

Oracle® Communications Network Charging and Control

CCS Provisioning Interface Commands



Release 15.0.0

October 2023

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

Copyright

Copyright © 2023, Oracle and/or its affiliates.

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

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

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

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

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

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

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

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

Contents

About This Document	v
Document Conventions	vi
Chapter 1	
PI Commands Overview	1
Overview	1
Command List	1
About Service Provider Restrictions	2
Parameter Formats	3
Chapter 2	
PI Subscriber Package	7
Overview	7
Add a Subscriber	7
Change Subscriber Details	11
Delete a Subscriber	12
Query a Subscriber	12
Change Call Barring Details	13
Change Credit Card details	14
Change Credit Card Status	15
Add an F&F/F&D Number	16
Change F&F/F&D Numbers	17
Delete an F&F/F&D Number	18
Change Subscriber Profile	19
Query Subscriber Profile	20
Add a CLI to a Closed User Group	21
Delete a CLI from a Closed User Group	23
Create a New Closed User Group	23
Modify a Closed User Group	24
Delete a Closed User Group	25
Query a Closed User Group	26
Perform a BPL	26
Change Subscriber Searchable Fields	28
Query Subscriber Searchable Fields	29
Chapter 3	
PI Wallet Package	31
Overview	31
Add and Link a Wallet	31
Change Wallet Details	35
Delete a Wallet and Balance	37
Query Wallet Details	39
Query Wallet Reservation	41
Delete Wallet Reservation	42
Change Barred Numbers with EDR	43
Add an F&F/F&D Number with EDR	44
Change F&F/F&D Numbers with EDR	45
Merge Subscriber Wallets	47
Query Wallet Users	48

Recharge an Account/Bucket.....	49
Credit Transfer.....	53
Query EDRs for Subscriber Account Details.....	54
Change Credit Card Recharge Request Status	55
Query Credit Card Recharge Requests	56
Add Service Charge	57
Query Service Charge.....	58
Add Charge for Call.....	59
Query Subscribed Periodic Charges	61

Chapter 4

PI Voucher Package..... 63

Overview.....	63
Voucher Recharge	63
Change Voucher Status	69
Change Voucher Details	70
Set Voucher Expiry.....	71
Freeze Voucher	72
Query a Recharge Voucher.....	73
Create On Demand Voucher.....	74
Query a Serial Range.....	76
Update a Serial Range	77

Chapter 5

PI SNST Package 79

Overview.....	79
Add Friends	79
Remove a Friend	80
Query SNST Information	81
Register for SNST	83
Block SNST Service	84

Chapter 6

Error Code Lists..... 87

Overview.....	87
PI Chassis Errors.....	87
PI Command Errors.....	88

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 Network Charging and Control 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:

- *Provisioning Interface User's and Technical Guide*
- *Virtual Private Network User's Guide*

Document Conventions

Typographical Conventions

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

Formatting Convention	Type of Information
Special Bold	Items you must select, such as names of tabs. Names of database tables and fields.
<i>Italics</i>	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.
<i>variable</i>	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.

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	1
About Service Provider Restrictions	2
Parameter Formats.....	3

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 run 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		

Function	piSubscriberSms	piWalletSms	piVoucherSms
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 a CCS BPL	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
Change voucher details			CCSVR1=CHG
Setting voucher expiry			CCSVR1=CHG
Mark voucher frozen			CCSVR1=FRZ
Query a recharge voucher			CCSVR1=QRY
Create on-demand voucher			CCSVR1=COV

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"
BARCODE	12 digit number
BE_ID	10 digit number (>0)
BPL	50 character string
CARD_DESIGN	4 digit number
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
CUG	50 character string
CURRENCY	3 character string
DATE	YYYYMMDDHHMMSS (24 hour clock)

Parameter	Format
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
DISTRIBUTOR_CHANNEL	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: <KEYn> and <VALUEn> must not contain the comma (','), bar (' ') or semi-colon (;) characters. Additionally, <KEYn> 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
FREE_TEXT_FIELD_1	50 character string
FREE_TEXT_FIELD_2	50 character string
FREE_TEXT_FIELD_3	50 character string
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) 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.

Parameter	Format
MAX_CONCURRENT_ACCE SS	Number (0 to 429467295) Note: Length determined by the length of an unsigned 32 bit number
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
PROVISIONING_DATE	YYYYMMDDHHMMSS (24 hour clock)
RECHARGE_CLASSIFICATI ON	50 character string
RECHARGE_TYPE	"Voucher", "Credit" or "Custom" (not case sensitive)
REFERENCE	See CCSCD3=RCH / CCSCC1
REFUND	"1" or "0"
RETAIL_CHANNEL	50 character string
SCENARIO	"1" to "9"
START_DATE	YYYYMMDDHHMMSS (24 hour clock)

Parameter	Format
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
TEXT	50 character string
TO	20 digit number
TRACKING_DOMAIN	28 digit number
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

PI Subscriber Package

Overview

Introduction

This chapter describes the Oracle Communications Network Charging and Control (NCC) provisioning interface (PI) 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	7
Change Subscriber Details	11
Delete a Subscriber	12
Query a Subscriber	12
Change Call Barring Details	13
Change Credit Card details	14
Change Credit Card Status	15
Add an F&F/F&D Number	16
Change F&F/F&D Numbers	17
Delete an F&F/F&D Number	18
Change Subscriber Profile	19
Query Subscriber Profile	20
Add a CLI to a Closed User Group	21
Delete a CLI from a Closed User Group	23
Create a New Closed User Group	23
Modify a Closed User Group	24
Delete a Closed User Group	25
Query a Closed User Group	26
Perform a BPL	26
Change Subscriber Searchable Fields	28
Query Subscriber Searchable Fields	29

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=" <i>language_name</i> "
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= <i>number</i>
Description:	Used by the operator to link subscriber accounts
Type:	Integer
Optionality:	Optional.
Allowed:	
Default:	
Notes:	
Example:	

PIN

Syntax:	PIN= <i>number</i>
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 88) 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= <i>ptupdatemode</i>	
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.
- For PREFIX, OPREFIX, and LOPREFIX profile tag data types, more than one entry may be added at once. Each value should be separated by a colon (":") character.

