.

Default:

Notes:

Example: `EXTRA_EDR="TEST1=1|TEST2=2"`

BALMODE

Syntax: `BALMODE=`*`str`*

Description: Determines how the BALANCE value is treated.

Type: String

Optionality: Optional

Allowed:
ABSOLUTE
DELTA

Default: DELTA

Notes: This parameter is only valid for custom recharges and will be ignored for credit card recharges.

Example: `BALMODE=ABSOLUTE`

SCENARIO_ID

Syntax: `SCENARIO_ID=`*`id`*`|`*`id`*

Description: The ID of the Scenario to apply in a voucher recharge.

Type: Integer

Optionality:

Allowed: 0-9

Default: 0

Notes: Cannot be used if SCENARIO_NAME is set.

This parameter is valid only for Voucher recharges and will be ignored for VoucherType recharges.

Scenario must be defined for the voucher. Scenarios are configured on the New/Edit Voucher Type screen. For more information, see *CCS User's Guide*.

Example: `SCENARIO_ID=2|3|4`

SCENARIO_NAME

Syntax: `SCENARIO_NAME=`*`name`*`|`*`name`*`...]`

Description: The name of the Scenario to apply in the voucher recharge.

Type: String

Optionality:

Allowed: Any valid scenario name up to 50 characters long.

Default: Whatever pi.CCSCD3.RCH.defaultScenarioName entry in the **eserv.config** file is set to. (It defaults to "Default").

Notes:	Cannot be used if SCENARIO_ID is set. This parameter is valid only for Voucher recharges and will be ignored for VoucherType recharges. Scenario must be defined for the voucher. Scenario names are configured on the New and Edit Voucher Type screen. For more information, see *CCS User's Guide*. If this parameter is used, the voucher number length (without the PIN) is expected to be 10. This can be altered by setting the pi.CCSCD3.RCH.fixedVoucherNumberLength entry in the **eserv.config** file.
Example:	`SCENARIO_NAME=Promotion Scenario`

Logic and constraints

Here are the general rules that apply when using the CCSCD3=RCH command on a VWS domain.

1. You must specify either the MSISDN or the ACCOUNT_NUMBER. If you specify both, then the MSISDN valued will be ignored.
2. If you just specify the MSISDN, then the ACCOUNT_NUMBER will be derived.
3. The specified or derived ACCOUNT_NUMBER must exist in the database.
4. The subscriber account will only be recharged if it is currently in one of these states: Active, Dormant or Pre-Use.
5. The REFERENCE parameter is used to hold the voucher number, voucher type or the recharge requests ID.
6. If the new BALANCE_EXPIRY and WALLET_EXPIRY value means that the new expiration dates are sooner than the current expiration dates, the current dates will be used.
7. For custom and credit card recharges, the AMOUNT may be modified by a bonus percentage specified in the Prepaid Charging bonus tables. A bonus recharge by voucher type may also occur, if specified in the bonus tables.
8. If WALLET_TYPE is not specified then the default wallet type will be used.
9. If BALANCE_TYPE is not specified then the default balance type will be used.

Custom recharges

Here are the additional rules that apply when using the CCSCD3=RCH command for custom recharges.

1. All values are taken from the input parameters, except BALANCE_EXPIRY and WALLET_EXPIRY where account default values will be used if not specified.
2. A type 8 EDR record will be generated by the billing engine for positive amounts and a type 2 for negative amounts.
3. For negative amounts, the last recharge date is not updated.
4. If no entry exists in the reload bonus table, a bonus of 0% is used (that is, no bonus).
5. If LIMIT_TYPE is not specified, the limit type defaults to "Debit" (that is, Prepaid).
6. If BALMODE is specified, AMOUNT must also be specified.
7. If BALMODE is “ABSOLUTE” then:
   - If MODE is 1, and BALANCE_EXPIRY is not specified, the new balance will be set to not expire.
   - If MODE is 2, and there is no default balance expiry extension, the new balance will be set to not expire.
   - If MODE is 3, and BALANCE_EXPIRY is not specified and there is no default expiry date, the new balance will be set to not expire.
   - If MODE is 4, the balance expiry date is treated as if MODE is 1, that is, the days from today, not days from the previous balance expiry date.

