Oracle® Communications Convergent Charging Controller

CCS Provisioning Interface Commands

Release 15.1

April 2025

Copyright

Contents

Chapter 5

Error Code Lists 43

About This Document

Scope

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

Audience

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

Prerequisites

A solid understanding of UNIX and a familiarity with IN concepts are an essential prerequisite for safely using the information contained in this technical guide.

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

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

Related Documents

The following documents are related to this document:

- *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 Convergent Charging Controller documentation.

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

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

Chapter 1

PI Commands Overview

Overview

Introduction

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

In this chapter

This chapter contains the following topics.

Command List

Command List

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

Example: To 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 subscriber profile	CCSCD9=CHG		
Query subscriber profile	CCSCD9=QRY		
Perform a CCS BPL	CCSBPL=EXE		
Change Subscriber Searchable Fields	CCSSF1=CHG		
Query Subscriber Searchable Fields	CCSSF1=QRY		
Query EDRs for an account		CCSCD7=QRY	
Change voucher status			CCSVR1=CHG

Function	piSubscriberSms	piWalletSms	piVoucherSms
Mark voucher frozen			CCSVR1=FRZ
Query a recharge voucher			CCSVR1=QRY
Create on-demand voucher			CCSVR1=COV
Create voucher batch			CCSVR1=CVB
Check voucher batch status			CCSVR1=VBS

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
BARCODE	12 digit number
BE_ID	10 digit number (>0)
BPL	50 character string
CARD_DESIGN	4 digit number
CHARGING_DOMAIN	28 digit number
CLEAR	“Y” or “N”
CURRENCY	3 character string
DATE	YYYYMMDDHHMMSS (24 hour clock)
DAYS	Integer >=0
DEL	String: “MSISDN1|MSMIDN2|…”

Parameter	Format
DISTRIBUTOR_CHANNEL	50 character string
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.
EXTn (n is 1 to 10)	160 character string
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
FROM	20 digit number
IGNORE	“Y” or “N”
IMSI	15 digit number (>0)
LANGUAGE	50 character string
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.
MAX_RECORDS	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.
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
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_CLASSIFICATION	50 character string
RECHARGE_TYPE	“Voucher”, “Credit” or “Custom” (not case sensitive)
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
TEXT	50 character string
TO	20 digit number
TYPE	“F”, “D” or “N”
VALUE	See CCSCD9=CHG
VOUCHER	20 digit number
WALLET_REFERENCE	50 character string
WALLET_TYPE	50 character string

Chapter 2

PI Subscriber Package

Overview

Introduction

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

These commands are added by the `piSubscriberSms` package.

In this chapter

This chapter contains the following topics.

Add a Subscriber

Description

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

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

Required parameters

This table describes each required parameter.

Parameter	Description
CHARGING_DOMAIN	The reference id of the charging domain for this subscriber.
PROVIDER	Service provider name
PRODUCT	Product type name

Constraint required parameters

This table describes each constraint required parameter.

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

Optional parameters

CCSCD1=ADD accepts the following optional parameters.

LANGUAGE

Syntax: LANGUAGE="*language_name*"

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

Type: String

Optionality: Optional (default used if not set).

Allowed: Up to 50 characters

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

Notes:

Example: LANGUAGE="French"

BYPASS_NUMBER

Syntax: BYPASS_NUMBER= *number*

Description: Used by the operator to link subscriber accounts

Type: Integer

Optionality: Optional.

Allowed:

Default:

Notes:

Example:

PIN

Syntax: PIN=*number*

Description: The subscriber's private PIN.

Type: Integer

Optionality: Optional

Allowed: Any decimal number between 4 and 10 digits long

Default:

Notes:

Example: PIN=4812

ACCOUNT_NUMBER

Syntax:	ACCOUNT_NUMBER=*reference_number*
Description:	The reference number (subscriber id) for the subscriber's account.
Type:	Integer
Optionality:	Optional.
Allowed:	Any number up to 20 digits long
Default:	
Notes:	
Example:	ACCOUNT_NUMBER=1047768832

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 PIN, the last 4 digits of the generated account number will be used for the PIN.

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.

