Oracle® Communications Network Charging and Control

ACS Provisioning Interface Commands

Release 15.2

January 2026

Copyright

Contents

Chapter 1

Chapter 2

Chapter 3

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*
- *Charging Control Services Provisioning Interface Commands*

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.
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

ACS PI Command List

The following table lists the ACS PI functions and their corresponding commands. To use these commands, the `piAcsSms` package must be installed.

Function	PI Command
Add CLI to ACS Numbers	ACSCLI=ADD
Delete a CLI from ACS Numbers	ACSCLI=DEL
Query a CLI using ACS Numbers	ACSCLI=QRY
Export a control plan for an ACS customer.	ACSCPL=EXP
Import a control plan for an ACS customer.	ACSCPL=IMP
Delete a control plan for an ACS customer.	ACSCPL=DEL
Add an ACS customer.	ACSCST=ADD
Query an ACS customer.	ACSCST=QRY
Change a profile entry.	ACSPFL=CHG
Query a profile entry.	ACSPFL=QRY
Add a service number and associated control plan for a customer.	ACSSNM=ADD
Change the scheduled a control plan for a service number.	ACSSNM=CHG
Query a service number for a customer.	ACSSNM=QRY
Delete a service number and associated control plan for a customer.	ACSSNM=DEL
Add a termination number for a customer.	ACSTNM=ADD
Query a termination number for a customer.	ACSTNM=QRY
Delete a termination number for a customer.	ACSTNM=DEL

About Service Provider Restrictions

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

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

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

Chapter 2

piAcsSms Package

Overview

Introduction

This chapter describes the available PI commands for provisioning ACS customers, CLIs, and profile entries on the SMS.

These commands are added by the `piAcsSms` package that is installed when you install NCC. For more information about the installed PI packages, see *PI User's and Technical Guide*.

In this chapter

This chapter contains the following topics.

Adding a CLI to ACS Numbers by Using PI

About Adding a CLI to ACS Numbers by Using PI

Use the ACSCLI=ADD PI command to add a new CLI in ACS Numbers, and to schedule the specified control plan for the CLI. Specify an existing CLI to update the CLI with new data.

Note: You must specify an existing control plan in the CALLPLAN parameter.

You can also set the list of allowed numbers for the CLI by specifying the optional ALLOWED parameter. If you do not specify ALLOWED, then the PI sets the IGNORE flag for the number list.

After successfully adding a CLI to ACS Numbers, the PI returns this message:

```
ACSCLI=ADD:ACK;
```

If unsuccessful, the PI may return these error codes: 101, 102, 103, 105, 106, 107, 108, 109 and 110.

Required Parameters

Here are the required parameters for this command.

`CLI`

Syntax: `CLI=`*`num`*

Description: CLI for the subscriber.

Format: 18-digit number (greater than 0)

Example: `CLI=1234567`

`CALLPLAN`

Syntax: `CALLPLAN=`*`cp`*

Description: The name of the control plan for the service.

Format: 50 character string

Example: `CALLPLAN=CP2`

Optional Parameters

ACSCLI=ADD accepts the following optional parameters.

ALLOWED

Syntax: `ALLOWED="`*`cell`*`|`*`cell`*`|..."`

Description: A pipe (|) separated list of up to 100 *cell*s, where each *cell* value is an allowed number for the CLI.

Format: "*CELL1*|*CELL2…*"

Note: Each *cell* value is a 15-digit number (zero padded), formatted as follows:

MSCID + *ServingCellID*

Where:

- *MSCID* (Optional) consists of a six digit market ID followed by a three digit switch number
- *ServingCellID* is a six digit number

Example: `ALLOWED="123456789012345|223456789012345"`

Deleting a CLI From ACS Numbers

About Deleting a CLI in ACS Numbers by Using PI

Use the ACSCLI=DEL PI command to delete a CLI from ACS Numbers. Deleting the CLI also unschedules any control plans that are scheduled for the CLI.