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 return 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.
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. Note: You specify how the CCSCD9=QRY command handles empty profile tag name fields by setting the <code>pi.CCSCD9.QRY.suppressEmptyFields</code> parameter in the <code>eserv.config</code> file. By default, the command prints empty profile tag name fields with a null value.

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: <code><pi:MSISDN>123456789 987654321 123123123</pi:MSISDN></code> Specify virtual member ranges to add using the following format: <code><pi:VM_RANGE>44-55 777 888 900-903</pi:VM_RANGE></code> 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 trigger a BPL for a specified subscriber. This allows the PI to trigger a control plan on the SMS to run a particular service scenario.

Note: The BPL is executed only if the security level of the logged-in PI user is greater than 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. The CLI must already exist.

Parameter	Description
BPL	The short name of the BPL 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	The IDP extension. This can be up to 160 characters long.
PROVIDER	Name of the service provider.
WALLET_TYPE	The applicable wallet type. Values are: "Primary" – The PI command sets only the calling party number in the IDP to the MSISDN from the incoming CCSBPL=EXE message. "Secondary" – The PI command sets both the calling party number and the called party number in the IDP to the MSISDN from the incoming CCSBPL=EXE message.

Note: The IDP extension parameter is defined in the BPL record and corresponds to the extension parameter 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.

Note: The PI command passes the WALLET_TYPE information to the content plan through the calledPartyNumber and callingPartyNumber parameters in the IDP.

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= <i>CLI</i>
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= <i>name</i> . . .
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= <i>value</i>
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

|                     |                                                             |
|---------------------|-------------------------------------------------------------|
| <b>Syntax:</b>      | MSISDN=CLI                                                  |
| <b>Description:</b> | The CLI for the subscriber account this command applies to. |
| <b>Type:</b>        | Integer                                                     |

|                     |                                                                                              |
|---------------------|----------------------------------------------------------------------------------------------|
| <b>Optionality:</b> | Mandatory                                                                                    |
| <b>Allowed:</b>     | The CLI must have already been set up as a valid subscriber in the CCS_ACCT_REFERENCE table. |
| <b>Default:</b>     | None                                                                                         |
| <b>Notes:</b>       |                                                                                              |
| <b>Example:</b>     | MSISDN=0123456789                                                                            |

### TAG

|                     |                                                                                             |
|---------------------|---------------------------------------------------------------------------------------------|
| <b>Syntax:</b>      | TAG= <i>name</i>   . . .                                                                    |
| <b>Description:</b> | A predefined profile tag name corresponding to a DISPLAY_NAME in the CCS_SEARCH_KEYS table. |
| <b>Type:</b>        | String                                                                                      |
| <b>Optionality:</b> | Mandatory                                                                                   |
| <b>Allowed:</b>     | The TAG should be a valid key display name as defined in the CCS_SEARCH_KEYS table.         |
| <b>Default:</b>     | None                                                                                        |
| <b>Notes:</b>       | More than one tag may be specified. They should be separated by the pipe ( ) symbol.        |
| <b>Example:</b>     | 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.

# PI Wallet Package

## Overview

### Introduction

This chapter describes the Oracle Communications Network Charging and Control (NCC) 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 and Link a Wallet .....                      | 31 |
| Change Wallet Details .....                      | 35 |
| Delete a Wallet and Balance .....                | 37 |
| Query Wallet Details .....                       | 39 |
| Query Wallet Reservation .....                   | 41 |
| Delete Wallet Reservation .....                  | 42 |
| Change Barred Numbers with EDR .....             | 43 |
| Add an F&F/F&D Number with EDR .....             | 44 |
| Change F&F/F&D Numbers with EDR .....            | 45 |
| Merge Subscriber Wallets .....                   | 47 |
| Query Wallet Users .....                         | 48 |
| Recharge an Account/Bucket .....                 | 49 |
| Credit Transfer .....                            | 53 |
| Query EDRs for Subscriber Account Details .....  | 54 |
| Change Credit Card Recharge Request Status ..... | 55 |
| Query Credit Card Recharge Requests .....        | 56 |
| Add Service Charge .....                         | 57 |
| Query Service Charge .....                       | 58 |
| Add Charge for Call .....                        | 59 |
| Query Subscribed Periodic Charges .....          | 61 |

## Add and 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                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------|
| CHARGING_DOMAIN | The reference ID of the charging domain for this subscriber. Specify a CHARGING_DOMAIN that matches a defined non-VWS domain |
| 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.<br><b>Note:</b> 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:<br>"KEY1=VALUE1   KEY2=VALUE2"<br>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.<br>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: <ul style="list-style-type: none"> <li>• "Debit",</li> <li>• "Single",</li> <li>• "Use",</li> <li>• "Credit"</li> </ul>                    |

| Parameter             | Description                                                                                                                                                                          |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | <ul style="list-style-type: none"> <li>"Limited Credit"</li> </ul>                                                                                                                   |
| 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: <ul style="list-style-type: none"> <li>"Primary"</li> <li>"Secondary"</li> </ul>                                                                   |

## 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.                   |

| Parameter          | Description                                                                                                                                                                                                                             |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LANGUAGE           | The language short name. The language must exist already in the language table.<br>Defaults to the language of the logged PI user if this is defined. Otherwise the system default language is used.                                    |
| 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: <ul style="list-style-type: none"> <li>• A = Active</li> <li>• D = Dormant</li> <li>• F = Frozen</li> <li>• P = Pre-Use</li> <li>• S = Suspended</li> <li>• T = Terminated</li> </ul> |
| 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.<br><b>Note:</b> 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: <ul style="list-style-type: none"> <li>• ABSOLUTE</li> <li>• DELTA (default)</li> </ul>                             |
| 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:<br>"KEY1=VALUE1   KEY2=VALUE2"<br>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: <ul style="list-style-type: none"> <li>• Primary</li> <li>• Secondary</li> </ul> |