Credit card recharges

Here are the additional rules that apply when using the CCSCD3=RCH command for credit card recharges.

1. All values are taken from the input parameters.
2. The REFERENCE must already be in the pending queue.
3. The credit amount must be covered by an entry in the reload bonus table (bonus type/bonus value combination), or an error is returned. The entry in the reload bonus table may be zero (that is, no bonus).
4. A credit card recharge will not actually be carried out, but the existing pending recharges in the recharge queue will be verified and the wallet and buckets will be adjusted accordingly.
5. The BALANCE_EXPIRY and WALLET_EXPIRY periods placed into the transaction will be the new dates calculated from the current date.
6. A type 9 EDR record will be generated by the billing engine.

Credit card recharges

Here are the additional rules that apply when using the CCSCD3=RCH command for voucher and voucher type recharges.

1. For Voucher recharges:
   - You must specify an existing voucher.
   - The voucher can only be used if it's either Active or Pending and it is within its effective period.
   - The signature provided must match the signature of the voucher.
2. For Voucher Type recharges:
   - You must specify an existing voucher type
   - The REFERENCE must start with the voucher type name. Optionally, it can be followed by an external reference, separated by the pipe symbol ("|"). If the external reference is specified, it will be added to the EDR as a REFERENCE parameter. The external reference may not contain the separator characters (comma or pipe).
   - The wallet will be recharged as if a voucher of that type had been used.
3. A Voucher or Voucher Type recharge can be:
   - Multi-Balance
   - Multi-Balance with Limited Product Type
   - Product Type Change
   - A combination of all three
4. When a voucher is redeemed the balances, expiry dates, or product will be altered via a billing engine message.
5. AMOUNT, BALMODE, MODE, BALANCE_EXPIRY and WALLET_EXPIRY are ignored for voucher and voucher type recharges. The value and expiry changes are all taken from the specified voucher or voucher type as appropriate.
6. If the voucher recharge includes multi-balances and limited product types, then:
   - Type 4 and type 15 EDR records will be generated by the billing engine
   - One of the voucher products must be the wallet’s current product
7. If the voucher recharge includes a product type change and one of the “From” products is this wallet’s current product then:
   - the wallet’s product will be changed to the corresponding voucher “To” product
   - a type 31 or 32 EDR record will be generated by the billing engine.

Change Voucher Status

Description

Use the the command CCSVR1 and action CHG to change the status of a voucher recharge to one of the following:

- Active
- Frozen

To successfully execute this command the subscriber must be on a VWS domain.

Note: This command cannot be used to change the status for voucher batches because the status is stored for the whole batch.

Required parameters

This table describes each required parameter.

Parameter	Description
STATUS	Sets the new status of the voucher. Valid values are: • "A" - Active • "F" - Frozen

Constraint required parameters

This table describes each constraint required parameter. You must specify either the voucher serial number or the voucher reference number.

Parameter	Description
PROVIDER	The service provider name.
SERIAL	The serial number for the voucher.
VOUCHER	The reference number for the voucher.

Optional parameters

This table describes each optional parameter.

Parameter	Description
DESCRIPTION	The reason (limited to 50 characters) for a voucher state change.

Logic and constraints

Here are the rules that apply when using the CCSVR1=CHG command on a VWS domain to change the status of a voucher.

1. You must specify an existing voucher. You can use the voucher number or the serial number.
2. You must specify a valid status. Valid values are:
   - A - Active
   - F - Frozen
3. If the VOUCHER is not unique, then you must specify the PROVIDER.

Freeze Voucher

Description

Use the command CCSVR1 and action FRZ to set the status of a specified prepaid voucher to frozen.

To successfully execute this command the subscriber must be on a VWS domain.

Required parameters

There are no required parameters for this command.

Constraint required parameters

This table describes each constraint required parameter. You must specify either the voucher serial number or the voucher reference number.

Parameter	Description
PROVIDER	The service provider name.
SERIAL	The serial number for the voucher.
VOUCHER	The reference number for the voucher.

Optional parameters

This table describes each optional parameter.

Parameter	Description
DESCRIPTION	The reason (limited to 50 characters) for a voucher state change.