After successfully deleting a CLI from ACS Numbers, the PI returns this message:

```
ACSCLI=DEL:ACK;
```

If unsuccessful, the PI may return these error codes: 101, 102, 104, 105, 107, 108, 109 and 110.

Required Parameter

Here is the required parameter for this command.

CLI

Syntax:	`CLI=`*num*
Description:	The CLI for the subscriber. The CLI must exist already.
Format:	18-digit number (greater than 0)
Example:	`CLI=1234567`

Querying a CLI in ACS Numbers

About Querying a CLI in ACS Numbers by Using PI

Use the ACSCLI=QRY PI command to query the ACS Numbers data for the list of allowed numbers for a specified CLI.

After successfully querying a CLI in ACS Numbers, the PI returns this message:

```
ACSCLI=QRY:ACK:ALLOWED=cell1|cell2|…;
```

Where *cell* is a number from the allowed numbers list for the CLI. For details about *cell* number formats, see *ALLOWED* (on page 4) parameter description.

If unsuccessful, the PI may return these error codes: 101, 102, 104, 105, 107 and 109.

Required Parameter

Here is the required parameter for this command.

CLI

Syntax:	`CLI=`*num*
Description:	The CLI for the subscriber. The CLI must exist already.
Format:	18-digit number (greater than 0)
Example:	`CLI=1234567`

Exporting a Control Plan

About Exporting a Control Plan by Using PI

Use the ACSCPL=EXP PI command to export a control plan for a specified ACS customer to a **.cpl** file. For example, the following PI command exports the Postpaid control plan for the Telco customer:

```
ACSCPL=EXP:CUSTOMER=Telco,CALLPLAN=Postpaid;
```

The PI exports the control plan to the following directory by default:

/IN/service_packages/PI/callplans

You can set a different control plan export directory by configuring the `exportCallPlanDirectory` parameter in the pi, ACSCPL section of the **eserv.config** configuration file. See *PI User's and Technical Guide* for more information.

After successfully exporting a control plan, the PI returns this message:

```
ACSCPL=EXP:FILENAME=export_path/filename;
```

Where:

- *export_path* is the full directory path for the exported control plan file on the SMS; for example, **/IN/service_packages/callplans**
- *filename* has the following format, based on the customer name, control plan name and control plan version number: *customer*___*controlplan*___*version*.**cpl**; for example, **Telco___Postpaid___1.cpl**

If unsuccessful, the PI may return these error codes: 101, 102, 103, 107, and 118.

Required Parameters

Here are the required parameters for this command.

CUSTOMER

Syntax:	`CUSTOMER=`*`name`*
Description:	The name of an existing customer.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	`CUSTOMER=Customer A`

CALLPLAN

Syntax:	`CALLPLAN=`*`string`*
Description:	The name of the control plan. You must specify an existing control plan that belongs to the specified customer.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	`CALLPLAN=Control Plan 2`

Importing a Control Plan

About Importing a Control Plan by Using PI

Use the ACSCPL=IMP PI command to import a control plan for a specified ACS customer from a **.cpl** file, and to compile the control plan after it is imported. For example, the following PI command imports the Postpaid control plan for the Telco customer from the **Postpaid_test.cpl** file:

```
ACSCPL=IMP:CUSTOMER=Telco,CALLPLAN=Postpaid,FILENAME=/IN/service_packages/PI
/callplans/Postpaid_test.cpl;
```

If the control plan:

- Does not exist for the specified customer, then the PI creates a new control plan and sets its version number to 1 (one)
- Already exists for the specified customer, then the PI updates the control plan and increments its version number, provided that:
  - Only a single successfully compiled version of the control plan exists
  - No other control plans use the control plan structure
  - The control plan is not associated with an ACS service number, an ACS CLI number, or a CCS product type

After successfully importing and compiling a control plan, the PI returns this message:

```
ACSCPL=IMP:ACK;
```

If unsuccessful, the PI may return these error codes: 101, 102, 107, 118, 120 and 121.