## 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 and 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                                                                                                                                                                                       |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DELETE_LINKED_MSISDNS | Sets whether or not to delete any other subscriber accounts that share this wallet. Values are: <ul style="list-style-type: none"> <li>• "Y" – Yes</li> <li>• "N" – No (default value)</li> </ul> |
| EXTRA_EDR             | List of EDR fields to add using the following format:<br>" <i>KEY1=VALUE1   KEY2=VALUE2</i> "<br>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.<br>Valid values are: <ul style="list-style-type: none"> <li>• BALANCE</li> <li>• BALANCE_TYPE</li> <li>• BYPASS_NUMBER</li> <li>• CHARGING_DOMAIN</li> <li>• CREATION_DATE</li> <li>• CUG</li> <li>• CURRENCY</li> <li>• FDN</li> <li>• FFD</li> <li>• FFN</li> <li>• FIRST_ACTIVATION_DATE</li> </ul>                                                         |

| Parameter   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | <ul style="list-style-type: none"> <li>• FREE_SWAPS_REMAINING</li> <li>• INITIAL_BALANCE</li> <li>• LANGUAGE</li> <li>• LAST_RECHARGE_AMOUNT</li> <li>• LAST_CC_RECHARGE_DATE</li> <li>• LAST_ESP_CREDIT</li> <li>• LAST_RECHARGE_DATE</li> <li>• LAST_STATE_CHANGE_DATE</li> <li>• LAST_STATE_CHANGE_REASON</li> <li>• LAST_SWAP_RESET_DATE</li> <li>• LAST_USE_DATE</li> <li>• PREV_BALANCE</li> <li>• PREV_BALANCE_EXPIRY_DATE</li> <li>• PREV_WALLET_EXPIRY_DATE</li> <li>• PRODUCT</li> <li>• SERVICE_PROVIDER</li> <li>• STATUS</li> <li>• TOTAL_EXP_DATE</li> <li>• TRACKING_DOMAIN</li> <li>• UNRESERVED_BALANCE</li> <li>• WALLET_EXPIRY_DATE</li> <li>• WALLET_LIFE_CYCLE_PERIOD</li> <li>• WALLET_REFERENCE</li> <li>• WALLET_TYPE</li> </ul> |
| 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.

## 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.<br>0 = Apply tariff default operation (either commit or revoke)<br>1 = Commit<br>2 = Revoke | Let server decide between commit and revoke based on applicable <b>eserv.config</b> . |
| 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: <ul style="list-style-type: none"> <li>"A" – Allowed</li> <li>"B" – Barred</li> </ul>              |
| IGNORE    | Sets whether or not the call barring list should be ignored. Valid values are: <ul style="list-style-type: none"> <li>"Y" – True</li> <li>"N" – False</li> </ul>               |
| 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".<br>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:<br><code>"KEY1=VALUE1   KEY2=VALUE2"</code><br>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.<br><b>Note:</b> 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.<br><b>Note:</b> 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:<br><i>"KEY1=VALUE1   KEY2=VALUE2"</i><br>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.<br><b>Note:</b> The number specified must not be on the F&F numbers barred list. |

| Parameter | Description                                                                                                                                                                                 |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TYPE      | Sets the type of service. Valid values are: <ul style="list-style-type: none"> <li>• "F" – Friends &amp; Family</li> <li>• "D" – Friends &amp; Destination</li> <li>• "N" – None</li> </ul> |
| FDNUM     | The F&D number to set the Active Discount Prefix. This can be up to 20 digits long.<br><b>Note:</b> 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:<br><code>"KEY1=VALUE1   KEY2=VALUE2"</code><br>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: <ul style="list-style-type: none"> <li>• Primary</li> <li>• Secondary</li> </ul> |
| WALLET_TYPE2 | The name of the wallet type for the MSISDN2 parameter. Values are: <ul style="list-style-type: none"> <li>• Primary</li> <li>• Secondary</li> </ul> |
| EXTRA_EDR    | List of EDR fields to add using the following format:<br><i>"KEY1=VALUE1   KEY2=VALUE2"</i><br>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: <ul style="list-style-type: none"> <li>• Credit</li> <li>• Custom</li> </ul>                                                                                                           |
| REFERENCE     | Holds either the voucher number or the recharge requests ID.<br>For credit type recharges this will be the reference id of the pending recharges.<br>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

|                     |                                                                                                    |
|---------------------|----------------------------------------------------------------------------------------------------|
| <b>Optionality:</b> | Optional                                                                                           |
| <b>Allowed:</b>     |                                                                                                    |
| <b>Default:</b>     |                                                                                                    |
| <b>Notes:</b>       | The expiry date for the <b>Free SMS</b> balance type cannot be set with the CCSCD3=RCH PI command. |
| <b>Example:</b>     | BALANCE_EXPIRY=7                                                                                   |

### WALLET\_EXPIRY

|                     |                                                                 |
|---------------------|-----------------------------------------------------------------|
| <b>Syntax:</b>      | WALLET_EXPIRY= <i>days</i>                                      |
| <b>Description:</b> | The number of days from today's date before the wallet expires. |
| <b>Type:</b>        | Integer                                                         |
| <b>Optionality:</b> | Optional                                                        |
| <b>Allowed:</b>     |                                                                 |
| <b>Default:</b>     |                                                                 |
| <b>Notes:</b>       |                                                                 |
| <b>Example:</b>     | WALLET_EXPIRY=7                                                 |

### MODE

|                     |                                                                                                                                                                                                                                        |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax:</b>      | MODE= <i>int</i>                                                                                                                                                                                                                       |
| <b>Description:</b> | Determines which recharge amounts are used.                                                                                                                                                                                            |
| <b>Type:</b>        | Integer                                                                                                                                                                                                                                |
| <b>Optionality:</b> | Optional                                                                                                                                                                                                                               |
| <b>Allowed:</b>     | <ul style="list-style-type: none"> <li>1 Use provided values.</li> <li>2 Use profile values.</li> <li>3 Use whichever is greater, provided values or profile values.</li> <li>4 Extend current values by specified amounts.</li> </ul> |
| <b>Default:</b>     | 1                                                                                                                                                                                                                                      |
| <b>Notes:</b>       |                                                                                                                                                                                                                                        |
| <b>Example:</b>     | MODE=3                                                                                                                                                                                                                                 |