Logic and constraints

Here are the rules that apply when using the CCSVR1=FRZ command on a VWS domain to freeze a voucher.

1. You must specify an existing voucher. The voucher must not have been used and it must not be frozen already. You can use the voucher number or the serial number.
2. If the VOUCHER is not unique, then you must specify the PROVIDER.

Query a Recharge Voucher

Description

Use the command CCSVR1 and action QRY to look for a specified recharge voucher. The following voucher details will be returned by default:

- Voucher reference number
- Voucher serial number
- Balances
- Voucher creation date
- Voucher batch activation date
- Voucher expiry date
- Wallet expiry

- Voucher status
- Voucher type
- Service provider
- List of valid product types
- List of product changes
- List of associated Scenarios
- If redeemed, the MSISDN and account reference of the redeeming CLI, and the voucher redeemed date.
- If the voucher state has changed, the reason for the change and the user who made the change.

To successfully execute this command the subscriber must be on a VWS domain.

To stop specified fields from being returned, such as the voucher serial number, specify the list of fields to suppress in the `suppressFields` parameter in the PI section of **eserv.config**. See *PI User's and Technical Guide* for more information.

Required parameters

There are no required parameters for this command.

Constraint required parameters

This table describes each constraint required parameter. You must specify either the voucher serial number or the voucher reference number.

Parameter	Description
PROVIDER	The service provider name.
SERIAL	The serial number for the voucher.
VOUCHER	The reference number for the voucher.

Optional parameters

No optional parameters are supported.

Logic and constraints

Here are the rules that apply when using the CCSVR1=QRY command on a VWS domain to find voucher details.

1 You must specify an existing voucher. You can use the voucher number or the serial number.
2 If the VOUCHER is not unique, then you must specify the PROVIDER.
3 If you used the voucher number, the voucher number should not have the PIN appended.

Chapter 5

PI SNST Package

Overview

Introduction

This chapter describes the Oracle Communications Convergent Charging Controller provisioning interface (PI) commands for Social Networking Service Template (SNST) users.

These commands are installed by the `piSrmSms` package.

In this chapter

This chapter contains the following topics.

Add Friends

Description

Use the command CCSSR1 and action ADD to add a friend or list of friends to the SNST user's friend list.

This command executes a control plan which will perform one of the following:

- Add the friend(s) to the subscriber's profile and send an SMS to each party confirming the action taken
- Return an error if the subscriber is not registered as an SNST user.

Note: If you are adding a list of friends to the SNST user and an add friend request fails, then subsequent friends in the list will not be processed.

Required parameters

CCSSR1=ADD requires the following parameters.

`MSISDN`

Syntax:	`MSISDN=`*`CLI`*
Description:	The CLI for the subscriber account this command applies to.
Type:	Integer
Optionality:	Required except when using the CCSSR1=QRY command.
Allowed:	The CLI must have already been set up as a valid SNST user in the CCS_ACCT_REFERENCE table.
Default:	None

Notes:

Example: MSISDN=0123456789

Constraint required parameters

CCSSR1=ADD accepts the following constraint required parameters.

FRIEND

Syntax: FRIEND=*CLI*

Description: The CLI of the friend to be added to the SNST friend list.

Type: Integer

Optionality: Optional.

Allowed:

Default: None.

Notes: Either the FRIEND or the FRIENDLIST parameter must be specified, but not both.

Example: FRIEND=0123456789

FRIENDLIST

Syntax: FRIENDLIST="*CLI1*|*CLI2*...>"

Description: Pipe "|" separated list of CLIs to be added to the subscriber's SNST friend list.

Type: String

Optionality: Optional.

Allowed:

Default: None

Notes: Either the FRIEND or the FRIENDLIST parameter must be specified.

Example: FRIENDLIST="0123456789|2123456789"

Optional parameters

No optional parameters are supported.

Logic and constraints

Here are the rules that apply when using the CCSSR1=ADD command to add friend(s) to the SNST friend list.

1. Either FRIEND or FRIENDLIST must be specified.
2. If the friend being added does not exist in the database then a record will be created for them.

Remove a Friend

Description

Use the command CCSSR1 and action DEL to remove a friend from the SNST user's friend list.