Error codes

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

Refer to *PI Command Errors* (on page 44) 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: • BYPASS_NUMBER • CHARGING_DOMAIN • CREATION_DATE • LANGUAGE • PRODUCT • SERVICE_PROVIDER • 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 third party charging domains, if LIST_TYPE is not specified then only the MSISDN and the ACCOUNT_NUMBER will be returned.
4. 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 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.

Parameter	Description
VALUE	The value for the tag. This must match the tag data type. If more than one tag is specified, tag values must be pipe separated and listed in the same order as for TAG.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

CCSCD9=CHG command accepts the following optional parameters.

PTMODE

Syntax: PTMODE=*ptupdatemode*

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

Type: String

Optionality: Optional.

Allowed:

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

Default:

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

Example: PTMODE=ADD

Profile tag types

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

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

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

Note:

- The limits for LNSTRING and LOPREFIX tag values may be set for the Product Type in the Subscriber Management screen, or in Resource Limits in the Service Management screen.
- 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 `pi.CCSCD9.QRY.suppressEmptyFields` parameter in the **eserv.config** 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.

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.
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=`*`CLI`*
Description:	The CLI for the subscriber account this command applies to.
Type:	Integer
Optionality:	Mandatory
Allowed:	The CLI must have already been set up as a valid subscriber in the CCS_ACCT_REFERENCE table.
Default:	None
Notes:	
Example:	`MSISDN=0123456789`

`TAG`

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

Example:	TAG=adContactNumber|adPostcode

VALUE

Syntax:	VALUE=*value*
Description:	The value for the corresponding tag.
Type:	Plain text (however, must match the tag data type of the destination field).
Optionality:	Mandatory
Allowed:	The VALUE is plain text, but the characters listed below need to be escaped: • To indicate a comma (,), use backslash-c (\c) • To indicate a semi-colon (;), use backslash-s (\s) • To indicate an equals sign (=), use backslash-e (\e) • To indicate a pipe (|), use backslash-pipe (\\|) • To indicate a new-line, use backslash-n (\n) • To indicate a backslash (\), use double backslash (\\) A single backslash will be interpreted as a syntax error. The VALUE is split into fields at every new-line indicator (\n)
Default:	None
Notes:	If more than one tag is specified, then tag values must be pipe separated and listed in the same order as for TAG.
Example:	VALUE=+440123456789|AB123

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

No optional parameters are supported.

Logic and constraints

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

1. If TAG is already defined, its value will be changed to the new value.
2. If VALUE is empty and the TAG is already defined, the entry will be deleted.
3. The number of values must match the number of tags:
   Example: If "Telephone Number" is "+440123456789" and "Postcode" is "AB123".
   TAG=adContactNumber|adPostcode,VALUE=+440123456789|AB123

Query Subscriber Searchable Fields

Description

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

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

Required parameters

CCSSF1 requires the following parameters.

`MSISDN`

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

`TAG`

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

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

No optional parameters are supported.

Logic and constraints

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

1. The MSISDN should be a valid MSISDN defined in the CCS_ACCT_REFERENCE table.
2. The TAG should be a valid key display name defined in the CCS_SEARCH_KEYS table.
3. To specify more than one tag to query, the tag parameter should contain the names of the tags separated by pipe symbols.

Chapter 3

PI Wallet Package

Overview

Introduction

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

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

In this chapter

This chapter contains the following topics.

Add 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. **Note:** START_MSISDN and END_MSISDN must be the same length.
ACCOUNT_NUMBER	The account reference for this subscriber. This is a 20 digit number that is created automatically when a new subscriber is added.
PROVIDER	The service provider name. The service provider must exist already.

Optional parameters

This table describes each optional parameter.

Parameter	Description
BYPASS_NUMBER	A reference number for linking subscriber accounts.
CURRENCY	Sets the currency code. This must be an existing currency code.
LANGUAGE	The language short name. The language must exist already in the language table. Defaults to the language of the logged PI user if this is defined. Otherwise the system default language is used.
LW_ACCOUNT_NUMBER	The subscriber account reference number for a linked wallet.
LW_MSISDN	The CLI for the linked 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.
WALLET_TYPE	The wallet type name. Values are: • "Primary" • "Secondary"

Logic and constraints

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

1. You must either specify the MSISDN to create a single new subscriber and wallet, or the START_MSISDN and END_MSISDN to create a range of new subscribers and wallets. The maximum size for a range of subscriber MSISDNs is 1000.
2. You must specify the PRODUCT and the PROVIDER to create new subscribers (single or range). This must be a valid combination. That is the PROVIDER must be the ACS_CUST_ID for the account type of the specified PRODUCT. They should be the names of the product or provider, rather than the internal ID.