### WALLET\_TYPE

|                     |                                                              |
|---------------------|--------------------------------------------------------------|
| <b>Syntax:</b>      | WALLET_TYPE= <i>str</i>                                      |
| <b>Description:</b> | The name of the wallet which is the focus of this command.   |
| <b>Type:</b>        | String                                                       |
| <b>Optionality:</b> | Optional                                                     |
| <b>Allowed:</b>     | Primary      Primary wallet<br>Secondary    Secondary wallet |
| <b>Default:</b>     |                                                              |
| <b>Notes:</b>       |                                                              |
| <b>Example:</b>     | WALLET_TYPE=Primary                                          |

### BALANCE\_TYPE

|                     |                                          |
|---------------------|------------------------------------------|
| <b>Syntax:</b>      | BALANCE_TYPE= <i>str</i>                 |
| <b>Description:</b> | The name of the balance to be recharged. |
| <b>Type:</b>        | 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 value 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: <ul style="list-style-type: none"> <li>Primary</li> <li>Secondary</li> </ul> |
| DEST_WALLET | The name of the wallet type transfer credit to. Values are: <ul style="list-style-type: none"> <li>Primary</li> <li>Secondary</li> </ul>      |
| 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.

| Parameter   | Description                                                                                                                                                                                                                                           |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WALLET_TYPE | The name of the wallet to query. Values are: <ul style="list-style-type: none"><li>• Primary</li><li>• Secondary</li></ul>                                                                                                                            |
| EDR_TYPE    | Pip ( ") separated list of EDR types to return.                                                                                                                                                                                                       |
| MAX_RECORDS | Sets the maximum number of EDRs to return (default = 5). If specified, this must be greater than 0.<br><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. |
| DAYS        | 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.                                                                                                      |
| START_DATE  | Sets the date for the earliest EDR to return.<br>This parameter will be ignored if DAYS is specified.                                                                                                                                                 |
| END_DATE    | Sets the date for the latest EDR to return.<br>This parameter will be ignored if DAYS is specified.                                                                                                                                                   |

| Parameter    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TAG<br>VALUE | <p>The parameters, TAG and VALUE work together.</p> <p>When set, returns records containing only the values mentioned in the TAG and VALUE. You can have multiple entries for TAG and VALUE.</p> <p><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.</p> <p>For example:</p> <p><b>CCSCD7=QRY:MSISDN=1234,TAG=EVENT_CLASS EVENT_CLASS,VALUE=SMS DATA,EDR_TYPE=5,MAX_RECORDS=100000</b></p> <p><b>Result:</b> Displays the records containing both the <b>EVENT_CLASS=SMS</b> and <b>EVENT_CLASS=DATA</b></p> |

## 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 *NCC 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. |

| Parameter | Description                                                                                                                                                                                        |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STATUS    | Sets the state of the recharge request. Must be set to one of the following: <ul style="list-style-type: none"> <li>• "P" - pending</li> <li>• "Y" - verified</li> <li>• "N" - invalid.</li> </ul> |

### 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: <ul style="list-style-type: none"> <li>• "P" – Pending</li> <li>• "Y" – Verified</li> <li>• "N" – Invalid</li> </ul> |
| 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 CCSCC1 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: <ul style="list-style-type: none"> <li>1 – Refund</li> <li>0 – Charge (default)</li> </ul>      |
| WALLET_TYPE  | The name of the wallet to be charged. Values are: <ul style="list-style-type: none"> <li>Primary</li> <li>Secondary</li> </ul>                                                  |
| BALANCE_TYPE | The name of the balance to be charged.                                                                                                                                          |
| EXTRA_EDR    | List of EDR fields to add using the following format:<br>" <i>KEY1=VALUE1   KEY2=VALUE2</i> "<br>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: <ul style="list-style-type: none"> <li>1 – Refund</li> <li>0 – Not a refund (default)</li> </ul> |
| 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:<br>" <i>KEY1=VALUE1   KEY2=VALUE2</i> "<br>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 `ccsSMDispatcher` 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

|                     |                                                                                              |
|---------------------|----------------------------------------------------------------------------------------------|
| <b>Syntax:</b>      | MSISDN= <i>CLI</i>                                                                           |
| <b>Description:</b> | The CLI for the subscriber account this command applies to.                                  |
| <b>Type:</b>        | Integer                                                                                      |
| <b>Optionality:</b> | Mandatory                                                                                    |
| <b>Allowed:</b>     | The CLI must have already been set up as a valid subscriber in the CCS_ACCT_REFERENCE table. |
| <b>Default:</b>     | None                                                                                         |
| <b>Notes:</b>       |                                                                                              |
| <b>Example:</b>     | MSISDN=0123456789                                                                            |

### Constraint required parameters

There are no constraint required parameters for this command.

### Optional parameters

CCSSC3 accepts the following optional parameters.