This command executes a control plan which will perform one of the following:

- Remove the friend from the subscriber's profile and send an SMS to each party confirming the action taken

- Return an error if the subscriber is not registered as an SNST user

Required parameters

CCSSR1=DEL requires the following parameters.

MSISDN

Syntax: MSISDN=*CLI*

Description: The CLI for the subscriber account this command applies to.

Type: Integer

Optionality: Required except when using the CCSSR1=QRY command.

Allowed: The CLI must have already been set up as a valid SNST user in the CCS_ACCT_REFERENCE table.

Default: None

Notes:

Example: MSISDN=0123456789

FRIEND

Syntax: FRIEND=*CLI*

Description: The CLI of the friend to be removed from the SNST friend list.

Type: Integer

Optionality: Required.

Allowed: The friend must be a valid SNST friend for the subscriber.

Default: None

Notes:

Example: FRIEND=0123456789

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

No optional parameters are supported.

Query SNST Information

Description

Use the command CCSSR1 and action QRY to query SNST information. You can retrieve the:

- Subscriber's list of friends
- Current size of the subscriber's friend list and size limit
- Subscriber's current SNST tariff or discount
- SNST specific service bundles and promotions
- SNST balances
- SNST user status

Required parameters

CCSSR1=QRY requires the following parameter.

`LIST_TYPE`

Syntax: `LIST_TYPE=(`*`parameter1`*`|`*`parameter2`*`|..`*`parametern`*`>)`

Description: Pipe "|" separated list of items to return.

Type: String

Optionality: Required.

Allowed:

STATUS	Retrieve the SNST user status.
FRIENDS	Retrieve a subscriber's list of friends.
SIZE	Retrieve the subscriber's friend list size.
LIMIT	Retrieve the limit to which the friend list may grow.
TARIFF	Retrieve a subscriber's current SNST tariff.
DISCOUNT	Retrieve a subscriber's current SNST discount.
SERVICE_BUNDLES	Retrieve the SNST specific service bundles.
PROMOTIONS	Retrieve the SNST Promotion Manager balances and targets

Default:

Notes:

Example:

```
LIST_TYPE=(FRIENDS|SIZE|LIMIT|TARIFF|DISCOUNT|SERVICE_BUNDLE
S|STATUS|PROMOTIONS)
```

Constraint required parameters

CCSSR1=QRY accepts the following constraint required parameters.

`MSISDN`

Syntax: `MSISDN=`*`CLI`*

Description: The CLI for the subscriber account this command applies to.

Type: Integer

Optionality: Required except when using the CCSSR1=QRY command.

Allowed: The CLI must have already been set up as a valid SNST user in the CCS_ACCT_REFERENCE table.

Default: None

Notes:

Example: `MSISDN=0123456789`

`PROVIDER`

Syntax: `PROVIDER="`*`service_provider_name`*`"`

Description: The name of the service provider.

Type: String.

Optionality: Optional.

Allowed:	The service provider must exist in the database.
Default:	Boss
Notes:	
Example:	`PROVIDER="Telco"`

PRODUCT_TYPE

Syntax:	`PRODUCT_TYPE="`*name*`"`
Description:	The name of the product type of the SNST subscriber.
Type:	String
Optionality:	Optional.
Allowed:	The product type must exist on the database.
Default:	None
Notes:	
Example:	`PRODUCT_TYPE="SRM"`

Optional parameters

No optional parameters are supported.

Logic and constraints

Here are the rules that apply when using the CCSSR1=QRY command to query and SNST friend list.

1. The SRM subscriber must exist in the database.
2. MSISDN must be specified for all LIST TYPE values except SERVICE_BUNDLES.
3. If the SERVICE_BUNDLES list type is specified then PROVIDER and PRODUCT_TYPE must be specified.
4. The DISCOUNT list type uses the discount tags defined in **eserv.config**.
5. The TARIFF list type uses the tariff names defined in **eserv.config**.

Register for SNST

Description

Use the command CCSSR2 and action ADD to register a subscriber for the SNST service.

This command executes a control plan which will:

- Register the subscriber to the SNST service
- Send a confirmation SMS to the subscriber

Required parameters

CCSSR2=ADD requires the following parameters.