3 You can optionally specify the ACCOUNT_NUMBER when creating a single new subscriber and wallet. You must not specify the ACCOUNT_NUMBER when creating a range of new subscribers and wallets.

4 When creating a new account to link to an existing wallet, MSISDN and PRODUCT must be specified. PROVIDER will be ignored as the service provider of the existing wallet will be used. You can optionally specify the ACCOUNT_NUMBER, but the START_MSISDN and END_MSISDN must not be specified.

5 When creating a new range of accounts to link to an existing wallet, START_MSISDN, END_MSISDN and PRODUCT must be specified. PROVIDER will be ignored as the service provider of the existing wallet will be used. You must not specify the ACCOUNT_NUMBER or MSISDN.

6 When creating a range of subscribers and /or wallets:
   - If WALLET_TYPE is not specified or “Primary”, then START_MSISDN, END_MSISDN and any MSISDN within the range must not already exist in the database.
   - If WALLET_TYPE is “Secondary”, then START_MSISDN, END_MSISDN and any MSISDN within the range must already exist in the database and not already have a “Secondary” wallet.
   - Failure to create any subscriber and/or wallet within the range will result in no subscribers and/or wallets within the range being created.
   - Each subscriber will have the same PROVIDER, PRODUCT, CHARGING_DOMAIN, LANGUAGE, CURRENCY, 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.

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

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
CURRENCY	Sets the currency code. This must be an existing currency code.
LANGUAGE	The language short name. The language must exist already in the language table. Defaults to the language of the logged PI user if this is defined. Otherwise the system default language is used.
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.

Optional parameters

This table describes each optional parameter.

Parameter	Description
WALLET_TYPE	The name of the wallet type. Values are: • Primary • Secondary

Logic and constraints

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

1 You must specify an existing MSISDN and at least one constraint required parameter.
2 The NEW_MSISDN, if specified, should not exist already.

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: • "Y" – Yes • "N" – No (default value)
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 is not specified, the whole account is deleted.
   - If WALLET_TYPE is 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.
3. If DELETE_LINKED_MSISDNS=Y:
   - All wallets are deleted for the MSISDN, and the MSISDN is deleted.
   - WALLET_TYPE is 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
LIST_TYPE	Pipe ("|") separated list of items to return. Valid values are: • BYPASS_NUMBER • CHARGING_DOMAIN • CREATION_DATE • CURRENCY

Parameter	Description
	• LANGUAGE • PRODUCT • SERVICE_PROVIDER • WALLET_REFERENCE • WALLET_TYPE
WALLET_TYPE	The name of the wallet type.

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. 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*.
5. CURRENCY values will always be returned in the wallet currency, even if the balance currency has been set to the system currency.
6. If LIST_TYPE is specified it should contain at least one valid item to return. If it is not specified, all values are returned.
7. All dates are returned in the 24 hour format YYYYMMDDHHMMSS if set or blank if not set.

Change Barred Numbers with EDR

Description

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

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

Required parameters

This table describes each required parameter.

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

Constraint required parameters

This table describes each constraint required parameter.

Parameter	Description
LIST_TYPE	Sets whether this is a list of allowed or barred numbers. Valid values are: • "A" – Allowed • "B" – Barred

Parameter	Description
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.

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.

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: • Primary

Parameter	Description
	• Secondary
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. **Note:** 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. This parameter will be ignored if DAYS is specified.
END_DATE	Sets the date for the latest EDR to return. This parameter will be ignored if DAYS is specified.
TAG VALUE	The parameters, TAG and VALUE work together. When set, returns records containing only the values mentioned in the TAG and VALUE. You can have multiple entries for TAG and VALUE. **Note:** The number of entries in TAG should be equal to the number of entries in the VALUE. If the number of entries are not equal, the command returns an error. For example: **CCSCD7=QRY:MSISDN=1234,TAG=EVENT_CLASS|EVENT_CLASS,VALUE=SMS|DATA, EDR_TYPE=5,MAX_RECORDS=100000** **Result:** Displays the records containing both the **EVENT_CLASS=SMS** and **EVENT_CLASS=DATA**