Required Parameters

Here are the required parameters for this command.

CUSTOMER

Syntax:	`CUSTOMER=`*`name`*
Description:	The name of an existing customer.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	`CUSTOMER=Customer A`

CALLPLAN

Syntax:	`CALLPLAN=`*`string`*
Description:	The name that the new control plan will use.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	`CALLPLAN=Control Plan 2`

FILENAME

Syntax:	`FILENAME=`*`string`*
Description:	The name of the **.cpl** import file. Specify the filename by using the following format: *`file_name`*`.`**`cpl`** Where *file_name* includes the full directory path of the **.cpl** import file on the SMS; for example, **/IN/service_packages/PI/callplans/Postpaid_test.cpl**.
Type:	String
Optionality:	Required
Example:	`FILENAME=`**/IN/service_packages/PI/callplans/Postpaid_test.cpl**

Deleting a Control Plan

About Deleting a Control Plan by Using PI

Use the ACSCPL=DEL PI command to delete a control plan for a specified ACS customer. For example, the following PI command deletes the Postpaid control plan for the Telco customer:

```
ACSCPL=DEL:CUSTOMER=Telco,CALLPLAN=Postpaid;
```

You can delete a control plan providing:

- Only one entry for the control plan exists in the database
- The control plan structure is not being used by other control plans

- The control plan is not associated with an ACS service number, an ACS CLI number, or a CCS product type

After successfully deleting a control plan, the PI returns this message:

```
ACSCPL=DEL:ACK;
```

If unsuccessful, the PI may return these error codes: 101, 102, 103, 107 and 118.

Required Parameters

Here are the required parameters for this command.

`CUSTOMER`

Syntax:	`CUSTOMER=`*`name`*
Description:	The name of an existing customer.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	`CUSTOMER=Customer A`

`CALLPLAN`

Syntax:	`CALLPLAN=`*`string`*
Description:	The name of the control plan. You must specify an existing control plan that belongs to the specified customer.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	`CALLPLAN=Control Plan 2`

Adding an ACS Customer

About Adding an ACS Customer by Using PI

Use the ACSCST=ADD PI command to add an ACS customer to the database.

To successfully add a customer, your PI user must be associated with the default service provider named "Boss". A negative acknowledgement (NACK) is returned if you attempt to add a customer and your PI user is not associated with the "Boss" service provider. See *About Service Provider Restrictions* (on page 2) for more information.

Note: The "Boss" service provider is automatically created at installation.

After successfully adding a customer, the PI returns this message:

```
ACSCST=ADD:ACK;
```

If unsuccessful, the PI can return any of the following error codes: 136, 137, 138, 140, or 142.

Required Parameters

Here are the required parameters for this command.

`CUSTOMER`

Syntax:	`CUSTOMER=`*`name`*
Description:	The name of the customer to add.

Type: String

Optionality: Required.

Allowed: A string of up to 50 characters.

Example: `CUSTOMER=Customer A`

PARENT

Syntax: PARENT=*string*

Description: The name of the customer's parent customer.

Type: String

Optionality: Required if the RESELLER_FLAG is set to A. The parent customer must be a reseller.

Optional if the RESELLER_FLAG is set to N or null.

If RESELLER_FLAG is set to R, then you cannot set the parent customer.

Allowed: A string of up to 50 characters.

Example: `PARENT=Customer A`

Optional Parameters

ACSCST=ADD accepts the following optional parameters.

DESCRIPTION

Syntax: DESCRIPTION=*string*

Description: The customer's description.

Type: String

Optionality: Optional.

Allowed: A string of up to 2000 characters.

Example: `DESCRIPTION=A customer description`

CUSTOMER_REFERENCE

Syntax: CUSTOMER_REFERENCE=*string*

Description: The customer reference ID.

Type: String

Optionality: Optional

Allowed: A string of up to 749 characters.

Example: `CUSTOMER_REFERENCE=CRX001A`

TELCO_MANAGED

Syntax: `TELCO_MANAGED=Y|N`

Description: Specifies whether the customer is managed by the operator.