CHARGE

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

WALLET\_TYPE

|                     |                                                            |
|---------------------|------------------------------------------------------------|
| <b>Syntax:</b>      | WALLET_TYPE= <i>str</i>                                    |
| <b>Description:</b> | The name of the wallet which is the focus of this command. |

|              |                                                              |
|--------------|--------------------------------------------------------------|
| Type:        | String                                                       |
| Optionality: | Optional                                                     |
| Allowed:     | Primary      Primary wallet<br>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;
```

# PI Voucher Package

## Overview

### Introduction

This chapter describes the Oracle Communications Network Charging and Control (NCC) 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.....          | 63 |
| Change Voucher Status .....    | 69 |
| Change Voucher Details .....   | 70 |
| Set Voucher Expiry.....        | 71 |
| Freeze Voucher .....           | 72 |
| Query a Recharge Voucher.....  | 73 |
| Create On Demand Voucher ..... | 74 |
| Query a Serial Range .....     | 76 |
| Update a Serial Range .....    | 77 |

## 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: <ul style="list-style-type: none"> <li>• Voucher</li> <li>• VoucherType</li> <li>• Credit</li> <li>• Custom</li> </ul>                                                                                                                                                                                           |
| REFERENCE     | Holds either the voucher number or the recharge requests id.<br>For voucher recharges this will be the voucher number.<br>For VoucherType recharges this will be the voucher type name.<br>For credit recharges this will be the reference id of the pending recharges.<br>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

|                     |                                                                  |
|---------------------|------------------------------------------------------------------|
| <b>Syntax:</b>      | BALANCE_EXPIRY= <i>days</i>                                      |
| <b>Description:</b> | The number of days from today's date before the balance expires. |
| <b>Type:</b>        | Integer                                                          |
| <b>Optionality:</b> | Optional                                                         |
| <b>Allowed:</b>     |                                                                  |
| <b>Default:</b>     |                                                                  |
| <b>Notes:</b>       |                                                                  |
| <b>Example:</b>     | BALANCE_EXPIRY=7                                                 |

### WALLET\_EXPIRY

|                     |                                                                 |
|---------------------|-----------------------------------------------------------------|
| <b>Syntax:</b>      | WALLET_EXPIRY= <i>days</i>                                      |
| <b>Description:</b> | The number of days from today's date before the wallet expires. |
| <b>Type:</b>        | Integer                                                         |
| <b>Optionality:</b> | Optional                                                        |
| <b>Allowed:</b>     |                                                                 |
| <b>Default:</b>     |                                                                 |

**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**

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax:</b>      | SCENARIO_NAME= <i>name</i>   <i>name</i> ...]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Description:</b> | The name of the Scenario to apply in the voucher recharge.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Type:</b>        | String                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Optionality:</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Allowed:</b>     | Any valid scenario name up to 50 characters long.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Default:</b>     | Whatever pi.CCSCD3.RCH.defaultScenarioName entry in the <b>eserv.config</b> file is set to. (It defaults to "Default").                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Notes:</b>       | <p>Cannot be used if SCENARIO_ID is set.</p> <p>This parameter is valid only for Voucher recharges and will be ignored for VoucherType recharges.</p> <p>Scenario must be defined for the voucher. Scenario names are configured on the New and Edit Voucher Type screen. For more information, see <i>CCS User's Guide</i>.</p> <p>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 <b>eserv.config</b> file.</p> |
| <b>Example:</b>     | 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.

### Voucher and voucher type 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 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: <ul style="list-style-type: none"> <li>• "A" – Active</li> <li>• "F" – Frozen</li> </ul> |

### 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.                                                                                                                                          |
| DECRYPT_PRIV<br>ATE_SECRET | If this parameter set to Y, decrypts the voucher private secret value and reports the original HRN.<br>If set to N (the default), does not decrypt the voucher private secret. |

### 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.

## Change Voucher Details

### Description

Use the command CCSVR1 and action CHG to change the details associated with an existing voucher. The following voucher details can be modified:

- Voucher barcode
- Voucher description
- Voucher additional description
- Voucher card design
- Voucher distributor channel details
- Voucher retailer channel details

Example:

```
CCSVR1=CHG:PROVIDER=Boss,SERIAL=114,FREE_TEXT_FIELD_1=FreeTextField1,FREE_TEXT_FIELD_2=FreeTextField2,FREE_TEXT_FIELD_3=FreeTextField3,EXPIRY=20081006090000
```

### Required parameters

This table describes each 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. |

### Constraint required parameters

This table describes each constraint required parameter.

| Field               | Description                                                              |
|---------------------|--------------------------------------------------------------------------|
| BARCODE             | The barcode number of the voucher. You can use a maximum of 12 digits.   |
| TEXT                | Description of the voucher. You can use a maximum of 50 characters.      |
| CARD_DESIGN         | Version number of the voucher design. You can use a maximum of 4 digits. |
| DISTRIBUTOR_CHANNEL | Distributor of the voucher. You can use a maximum of 50 characters.      |
| RETAIL_CHANNEL      | Retailer of the voucher. You can use a maximum of 50 characters.         |

| Field             | Description                                                                               |
|-------------------|-------------------------------------------------------------------------------------------|
| FREE_TEXT_FIELD_1 | Field for capturing business process information. You can use a maximum of 50 characters. |
| FREE_TEXT_FIELD_2 | Field for capturing business process information. You can use a maximum of 50 characters. |
| FREE_TEXT_FIELD_3 | Field for capturing business process information. You can use a maximum of 50 characters. |

### Optional parameters

No optional parameters are supported.

### 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 If the VOUCHER is not unique, then you must specify the PROVIDER.

## Set Voucher Expiry

### Description

Use the command CCSVR1 and action CHG to set, modify, or remove the voucher expiry date.

Example 1: Setting the expiry date to blank (or NULL)

```
CCSVR1=CHG:PROVIDER=Boss,SERIAL=114,EXPIRY=<>
```

Example 2: Setting the expiry date to a valid date (YYYYMMDDHHMMSS)