Logic and constraints

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

1. You must specify an existing MSISDN.
2. If you specify EDR_TYPE, then only EDRs of the specified types will be returned. For details on EDR types, see *Convergent Charging Controller Event Detail Record Reference Guide*.
3. If you specify WALLET_TYPE, only records for that wallet will be returned. Otherwise, records for the default wallet for the subscriber will be returned.
4. If you specify MAX_RECORDS, then only the first MAX_RECORDS records will be returned.
5. If you specify DAYS, then data for the last DAYS days is returned, starting from midnight (00:00:00) local time of the earliest day.
6. If you specify START_DATE, then no EDRs earlier than that date will be returned.
7. If you specify END_DATE, then no EDRs later than that date will be returned.
8. If both START_DATE and END_DATE are specified, END_DATE must be later than START_DATE.
9. START_DATE and END_DATE must be specified in local time.
10. RECORD_DATE will be returned in local time.
11. Any dates in EXTRA_INFORMATION will be returned unaltered from the database record and will be in GMT.

Chapter 4

PI Voucher Package

Overview

Introduction

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

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

In this chapter

This chapter contains the following topics.

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: • "A" – Active • "F" – Frozen

Constraint required parameters

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

Parameter	Description
PROVIDER	The service provider name.
SERIAL	The serial number for the voucher.
VOUCHER	The reference number for the voucher.
DECRYPT_PRIVATE_SECRET	If this parameter set to Y, decrypts the voucher private secret value and reports the original HRN. 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.
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. **Note**: 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_PRIVATE_SECRET	If this parameter set to Y, decrypts the voucher private secret value and reports the original HRN. 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. 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. **Note**: 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. Valid values are: • **C** Generate vouchers with state = Created • **A** Generate vouchers with state = Activated • **F** Generate vouchers with state = Frozen

Parameter	Description
DOMAIN_NAME	The name of the domain or billing engine under which on-demand voucher needs to be generated. **Note**: Co-relation between request parameter DOMAIN_NAME and **eserv.config** parameter DEFAULT_DOMAIN_NAME. • DOMAIN_NAME has precedence over DEFAULT_DOMAIN_NAME. • If DOMAIN_NAME is supplied in CCSVR1=COV request, then DEFAULT_DOMAIN_NAME is not used. • If DOMAIN_NAME is not supplied in CCSVR1=COV request, then DEFAULT_DOMAIN_NAME is used. • If DOMAIN_NAME is not supplied in CCSVR1=COV request, and the DEFAULT_DOMAIN_NAME is not specified in **eserv.config**, then: ▪ If there is only one domain in set-up: The only available domain is used in voucher generation. ▪ 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 **eserv.config** file.

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

CSVR1=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.

Create Voucher Batch

Description

Use the command CCSVR1 and action CVB to create a voucher batch.

Example: CCSVR1=CVB:PROVIDER=Boss:BATCH_SIZE=10

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

- **STATUS**: Status of the triggered voucher batch creation process (ccsVoucher_CCS3). 0 - If triggered successfully. If triggering the voucher batch creation process fails, NACK response with appropriate error message is returned.
- **JOB_ID**: Tracking id for the triggered voucher batch creation process. This is the epoch milli-second timestamp of the Voucher batch progress file. In the voucher batch creation process, progress file is named as following: *ccsVoucherStartupReport.log<epoch milli-seonds timestamp>*
  This JOB_ID can be used in CCSVR1=VBS command to fetch the status of the voucher batch creation process. Using this JOB_ID, appropriate progress file can be fetched, and status can be reported.

Required parameters

This table describes the required parameter.

Parameter	Description
BATCH_SIZE	Number of vouchers to be created in the batch.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

This table describes each optional parameter.