Type: Boolean

Optionality: Optional (default used if not set).

Allowed: Y – The customer is managed by the operator.

N – The customer is not managed by the operator.

Default: Y

Example: `TELCO_MANAGED=N`

RESELLER_FLAG

Syntax:	`RESELLER_FLAG=N|R|A`
Description:	Defines whether or not the customer is a reseller or an agent.
Type:	String
Optionality:	Optional (default used if not set).
Allowed:	N – The customer is not a reseller or an agent. R – The customer is a reseller. A – The customer is an agent.
Default:	N
Example:	`RESELLER_FLAG=N`

Querying an ACS Customer

About Querying an ACS Customer by Using PI

Use the ACSCST=QRY PI command to query the database for the details of the specified ACS customer. After successfully performing a customer query, the PI returns this message:

```
ACSCST=QRY:ACK:
    CUSTOMER=name, DESCRIPTION=customer_description,
CUSTOMER_REFERENCE=customer_reference, TELCO_MAN=Y|N, RESELLER_FLAG=N|R|A,
[PARENT=parent_name]
```

Where:

- *name* is the name of the customer.
- *customer_description* is the description for the customer.
- *customer_reference* is the customer reference ID.
- *parent_name* is name of the customer's parent. No value is returned when the operator is the customer's parent.

If the PI fails to find the specified customer, then the PI returns error code 118.

Required Parameter

Here is the required parameter for this command.

CUSTOMER

Syntax:	`CUSTOMER=`*name*
Description:	The name of an existing customer.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	`CUSTOMER=Customer A`

Changing a Profile Entry

About Changing a Profile Entry by Using PI

Use the ACSPFL=CHG PI command to change a profile entry in the database. After successfully changing a profile entry, the PI returns this message:

```
ACSPFL=CHG:ACK;
```

If unsuccessful, then the PI may return these error codes: 101, 102, 107, 110, 111, 112, 113, 114, 115, 116 and 117.

Required Parameters

Here are the required parameters for this command.

TABLE

Syntax:	`TABLE=`*`table`*
Description:	The name of the profile table that holds the profile entry you want to change or query. You must specify an existing profile table.
Type:	30-character string
Example:	`TABLE=ACS_PROFILE_DETAILS`

ID

Syntax:	`ID=`*`id`*
Description:	The ID of the row.
Type:	Integer
Example:	`ID=45`

TAG

Syntax:	`TAG=`*`tag_id`*
Description:	The profile tag.
Type:	Hex or String
Allowed:	Either a hexadecimal value prefixed by 0x, for example, 0x123ABC or a pre-defined name corresponding to PROFILE_TAG_NAME in the ACS_PROFILE_DETAILS table.
Default:	HEX
Notes:	If TAG is: • A predefined name, any specified TYPE is ignored • A hexadecimal value, and TYPE is omitted, the value will be returned as hexadecimal data • Omitted, a hex dump of the whole profile is returned
Example:	`TAG=0x123ABC`

Optional Parameters

The ACSPFL=CHG PI command accepts the following optional parameters.

PROFILE_COL

Syntax:	`PROFILE_COL=`*`column_name`*
Description:	The name of the profile column.
Type:	String
Optionality:	Optional (default used if omitted)
Allowed:	A valid profile column name from the specified profile table.
Default:	PROFILE
Example:	`PROFILE_COL=CLI`

ID_COL

Syntax: ID_COL=*id_col_name*

Description: The name of the ID column.

Type: String

Optionality: Optional (default used if omitted).

Allowed: ID

Default: ID

Notes: A valid column in TABLE containing unique ID.

Example: `ID_COL=ID`

TYPE

Syntax: TYPE=*tag_type*

Description: The type of the tag.

Type: String

Optionality: Optional (default used if omitted).