```
CCSVR1=CHG:PROVIDER=Boss,VOUCHER=114,FREE_TEXT_FIELD_1=FreeTextField1,FREE_T
EXT_FIELD_2=FreeTextField2,FREE_TEXT_FIELD_3=FreeTextField3,EXPIRY=200810060
90000
```

### Required parameters

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

| Field    | Description                           |
|----------|---------------------------------------|
| PROVIDER | The service provider name.            |
| SERIAL   | The serial number for the voucher.    |
| VOUCHER  | The reference number for the voucher. |

## Constraint required parameters

This table describes each constraint required parameter.

| Field  | Description                                                                                                                                                                                                                                      |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EXPIRY | Sets the voucher expiry date using this format: YYYYMMDDHHMMSS. If it is set to blank, then the voucher will not expire.<br><br><b>Note:</b> Setting a date will either add the voucher expiry date or replace the existing voucher expiry date. |

## Optional parameters

No optional parameters are supported.

## Logic and constraints

Here are the rules that apply when using the CCSVR1=CHG command on a VWS domain to change the expiry 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 expiry date.
- 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.                                                                                                                                          |
| DECRYPT_PRIV<br>ATE_SECRET | If this parameter set to Y, decrypts the voucher private secret value and reports the original HRN.<br>If set to N (the default), does not decrypt the voucher private secret. |

## 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.
- Voucher barcode
- Voucher description
- Voucher additional description
- Voucher card design
- Voucher distributor channel details
- Voucher retail channel details
- Provisioning date
- Recharge classification

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.                                                                                                                                          |
| DECRYPT_PRIVATE_SECRET | If this parameter set to Y, decrypts the voucher private secret value and reports the original HRN.<br>If set to N (the default), does not decrypt the voucher private secret. |

## 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 `PROVIDER` is not specified, a value is defaulted. The system checks for the `eserv.config` parameter `pi.CCSVR1.acsCustomerId`, which is an integer service provider id. If this is defined and if the value exists in the database, then the corresponding service provider name is used for the defaulted `PROVIDER` value. If the parameter is either not defined or the defined value does not exist in the database, then an error message is returned.
- 4 If you used the voucher number, the voucher number should not have the PIN appended.

## Create On Demand Voucher

### Description

Use the command `CCSVR1` and action `COV` to create one voucher.

Example: `CCSVR1=COV:PROVIDER=Boss`

The following voucher details are returned by default on successful creation of vouchers:

- Voucher HRN
- Voucher Pre-Use Expiry
- Voucher Serial Number

To stop specific fields from being returned, such as the voucher Pre-Use Expiry, specify the list of fields to suppress in the **suppressFields** parameter in the PI section of **eserv.config**. See *Provisioning Interface User's and Technical Guide* for more information.

## Required parameters

There are no required parameters for this command.

## Constraint required parameters

There are no constraint required parameters for this command.

## Optional parameters

This table describes each optional parameters.

| Parameter        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PROVIDER         | Service Provider Name (ACS customer)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| VOUCHER_TYPE     | Voucher Type name with which the voucher needs to be generated.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| AUTH_MODULE_NAME | Authentication Module/Encryption mechanism to be used for generating the voucher.<br><br><b>Note:</b> It is recommended that if one of the parameters AUTH_MODULE_NAME or AUTH_RULE_NAME is specified in the request, then the other parameter should also be specified. This is to avoid using the default value which may not match up to the pair.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| AUTH_RULE_NAME   | Authentication Rule name to be used for generating the voucher.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| INITIAL_STATE    | The state in which voucher created in this request will have, when it is first generated.<br>Valid values are: <ul style="list-style-type: none"> <li>• <b>C</b> Generate vouchers with state = Created</li> <li>• <b>A</b> Generate vouchers with state = Activated</li> <li>• <b>F</b> Generate vouchers with state = Frozen</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| DOMAIN_NAME      | The name of the domain or billing engine under which on-demand voucher needs to be generated.<br><br><b>Note:</b> Co-relation between request parameter DOMAIN_NAME and <b>eserv.config</b> parameter DEFAULT_DOMAIN_NAME. <ul style="list-style-type: none"> <li>• DOMAIN_NAME has precedence over DEFAULT_DOMAIN_NAME.</li> <li>• If DOMAIN_NAME is supplied in CCSVR1=COV request, then DEFAULT_DOMAIN_NAME is not used.</li> <li>• If DOMAIN_NAME is not supplied in CCSVR1=COV request, then DEFAULT_DOMAIN_NAME is used.</li> <li>• If DOMAIN_NAME is not supplied in CCSVR1=COV request, and the DEFAULT_DOMAIN_NAME is not specified in <b>eserv.config</b>, then: <ul style="list-style-type: none"> <li>▪ If there is only one domain in set-up: The only available domain is used in voucher generation.</li> <li>▪ If there are multiple domains in set-up: This is an error scenario. You must specify domain either in request, or define a default value in <b>eserv.config</b> file.</li> </ul> </li> </ul> |

## Logic and constraints

Here are the rules that apply when using the CCSVR1=COV command to create a voucher.

- 1 All the request parameters are optional, however the defaults must be defined in SMS `eserv.config` CCSVR1 > COV section.
- 2 When the optional parameters are not provided in the request, it is taken from the defaults defined in the `eserv.config` file.

## Query a Serial Range

### Description

Use the command CCSVR1 and action QRY to query a range of voucher serial numbers.

This table describes the list of the status returned by the query.

| Status | Description                             |
|--------|-----------------------------------------|
| C      | CREATED                                 |
| A      | ACTIVE                                  |
| F      | FROZEN                                  |
| U      | USED (Redeemed)                         |
| E      | ERROR (Example: Voucher does not exist) |

Example: CCSVR1=QRY:PROVIDER=Boss,SERIAL\_START=10019,SERIAL\_END=10023

This query returns the status of the SERIALs. They are delimited with colon (:) and separated with pipe (|) as shown below:

```
CCSVR1=QRY:ACK:VOUCHERS=10019:E|10020:E|10021:E|10022:E|10023:E,STATUS=FAILURE,TRANSACTION_FILE=/IN/service_packages/CCS/voucher/export/CCSVR1-QUERY-FAILURES.20221020133915.txt
```

TRANSACTION\_FILE contains the details of errors for each of the serials if they encountered any.

### Required parameters

There are no required parameters for this command.

### Constraint required parameters

| Status       | Description                            |
|--------------|----------------------------------------|
| SERIAL_START | This is the start of the serial range. |
| SERIAL_END   | This is the end of the serial range.   |

## Optional parameters

No optional parameter supported.

## Logic and constraints

- 1 The value of SERIAL\_START should be less than SERIAL\_END.
- 2 If PROVIDER is not specified, a value is defaulted. The system checks for the `eserv.config` parameter `pi.CCSVR1.acsCustomerId`, which is an integer service provider id. If this is defined and if the value exists in the database, then the corresponding service provider name is used for the defaulted PROVIDER value. If the parameter is either not defined or the defined value does not exist in the database, then an error message is returned.
- 3 Maximum length of the range supported is 2000 serials.

## Update a Serial Range

### Description

Use the command CCSVR1 and action CHG to update a range of voucher serial numbers.

Example: `CCSVR1=QRY:PROVIDER=Boss,SERIAL_START=10019,SERIAL_END=10023`

This query returns the status of the SERIALs. They are delimited with colon (:) and separated with pipe (|) as shown below:

```
CCSVR1=QRY:ACK:VOUCHERS=10019:E|10020:E|10021:E|10022:E|10023:E,STATUS=FAILURE,TRANSACTION_FILE=/IN/service_packages/CCS/voucher/export/CCSVR1-QUERY-FAILURES.20221020133915.txt
```

Example: `CCSVR1=CHG:PROVIDER=Boss,SERIAL_START=10019,SERIAL_END=10023,STATUS=A`

```
CCSVR1=QRY:ACK:STATUS=PARTIAL,TRANSACTION_FILE=/IN/service_packages/CCS/voucher/export/CCSVR1-STATUS-UPDATE-FAILURES.20221020143518.txt
```

OR