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. **Note**: 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 of voucher created in this request when it is first generated. Valid values are: • **C** Generate vouchers with state = Created • **A** Generate vouchers with state = Activated • **F** Generate vouchers with state = Frozen

Parameter	Description
DOMAIN_NAME	The name of the domain or billing engine under which voucher batch needs to be generated. **Note**: Co-relation between request parameter DOMAIN_NAME and **eserv.config** parameter DEFAULT_DOMAIN_NAME. • DOMAIN_NAME has precedence over DEFAULT_DOMAIN_NAME. • If DOMAIN_NAME is supplied in CCSVR1=CVB request, then DEFAULT_DOMAIN_NAME is not used. • If DOMAIN_NAME is not supplied in CCSVR1=CVB request, then DEFAULT_DOMAIN_NAME is used. • If DOMAIN_NAME is not supplied in CCSVR1=CVB request, and the DEFAULT_DOMAIN_NAME is not specified in **eserv.config**, then: ▪ If there is only one domain in set-up: The only available domain is used in voucher generation. ▪ 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 **eserv.config** file.
START	Start voucher number (This is the start of voucher range)
END	End voucher number (This is the end of voucher range)
BATCH_NAME	Voucher batch name
BATCH_CODE	Voucher batch code
BATCH_EXPIRY	Voucher batch expiry in 'YYYYMMDDHHMMSS' format.
CDR_DESCRIPTION	The voucher batch creation CDR description.

Logic and constraints

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

5 BATCH_SIZE is mandatory. It must be present in the request.
6 Except for START/END, if the optional parameters are not provided in the request, it is taken from the defaults defined in the `eserv.config` file.
7 When START/END are not provided in the request, they are generated by the voucher creation process as per the authentication rule used.
8 For the parameters (BATCH_CODE, BATCH_NAME, CDR_DESCRIPTION) which are not configurable in `eserv.config` and not provided in the SOAP request, empty values are allowed for the same in ccsVoucher binary.
9 For parameter BATCH_EXPIRY, when not specified in the SOAP request, default expiry will be used.

Voucher Batch Status

Description

Use the command CCSVR1 and action VBS to check the status of a voucher batch creation.

Example: CCSVR1=CVB:PROVIDER=Boss:JOB_ID='1729502125319'

The following details are returned:

- **STATUS**: Status of the voucher batch creation process. Values are:
  0 - Completed Successfully
  1 - Still in Progress
  2 - Error in Voucher creation
  3 - Unable to check the status right now. Unable to run the Unix commands right now.
- **MESSAGE**: Exact message associated with the STATUS. As fetched from the progress file.

Required parameters

This table describes each required parameter.

Parameter	Description
JOB_ID	This is the JOB_ID returned by the PI command CCSVR1=CVB (Create Voucher Batch). Status of the voucher batch creation process will be returned from the progress file associated with this id.

Constraint required parameters

There are no constraint required parameters for this command.

Optional parameters

There are no optional parameters for this command.

Logic and constraints

None

Chapter 5

Error Code Lists

Overview

Introduction

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

In this chapter

This chapter contains the following topics.

PI Chassis Errors

Error List

This table describes the PI Chassis error codes.

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

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

PI Command Errors

Error list

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

Code	Message
1	MSISDN <msisdn> already exists in the user table
2	PRODUCT <product> and PROVIDER <provider> are not a valid combination
3	LANGUAGE <language> does not exist in the language table
4	ACCOUNT_NUMBER <account number> does not exist
5	PRODUCT is null
6	PROVIDER is null
7	PRODUCT <product> does not exist
8	WALLET_TYPE <wallet type> is not valid
9	The account already has WALLET_TYPE <wallet type>
10	The CHARGING_DOMAIN_ID <id> does not exist
11	MSISDN <msisdn> does not exist
12	CURRENCY <currency> does not exist
13	PROVIDER is invalid
14	Expiry dates cannot be set to a date in the past
16	Nothing to change
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

Code	Message
23	The account status <status> prohibits recharge for MSISDN <msisdn>
24	Voucher [Type] <voucher [type]> does not exist
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
50	Number <number> is barred
52	TYPE <type> is not valid
57	FROM <from number> is not valid
58	TO <to number> is not valid
59	MAX_RECORDS <max_records> is out of range
60	The Primary wallet cannot 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>
64	Could not link to existing wallet due to <reason> where <reason> can be: • bad LW_MSISDN <cli> • bad LW_ACCOUNT_NUMBER <number> • LW_MSISDN <cli> has no WALLET_TYPE <type> • LW_ACCOUNT_NUMBER <number> has no WALLET_TYPE <type>
65	Could not link to charging domain due to <reason>