Allowed: Valid options are:

- STRING – String value
- DATE – Date value
- INTEGER – 32-bit signed integer value
- HEX – Raw hexadecimal data
- TN – Telephone number

Default: HEX

Notes: DATE type values have the following format: YYYYDDMMHHMMSS; for example, 20131609100500.

Example: `TYPE=DATE`

VALUE

Syntax: VALUE=*val*

Description: The value of the tag.

Type: Defined by the TYPE parameter.

Optionality: Optional

Allowed:

Default: If VALUE is not specified, any existing profile tag will be deleted.

Notes: For TYPE = TN, the VALUE format:

- Must contain only numeric digits
- First character is interpreted as a TN type
- TN type has range 0-5
- TN has max length 33 characters (including TN type)

Examples:

`VALUE=12345`	Integer type
`VALUE=Monday`	String type
`VALUE=2009010112345`	Date type (*YYYYMMDDHHMMSS*)
`VALUE=0x123ABC`	Hex type
`VALUE=12345`	TN type

Querying a Profile Entry

About Querying a Profile Entry by Using PI

Use the ACSPFL=QRY PI command to query a profile entry in the database.

After successfully querying a profile entry, the PI returns this message if TAG was specified:

- ACSPFL=QRY:ACK:TAG=*tag*,VALUE=*value*;
- ACSPFL=QRY:ACK:TAG=*tag*;

The PI returns this message if TAG was not specified:

- ACSPFL=QRY:ACK:
```
    TAG=hex tag,VALUE=hex value[,
    TAG=hex tag,VALUE=hex value[,
    ...]];
```

Note: No carriage returns are returned.

If unsuccessful, then the PI may return these error codes: 101, 102, 107, 111, 112, 113, 114, 115 and 116.

Required Parameters

Here are the required parameters for this command.

TABLE

Syntax:	TABLE=*table*
Description:	The name of the profile table that holds the profile entry you want to change or query. You must specify an existing profile table.
Type:	30-character string
Example:	TABLE=ACS_PROFILE_DETAILS

ID

Syntax:	ID=*id*
Description:	The ID of the row.
Type:	Integer
Example:	ID=45

Optional Parameters

The ACSPFL=QRY PI command accepts the following optional parameters.

PROFILE_COL

Syntax:	PROFILE_COL=*column_name*
Description:	The name of the profile column.
Type:	String
Optionality:	Optional (default used if omitted)
Allowed:	A valid profile column name from the specified profile table.
Default:	PROFILE
Example:	PROFILE_COL=CLI

ID_COL

Syntax:	`ID_COL=`*`id_col_name`*
Description:	The name of the ID column.
Type:	String
Optionality:	Optional (default used if omitted).
Allowed:	ID
Default:	ID
Notes:	A valid column in TABLE containing unique ID.
Example:	`ID_COL=ID`

TAG

Syntax:	`TAG=`*`tag_id`*
Description:	The profile tag.
Type:	Hex or String
Allowed:	Either a hexadecimal value prefixed by 0x, for example, 0x123ABC or a pre-defined name corresponding to PROFILE_TAG_NAME in the ACS_PROFILE_DETAILS table.
Default:	HEX
Notes:	If TAG is: • A predefined name, any specified TYPE is ignored • A hexadecimal value, and TYPE is omitted, the value will be returned as hexadecimal data • Omitted, a hex dump of the whole profile is returned
Example:	`TAG=0x123ABC`

TYPE

Syntax:	`TYPE=`*`tag_type`*
Description:	The type of the tag.
Type:	String
Optionality:	Optional (default used if omitted).
Allowed:	Valid options are: • STRING – String value • DATE – Date value • INTEGER – 32-bit signed integer value • HEX – Raw hexadecimal data • TN – Telephone number
Default:	HEX
Notes:	DATE type values have the following format: YYYYDDMMHHMMSS; for example, 20131609100500.
Example:	`TYPE=DATE`

Adding Customer Service Numbers

About Adding Customer Service Numbers by Using PI

Use the ACSSNM=ADD PI command to add a service number for a specified ACS customer. You can also use this command to add a scheduled control plan to the service number. After successfully adding a service number for a customer, the PI returns this message:

```
ACSSNM=ADD:ACK;
```

If unsuccessful, then the PI may return any of the following error codes: 118, 140, 141, or 143.