```
CCSVR1=CHG:PROVIDER=Boss,SERIAL_START=10019,SERIAL_END=10023,EXPIRY=20230530113455
```

```
CCSVR1=QRY:ACK:STATUS=PARTIAL,TRANSACTION_FILE=/IN/service_packages/CCS/voucher/export/CCSVR1-EXPIRY-UPDATE-FAILURES.20221020143518.txt
```

TRANSACTION\_FILE contains the details of errors for each of the serials if they encountered any.

### Required parameters

There are no required parameters for this command.

### Constraint required parameters

| Parameter    | Description                                  |
|--------------|----------------------------------------------|
| SERIAL_START | This is the start of the serial range.       |
| SERIAL_END   | This is the end of the serial range.         |
| STATUS       | If STATUS needs to be updated for the range. |
| EXPIRY       | If EXPIRY needs to be updated for the range. |

## Optional parameters

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

## Logic and constraints

- 1 The value of SERIAL\_START should be less than SERIAL\_END.
- 2 If PROVIDER is not specified, a value is defaulted. The system checks for the *eserv.config* parameter **pi.CCSVR1.acsCustomerId**, which is an integer service provider id. If this is defined and if the value exists in the database, then the corresponding service provider name is used for the defaulted PROVIDER value. If the parameter is either not defined or the defined value does not exist in the database, then an error message is returned.
- 3 Maximum length of the range supported is 2000 serials.
- 4 Updating the Status/Expiry of the **Frozen** or **Redeemed** serials are not allowed.

**Note:** Status returned in the response of the QRY and CHG operation performed on a range of serials is as follows:

- **SUCCESS:** If no error are encountered.
- **FAILURE:** If the processing was not started due to any communication error, or parameter validation failure, or none of the serials were valid.
- **PARTIAL:** If there are errors for some of the vouchers, like 'voucher not found in database', or a database connection error, a transaction file is created with the details about the errors.

# PI SNST Package

## Overview

### Introduction

This chapter describes the Oracle Communications Network Charging and Control (NCC) 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 .....            | 79 |
| Remove a Friend .....        | 80 |
| Query SNST Information ..... | 81 |
| Register for SNST .....      | 83 |
| Block SNST Service .....     | 84 |

## 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`

|                     |                                                                                                          |
|---------------------|----------------------------------------------------------------------------------------------------------|
| <b>Syntax:</b>      | <code>MSISDN=CLI</code>                                                                                  |
| <b>Description:</b> | The CLI for the subscriber account this command applies to.                                              |
| <b>Type:</b>        | Integer                                                                                                  |
| <b>Optionality:</b> | Required except when using the <code>CCSSR1=QRY</code> command.                                          |
| <b>Allowed:</b>     | The CLI must have already been set up as a valid SNST user in the <code>CCS_ACCT_REFERENCE</code> table. |
| <b>Default:</b>     | 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

|                     |                                                                                             |
|---------------------|---------------------------------------------------------------------------------------------|
| <b>Syntax:</b>      | MSISDN= <i>CLI</i>                                                                          |
| <b>Description:</b> | The CLI for the subscriber account this command applies to.                                 |
| <b>Type:</b>        | Integer                                                                                     |
| <b>Optionality:</b> | Required except when using the CCSSR1=QRY command.                                          |
| <b>Allowed:</b>     | The CLI must have already been set up as a valid SNST user in the CCS_ACCT_REFERENCE table. |
| <b>Default:</b>     | None                                                                                        |
| <b>Notes:</b>       |                                                                                             |
| <b>Example:</b>     | MSISDN=0123456789                                                                           |

#### FRIEND

|                     |                                                                |
|---------------------|----------------------------------------------------------------|
| <b>Syntax:</b>      | FRIEND= <i>CLI</i>                                             |
| <b>Description:</b> | The CLI of the friend to be removed from the SNST friend list. |
| <b>Type:</b>        | Integer                                                        |
| <b>Optionality:</b> | Required.                                                      |
| <b>Allowed:</b>     | The friend must be a valid SNST friend for the subscriber.     |
| <b>Default:</b>     | None                                                           |
| <b>Notes:</b>       |                                                                |
| <b>Example:</b>     | 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

|                     |                                                                                  |                                                          |
|---------------------|----------------------------------------------------------------------------------|----------------------------------------------------------|
| <b>Syntax:</b>      | LIST_TYPE=( <i>parameter1</i>   <i>parameter2</i>  .. <i>parameterN</i> >)       |                                                          |
| <b>Description:</b> | Pipe " " separated list of items to return.                                      |                                                          |
| <b>Type:</b>        | String                                                                           |                                                          |
| <b>Optionality:</b> | Required.                                                                        |                                                          |
| <b>Allowed:</b>     | 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 |
| <b>Default:</b>     |                                                                                  |                                                          |
| <b>Notes:</b>       |                                                                                  |                                                          |
| <b>Example:</b>     | LIST_TYPE=(FRIENDS SIZE LIMIT TARIFF DISCOUNT SERVICE_BUNDLES STATUS PROMOTIONS) |                                                          |

## Constraint required parameters

CCSSR1=QRY accepts the following constraint required parameters.

### MSISDN

|                     |                                                                                             |
|---------------------|---------------------------------------------------------------------------------------------|
| <b>Syntax:</b>      | MSISDN= <i>CLI</i>                                                                          |
| <b>Description:</b> | The CLI for the subscriber account this command applies to.                                 |
| <b>Type:</b>        | Integer                                                                                     |
| <b>Optionality:</b> | Required except when using the CCSSR1=QRY command.                                          |
| <b>Allowed:</b>     | The CLI must have already been set up as a valid SNST user in the CCS_ACCT_REFERENCE table. |
| <b>Default:</b>     | None                                                                                        |
| <b>Notes:</b>       |                                                                                             |
| <b>Example:</b>     | MSISDN=0123456789                                                                           |

### PROVIDER

|                     |                                           |
|---------------------|-------------------------------------------|
| <b>Syntax:</b>      | PROVIDER=" <i>service_provider_name</i> " |
| <b>Description:</b> | The name of the service provider.         |
| <b>Type:</b>        | String.                                   |
| <b>Optionality:</b> | 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 they 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.



# Error Code Lists

## Overview

### Introduction

This chapter explains the error codes for Oracle Communications Network Charging and Control (NCC) provisioning interface (PI) commands.

### In this chapter

---

This chapter contains the following topics.

|                        |    |
|------------------------|----|
| PI Chassis Errors..... | 87 |
| PI Command Errors..... | 88 |

## 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: <ul style="list-style-type: none"> <li>• Did not send one</li> <li>• It was invalid</li> </ul> |
| 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 is invalid                                                                                   |
| 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.<br><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><br>where <reason> can be: <ul style="list-style-type: none"> <li>• bad LW_MSISDN &lt;cli&gt;</li> <li>• bad LW_ACCOUNT_NUMBER &lt;number&gt;</li> <li>• LW_MSISDN &lt;cli&gt; has no WALLET_TYPE &lt;type&gt;</li> <li>• LW_ACCOUNT_NUMBER &lt;number&gt; has no WALLET_TYPE &lt;type&gt;</li> </ul> |
| 65   | Could not link to charging domain due to <reason><br>where <reason> can be: <ul style="list-style-type: none"> <li>• bad CHARGING_DOMAIN &lt;charging domain&gt;</li> <li>• bad WALLET_REFERENCE &lt;wallet reference&gt;</li> </ul>                                                                                                                   |
| 66   | Invalid operation for subscriber                                                                                                                                                                                                                                                                                                                       |
| 67   | Bad date range <start date> to <end date>                                                                                                                                                                                                                                                                                                              |
| 68   | Badly formatted parameter <parameter>                                                                                                                                                                                                                                                                                                                  |
| 69   | <text><br>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><br>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> ...<br>where: <ul style="list-style-type: none"> <li>• &lt;parameter&gt; is either SCENARIO_ID or SCENARIO_NAME.</li> <li>• &lt;scenario name n&gt; may be blank if the scenario does not have a name.</li> </ul> |
|      | <parameter> <scenario> is not valid. No scenarios defined.<br>Occurs when a scenario is specified and none are defined.<br><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> ...<br>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<br><b>Note:</b> 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 <i>number</i><br>Where <i>number</i> 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                                                                                                                                                                                                           |
| 164    | DECRYPTION NOT ALLOWED                                                                                                                                                                                                                              |
| 165    | Error while checking if ccsVoucher binary already running ( <i>specific to CCSVR1=COV command</i> )                                                                                                                                                 |
| 166    | ccsVoucher already running (failed to execute even after wait & retries). ( <i>specific to CCSVR1=COV command</i> )                                                                                                                                 |
| 167    | Error occurred while executing ccsVoucher binary. Unable to execute the binary. ( <i>specific to CCSVR1=COV command</i> )                                                                                                                           |
| 168    | PROVIDER is not in incoming request, nor configured in eserv.config ( <i>specific to CCSVR1=COV command</i> )                                                                                                                                       |
| 169    | VOUCHER_TYPE is not in incoming request, nor configured in eserv.config ( <i>specific to CCSVR1=COV command</i> )                                                                                                                                   |
| 170    | AUTH_MODULE_NAME is not in incoming request, nor configured in eserv.config ( <i>specific to CCSVR1=COV command</i> )                                                                                                                               |
| 171    | AUTH_RULE_NAME is not in incoming request, nor configured in eserv.config ( <i>specific to CCSVR1=COV command</i> )                                                                                                                                 |
| 172    | INITIAL_STATE is not in incoming request, nor configured in eserv.config ( <i>specific to CCSVR1=COV command</i> )                                                                                                                                  |
| 200    | On demand voucher created successfully ( <i>specific to CCSVR1=COV command</i> )                                                                                                                                                                    |
| 202    | Invalid Domain name (billing engine name) ( <i>specific to CCSVR1=COV command</i> )                                                                                                                                                                 |
| 203    | Invalid AUTH rule name / AUTH module name ( <i>specific to CCSVR1=COV command</i> )                                                                                                                                                                 |
| 204    | Invalid Voucher type ( <i>specific to CCSVR1=COV command</i> )                                                                                                                                                                                      |
| 205    | Other error in voucher creation. ( <i>specific to CCSVR1=COV command</i> )                                                                                                                                                                          |
| 130001 | Default PROVIDER not defined by config item pi.CCSVR1.acsCustomerId                                                                                                                                                                                 |