MSISDN

Syntax:	`MSISDN=`*CLI*
Description:	The CLI for the subscriber account this command applies to.
Type:	Integer
Optionality:	Required.

Allowed:

Default: None

Notes: The CLI must have already been set up as a valid subscriber in the CCS_ACCT_REFERENCE table.

Example: `MSISDN=0123456789`

`NICKNAME`

Syntax: `NICKNAME="`*`name`*`"`

Description: The nickname of the SNST user.

Type: String

Optionality: Required.

Allowed: NICKNAME must:

- Be less than 21 characters at least one of which must be a non-numeric character
- Not contain a key word such as YES, NO, STOP, REREG, or ALL

Default: None

Notes: The nickname specified does not have to be unique and is stored in the subscriber's profile.

Example: `NICKNAME="Jack"`

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

No optional parameters are supported.

Block SNST Service

Description

Use the command CCSSR2 and action DEL to block a subscriber's registration to the SNST service.

This command executes a control plan which will:

- Block the subscriber's registration to the SNST service
- Send a confirmation SMS to the subscriber

Note: The user may only block the SRM service if they have registered for SNST or if the have received a FRIEND request.

Required parameters

CCSSR2=DEL requires the following parameter.

`MSISDN`

Syntax: `MSISDN=`*`CLI`*

Description: The CLI for the subscriber account this command applies to.

Type: Integer

Optionality: Required except when using the CCSSR1=QRY command.

Allowed: The CLI must have already been set up as a valid SNST user in the CCS_ACCT_REFERENCE table.

Default: None

Notes:

Example: `MSISDN=0123456789`

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

No optional parameters are supported.

Chapter 6

Error Code Lists

Overview

Introduction

This chapter explains the error codes for Oracle Communications Convergent Charging Controller provisioning interface (PI) commands.

In this chapter

This chapter contains the following topics.

PI Chassis Errors

Error List

This table describes the PI Chassis error codes.

Code	Message	Description
70	`TOO MANY SESSIONS`	All PI sessions are in use.
71	`LOGON SYNTAX ERROR`	The login string was incorrectly formatted.
72	`INVALID LOGON - username, password`	Invalid username and/or password
73	`INVALID LOGON - user not allowed on this port`	The user attempted to log in to the wrong PI port.
74	`INVALID LOGON - host`	The PI client is unknown.
75	`UNKNOWN COMMAND`	Client sent an unknown command.
76	`USER DOES NOT HAVE SUFFICIENT SECURITY`	The user's security level is less than the command's security level.
77	`SYNSTAMP NOT FOUND`	Synstamps are turned on, but the client did not send one.
78	`SYNSTAMP NOT VALID`	Synstamps are turned on, but the synstamp sent by the client is invalid.
79	`INVALID OR MISSING CHECKSUM`	Checksums are turned on, but the client is one of the following: • Did not send one • It was invalid
80	`UNKNOWN PARAMETER FOR COMMAND`	A parameter was sent that was not valid for this command.
81	`MISSING PARAMETERS FROM COMMAND`	A required parameter is missing.

Code	Message	Description
82		Undefined
83	DUPLICATE PARAMETER	The client sent two identically named parameters.
84	ERROR RUNNING PROCEDURE	An internal error occurred running the command.
85	USER SESSION TERMINATED	The user's session has been terminated by an administrator.
86	COMMAND TOO BIG	The command sent is too long. Indicates an incorrectly formatted command.
87	COMMAND SYNTAX ERROR	The command sent is incorrectly formatted.
88	PARAMETER NAME TOO BIG	A parameter name is too long. Indicates the command was incorrectly formatted.
89	PARAMETER VALUE TOO BIG	A parameter value is too long. Indicates the command was incorrectly formatted.
90	SYNSTAMP OUT OF PLACE	The synstamp is not at the end of the command, but before the checksum.
91	TIMEOUT	The command took too long to run.

PI Command Errors

Error list

This table lists the PI command error codes and corresponding messages.