Required Parameters

Here are the required parameters for this command.

CUSTOMER

Syntax:	`CUSTOMER=`*`name`*
Description:	The name of an existing customer.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	`CUSTOMER=Customer A`

SERVICE_NUMBER

Syntax:	`SERVICE_NUMBER=`*`string`*
Description:	The service number to add or delete.
Type:	String
Optionality:	Required
Allowed:	A string of up to 32 characters. Valid characters are 0–9, A–F, *, #.
Example:	`SERVICE_NUMBER=115`

Optional Parameter

The ACSSSN=ADD PI command accepts the following optional parameter.

CONTROLPLAN

Syntax:	`CONTROLPLAN=`*`string`*
Description:	The control plan name.
Type:	String
Optionality:	Optional
Allowed:	A string of up to 50 characters
Example:	`CONTROLPLAN=Control Plan 2`

Changing the Scheduled Control Plan

About Changing the Scheduled Control Plan by Using PI

Use the ACSSNM=CHG PI command to change or add the control plan that is scheduled for a customer service number. You can schedule a future date when the control plan will become active for the service number. Otherwise the control plan is activated immediately for the service number. For example, the following PI command schedules the Postpaid control plan for the 012234787 service number for the Telco customer:

```
ACSSNM=CHG:CUSTOMER=Telco,SERVICE_NUMBER=012234787,CONTROLPLAN=Postpaid,SCHE
DULE_DATE=20130901080000;
```

After successfully updating the control plan scheduled to a customer service number, the PI returns this message:

```
ACSSNM=CHG:ACK;
```

If unsuccessful, then the PI may return any of the following error codes: 101, 102, 103, 108, 118, 141, 143, 144, or 145.

Required Parameters

Here are the required parameters for this command.

CUSTOMER

Syntax:	CUSTOMER=*name*
Description:	The name of an existing customer.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	CUSTOMER=Customer A

SERVICE_NUMBER

Syntax:	SERVICE_NUMBER=*string*
Description:	The service number for which you want to add or change the scheduled control plan.
Type:	String
Optionality:	Required
Allowed:	A string of up to 32 characters. Valid characters are 0–9, A–F, *, #.
Example:	SERVICE_NUMBER=012234787

CONTROLPLAN

Syntax:	CONTROLPLAN=*string*
Description:	The name of an existing control plan that the service number will use.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters
Example:	CONTROLPLAN=Control Plan 2

Optional Parameter

The ACSSSN=CHG PI command accepts the following optional parameter.

SCHEDULE_DATE

Syntax:	`SCHEDULE_DATE=`*`date`*
Description:	The date when the control plan will be activated for the service number. Specify a date in the following format: YYYYMMDDHHmmSS.
Type:	String
Optionality:	Optional
Notes:	To activate the control plan immediately for the service number, do not specify a schedule date.
Example:	`SCHEDULE_DATE=20130901080000`

Querying Customer Service Numbers

About Querying Customer Service Numbers by Using PI

Use the ACSSNM=QRY PI command to query the database for the details of the service numbers allocated to a specified ACS customer. After successfully performing a service number query, the PI returns this message:

```
ACSSNM=QRY:ACK:
    CUSTOMER=name, SERVICE_NUMBER=n[,SERVICE_NUMBER=n][,...]
```

Where:

- *name* is the name of the customer.
- *n* is a service number allocated to the customer.

If unsuccessful, then the PI may return either of the following error codes: 118 or 140.

Required Parameter

Here is the required parameter for this command.

CUSTOMER

Syntax:	`CUSTOMER=`*`name`*
Description:	The name of an existing customer.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	`CUSTOMER=Customer A`

Deleting Customer Service Numbers

About Deleting Customer Service Numbers by Using PI

Use the ACSSNM=DEL PI command to delete a service number for a specified ACS customer. After successfully deleting a service number, the PI returns this message:

```
ACSSNM=DEL:ACK;
```

If unsuccessful, then the PI may return either of the following error codes: 118 or 140.