Code	Message
	where <reason> can be: • bad CHARGING_DOMAIN <charging domain> • bad WALLET_REFERENCE <wallet reference>
66	`Invalid operation for subscriber`
67	`Bad date range <start date> to <end date>`
68	`Badly formatted parameter <parameter>`
69	`<text>` where `<text>` reports any other undefined errors.
69	`Data object changing while query was executing for msisdn <MSISDN> please retry query later`
101	`TAG <tag name> does not exist`
102	`VALUE <value> is not valid`
103	`Wrong number of values for the number of tags`
104	`VALUE <value> failed limited value check`
105	`Could not merge MSISDNs <MSISDN1> and <MSISDN2>`
106	`Could not send SMS notification to MSISDN <MSISDN>`
107	`CT_TYPE <credit transfer type> is not defined`
108	`Incorrect PIN`
109	`BPL <bpl> not defined`
110	`User <user> security level is too low for BPL <bpl short name>`
111	`EXTRA_EDR is too long`
112	`Could not find a matching service charge`
113	`EXT<n> specified when EXT<n-1> is not specified`
114	`EXT<n> is invalid`
115	`SMS notification <SMSNOT> not defined`
116	`Call plan error: <text>`
117	`MSISDNs <START_MSISDN> to <END_MSISDN> do not define a valid range`
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`
123	`BPL error code: <BPL error>` where <BPL error> is either as returned to PI from the trigger daemon, or as modified by the BPL error translation table, available in the CCS screens.
125	`Too many zone definitions for MSISDN <MSISDN>`
126	`Too many shapes in zone <Zone name> for MSISDN <MSISDN>`
127	`<parameter> <scenario> is not valid. Valid scenarios: <scenario id 1>:<scenario name 1>|<scenario id 2>:<scenario name 2>|...` where: • <parameter> is either SCENARIO_ID or SCENARIO_NAME. • <scenario name *n*> may be blank if the scenario does not have a name.

Code	Message
	<parameter> <scenario> is not valid. No scenarios defined. Occurs when a scenario is specified and none are defined. <parameter> is either SCENARIO_ID or SCENARIO_NAME.
	A scenario is required. Valid scenarios: <scenario id 1>:<scenario name 1>|<scenario id 2>:<scenario name 2>|... Occurs when both SCENARIO_ID and SCENARIO_NAME are not specified and a scenario is required
128	SCENARIO_ID and SCENARIO_NAME should not be specified together
129	VOUCHER <voucher> is not unique: also specify PROVIDER
130	The VALUE of TAG <tag> creates too many fields
131	Unable to merge wallets from different service providers
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>
140	Error retrieving status, <specific error message>
141	Error retrieving friends, <specific error message>
142	No rewards
143	No balances
144	Balances exceed all rewards
145	No tag data in profile
146	PROVIDER ID <provider> does not exist
147	Too many prefixes
148	MSISDN is subscribed to a CUG. Could not update CLI, account number or profile for MSISDN *number* Where *number* is the subscriber's msisdn number. You cannot change the MSISDN for a subscriber who is part of a closed user group.
149	Wallet not rechargeable
150	PROMOTION <promotion> is not a defined promotion
151	MSISDN <cli> is not subscribed to PROMOTION <promotion>
152	MSISDN <cli> has no subscribed promotions
164	DECRYPTION NOT ALLOWED
165	Error while checking if ccsVoucher binary is already running.
166	ccsVoucher already running (failed to execute even after wait & retries).
167	Error occurred while executing ccsVoucher binary. Unable to execute the binary.
168	PROVIDER is not in incoming request, nor configured in eserv.config (*specific to CCSVR1=COV command)*
169	VOUCHER_TYPE is not in incoming request, nor configured in eserv.config.

Code	Message
170	AUTH_MODULE_NAME is not in incoming request, nor configured in eserv.config.
171	AUTH_RULE_NAME is not in incoming request, nor configured in eserv.config.
172	INITIAL_STATE is not in incoming request, nor configured in eserv.config.
173	BATCH_SIZE is NULL in the incoming request.
174	JOB_ID is not present in the incoming request.
175	Incorrect voucher batch JOB_ID.
176	Error in reading voucher batch progress file.
200	On demand voucher created successfully (specific to CCSVR1=COV command)
202	Invalid Domain name (billing engine name)
203	Invalid AUTH rule name / AUTH module name
204	Invalid Voucher type
205	Other error in voucher creation.
130001	Default PROVIDER not defined by config item pi.CCSVR1.acsCustomerId