Code	Message
1	MSISDN <msisdn> already exists in the user table
2	PRODUCT <product> and PROVIDER <provider> are not a valid combination
3	LANGUAGE <language> does not exist in the language table
4	ACCOUNT_NUMBER <account number> does not exist
5	PRODUCT is null
6	PROVIDER is null
7	PRODUCT <product> does not exist
8	WALLET_TYPE <wallet type> is not valid
9	The account already has WALLET_TYPE <wallet type>
10	The CHARGING_DOMAIN_ID <id> does not exist
11	MSISDN <msisdn> does not exist
12	CURRENCY <currency> does not exist
13	PROVIDER <provider> does not exist
14	Expiry dates can not be set to a date in the past
15	BALANCE_TYPE <balance type> is not valid
16	Nothing to change
17	BALMODE <balmode> is not valid
18	<item> must be specified with BALMODE. <item> is BALANCE for CCSCD1=CHG and AMOUNT for CCSCD3=RCH.

Code	Message
19	MSISDN <msisdn> does not have WALLET_TYPE <wallet type>
20	No MSISDN or ACCOUNT_NUMBER specified
21	RECHARGE_TYPE <recharge_type> is not valid
22	REFERENCE <reference> is not valid
23	The account status <status> prohibits recharge for MSISDN <msisdn>
24	Voucher [Type] <voucher [type]> does not exist
25	LIMIT_TYPE <limit type> is not valid
26	Voucher <voucher> is not yet valid
27	Voucher <voucher> has expired
28	Voucher <voucher> has already been used
29	Voucher <voucher> is currently at status <status>
30	Unable to generate CDR
31	Unable to calculate expiry periods
32	CC_ALLOWED <cc_allowed> is not valid
33	CC_PIN_RESET <cc_pin_reset> is not valid
34	STATUS <status> is not a valid status
35	STATUS is already set to <status>
36	VOUCHER <voucher> is already frozen
37	CLASS is null
38	EVENT is null
39	CLASS <class> is not valid
40	EVENT <event> is not valid
41	CHARGE <charge> is not valid
42	Could not add CHARGE <charge>
43	LIST_TYPE <list type> is not valid
44	Attempt to add MSISDN <msisdn> that already exists in list
45	Attempt to delete MSISDN <msisdn> that does not exist in list
46	Too many numbers in ADD or DEL
47	Number <number> not in prefix map list
48	Maximum F&F numbers reached
49	Can’t specify both FFNUM and FDNUM
50	Number <number> is barred
51	FFNUM <number> does not exist
52	TYPE <type> is not valid
53	FFLIST has too many numbers
54	Bad combination of FFNUM, NEW_FFNUM and FFLIST
55	NEW_FFNUM already exists
56	Current F&D Number is not FDNUM <fdnum>
57	FROM <from number> is not valid

Code	Message
58	`TO <to number> is not valid`
59	`MAX_RECORDS <max_records> is out of range`
60	`The Primary wallet can not be deleted until the Secondary wallet has been deleted`
61	`Number <number> is not in the whitelist`
62	`Invalid limited product voucher <voucher> for cli <cli>`
63	`Bad CUG <CUG name> for cli <cli>`
64	`Could not link to existing wallet due to <reason>` where <reason> can be: • bad LW_MSISDN <cli> • bad LW_ACCOUNT_NUMBER <number> • LW_MSISDN <cli> has no WALLET_TYPE <type> • LW_ACCOUNT_NUMBER <number> has no WALLET_TYPE <type>
65	`Could not link to charging domain due to <reason>` where <reason> can be: • bad CHARGING_DOMAIN <charging domain> • bad WALLET_REFERENCE <wallet reference>
66	`Invalid operation for subscriber`
67	`Bad date range <start date> to <end date>`
68	`Badly formatted parameter <parameter>`
69	`<text>` where `<text>` reports any other undefined errors.
69	`Data object changing while query was executing for msisdn <MSISDN> please retry query later`
101	`TAG <tag name> does not exist`
102	`VALUE <value> is not valid`
103	`Wrong number of values for the number of tags`
104	`VALUE <value> failed limited value check`
105	`Could not merge MSISDNs <MSISDN1> and <MSISDN2>`
106	`Could not send SMS notification to MSISDN <MSISDN>`
107	`CT_TYPE <credit transfer type> is not defined`
108	`Incorrect PIN`
109	`BPL <bpl> not defined`
110	`User <user> security level is too low for BPL <bpl short name>`
111	`EXTRA_EDR is too long`
112	`Could not find a matching service charge`
113	`EXT<n> specified when EXT<n-1> is not specified`
114	`EXT<n> is invalid`
115	`SMS notification <SMSNOT> not defined`
116	`Call plan error: <text>`
117	`MSISDNs <START_MSISDN> to <END_MSISDN> do not define a valid range`