Required Parameters

Here are the required parameters for this command.

CUSTOMER

Syntax:	`CUSTOMER=`*`name`*
Description:	The name of an existing customer.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	`CUSTOMER=Customer A`

SERVICE_NUMBER

Syntax:	`SERVICE_NUMBER=`*`string`*
Description:	The service number to add or delete.
Type:	String
Optionality:	Required
Allowed:	A string of up to 32 characters. Valid characters are 0–9, A–F, *, #.
Example:	`SERVICE_NUMBER=115`

Adding Customer Termination Numbers

About Adding Customer Termination Numbers by Using PI

Use the PI ACSTNM=ADD command to add termination numbers for a specified ACS customer.

To add a single termination number for a customer specify, only the *CUSTOMER* (on page 6) and *PREFIX* (on page 19) parameters.

To add a range of termination numbers for a customer, specify all the parameters. You specify the range by using a combination of the PREFIX, *MIN_LENGTH* (on page 19) and *MAX_LENGTH* (on page 19) parameters. For example, you could specify the following range:

```
PREFIX=01473, MIN_LENGTH=6, MAX_LENTGH=10
```

The resulting termination number range would be: `01473`*`XYYYY`*

Where *`X`* is a digit that must be present in the termination number, and *`Y`* is a digit that may be present up to MAX_LENGTH.

After successfully adding termination numbers for a customer, the PI returns this message:

```
ACSTNM=ADD:ACK;
```

If unsuccessful, then the termination numbers are not added and the PI may return any of the following error codes: 118, 130, 131, 132, 133, 139, or 140.

Required Parameters

Here are the required parameters for this command.

CUSTOMER

Syntax:	`CUSTOMER=`*`name`*
Description:	The name of an existing customer.
Type:	String
Optionality:	Required

Allowed: A string of up to 50 characters.

Example: `CUSTOMER=Customer A`

`PREFIX`

Syntax: `PREFIX=`*`string`*

Description: The termination number prefix to add or delete.

Type: String

Optionality: Required.

Allowed: A string of up to 32 characters. Valid characters are 0–9, A–F, *, #.

Example: `PREFIX=0179`

Optional Parameters

ACSTNM=ADD accepts the following optional parameters.

`MIN_LENGTH`

Syntax: `MIN_LENGTH=`*`integer`*

Description: The minimum length for the termination number.

Type: Integer

Optionality: Optional (default used if not set).

Allowed: A value that is not less than the PREFIX length and not greater than 32.

Default: PREFIX length

Example: `MIN_LENGTH=4`

`MIN_LENGTH`

Syntax: `MAX_LENGTH=`*`integer`*

Description: The maximum length for the termination number.

Type: Integer

Optionality: Optional (default used if not set).

Allowed: A value that is not less than MIN_LENGTH and not greater than 32.

Default: PREFIX length plus one (1), or 32 if the PREFIX length is 32.

Example: `MAX_LENGTH=10`

Querying Customer Termination Numbers

About Querying Customer Termination Numbers by Using PI

Use the ACSTNM=QRY PI command to query the database for the details of the termination numbers assigned to a specified ACS customer. After successfully performing a termination number query, the PI returns this message:

```
ACSTNM=QRY:ACK:
   CUSTOMER=name, TERMINATION_NUMBER=n[,TERMINATION_NUMBER=n[,...]]
```

Where:

- *name* is the name of the customer.
- *n* is a valid termination number for the customer.

If unsuccessful, then the PI may return the following error codes: 118 or 140.

Required Parameters

Here are the required parameters for this command.