Code	Message
118	START_MSISDN and/or END_MSISDN specified when MSISDN or ACCOUNT_NUMBER is specified
119	Neither MSISDN nor START_MSISDN and END_MSISDN specified
120	START_MSISDN and END_MSISDN must be specified together
121	TARIFF_PLAN <tariff plan> does not exist
122	Could not determine a tariff plan
123	BPL error code: <BPL error> where <BPL error> is either as returned to PI from the trigger daemon, or as modified by the BPL error translation table, available in the CCS screens.
124	Closed user group already exists
125	Too many zone definitions for MSISDN <MSISDN>
126	Too many shapes in zone <Zone name> for MSISDN <MSISDN>
127	<parameter> <scenario> is not valid. Valid scenarios: <scenario id 1>:<scenario name 1>|<scenario id 2>:<scenario name 2>|... where: • <parameter> is either SCENARIO_ID or SCENARIO_NAME. • <scenario name *n*> may be blank if the scenario does not have a name.
	<parameter> <scenario> is not valid. No scenarios defined. Occurs when a scenario is specified and none are defined. <parameter> is either SCENARIO_ID or SCENARIO_NAME.
	A scenario is required. Valid scenarios: <scenario id 1>:<scenario name 1>|<scenario id 2>:<scenario name 2>|... Occurs when both SCENARIO_ID and SCENARIO_NAME are not specified and a scenario is required
128	SCENARIO_ID and SCENARIO_NAME should not be specified together
129	VOUCHER <voucher> is not unique: also specify PROVIDER
130	The VALUE of TAG <tag> creates too many fields
131	Unable to merge wallets from different service providers
132	The TRACKING_DOMAIN <id> does not exist
133	The TRACKING_DOMAIN <id> is not a VWS
134	The CHARGING_DOMAIN <id> and TRACKING_DOMAIN <id> combination is not permitted **Note:** When the CHARGING_DOMAIN is a VWS domain, then the same id must be used for the TRACKING_DOMAIN.
135	The MSISDN is not a registered SRM subscriber
136	The PRODUCT OR PROVIDER is not valid or no service bundles exist
137	The PRODUCT OR PROVIDER is not valid <product type / service provider>
138	The LIST_TYPE is not recognised, <list type value>
139	The Friend limit has been met or exceeded. <specific error message>
140	Error retrieving status, <specific error message>
141	Error retrieving friends, <specific error message>

Code	Message
142	`No rewards`
143	`No balances`
144	`Balances exceed all rewards`
145	`No tag data in profile`
146	`PROVIDER ID <provider> does not exist`
147	`Too many prefixes`
148	`MSISDN is subscribed to a CUG. Could not update CLI, account number or profile for MSISDN` *`number`* Where *`number`* is the subscriber's msisdn number. You cannot change the MSISDN for a subscriber who is part of a closed user group.
149	`Wallet not rechargeable`
150	`PROMOTION <promotion> is not a defined promotion`
151	`MSISDN <cli> is not subscribed to PROMOTION <promotion>`
152	`MSISDN <cli> has no subscribed promotions`

Glossary of Terms

ACS

Advanced Control Services configuration platform.

ANI

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

AS

Application Server. The logical entity serving a SUA routing key. An AS is equivalent to an SS7 end point (for example, HLR, MSC,…). An AS contains, at least, one ASP.

CC

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

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.

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”.

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.

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.

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).

ISDN

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

ITU

International Telecommunication Union

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.

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

PI

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

PIN

Personal Identification Number

PLMN

Public Land Mobile Network

SCP

Service Control Point. Also known as SLC.

SGSN

Serving GPRS Support Node

SLC

Service Logic Controller (formerly UAS).

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.

SSP

Service Switching Point

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.

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