CUSTOMER

Syntax:	`CUSTOMER=`*`name`*
Description:	The name of an existing customer.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	`CUSTOMER=Customer A`

PREFIX

Syntax:	`PREFIX=`*`string`*
Description:	The termination number prefix to add or delete.
Type:	String
Optionality:	Required.
Allowed:	A string of up to 32 characters. Valid characters are 0–9, A–F, *, #.
Example:	`PREFIX=0179`

Deleting Customer Termination Numbers

About Deleting Customer Termination Numbers by Using PI

Use the ACSTNM=DEL PI command to delete a termination number prefix for a specified ACS customer. After successfully deleting a termination number prefix, the PI returns this message:

```
ACSTNM=DEL:ACK;
```

If unsuccessful, then the PI may return any of the following error codes: 118, 134, or 140.

Required Parameter

Here is the required parameter for this command.

CUSTOMER

Syntax:	`CUSTOMER=`*`name`*
Description:	The name of an existing customer.
Type:	String
Optionality:	Required
Allowed:	A string of up to 50 characters.
Example:	`CUSTOMER=Customer A`

Managing FCI Data by Using PI

About Managing FCI Data by Using PI

Use the ACSPFL=CHG PI command to update the furnish charging information (FCI) data held in the following profile tags in the ACS service number profile block:

- FCI Country Code
- FCI On
- FCI Service Code

For more information about using ACSPFL=CHG, see *Changing a Profile Entry* (on page 10).

Use the ACSPFL=QRY PI command to query the FCI data for a customer. For more information, see *Querying a Profile Entry* (on page 13).

For more information about profile blocks and profile tags, see the discussion on the main components of ACS in *ACS Technical Guide*.

Chapter 3

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

Format

NACK:*Code-Message*;

Example: `ACSCLI=QRY:NACK:70-TOO MANY SESSIONS`

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.

Code	Message	Description
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.
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

Format

Command:NACK:*Code–Message*:*value*;

Where:

- *Command* is the PI command
- *Code* is the PI error code
- *Message* is the PI error message and can contain any characters except semi-colon
- In the message, *value* will be replaced with a suitable value by the command.

Example: `ACSCLI=QRY:NACK:104-CLI does not exist:912233445566`

Error List

This table lists the error codes and error code descriptions for the PI commands listed in the ACS PI command list.

Code	Message	Commands
101	*text* Where the *text* contains the details of the error. This error is used to report any other errors that are not defined below.	All commands

Code	Message	Commands
102	`Badly formatted parameter` *parameter* Where *parameter* is the name of the parameter causing the error.	All commands

103	`CALLPLAN does not exist` *`control_plan`*
	Where *control_plan* is the name of an invalid control plan.
		ACSCLI=ADD
		ACSCPL=EXP
		ACSCPL=DEL
		ACSSNM=CHG

104	`CLI does not exist` *cli*
	Where *cli* is an invalid CLI number.
		ACSCLI=DEL
		ACSCLI=QRY

105	`No required parameters`	All commands
106	`Too many numbers in ALLOWED list`	ACSCLI=ADD
107	`Invalid action` *`action`* Where *action* is an invalid command action.	All commands

108	`SQL write error`	ACSCLI=ADD
		ACSCLI=DEL
		ACSSNM=CHG

109	`SQL read error`	ACSCLI=ADD
		ACSCLI=DEL
		ACSCLI=QRY

110	`Internal error encoding profile`	ACSCLI=ADD
		ACSCLI=DEL
		ACSPFL=CHG

111	`TABLE does not exist` *`table_name`*
	Where *table_name* is an invalid database table name.
		ACSPFL=CHG
		ACSPFL=QRY

112	`ID_COLUMN does not exist` *`column_name`*
	Where *column_name* is an invalid column name for the specified table.
		ACSPFL=CHG
		ACSPFL=QRY

113	`ID does not exist` *`id`*
	Where *id* is invalid.
		ACSPFL=CHG
		ACSPFL=QRY

114	`PROFILE_COLUMN does not exist` *`column_name`*
	Where *column_name* is an invalid column name for the profile.
		ACSPFL=CHG
		ACSPFL=QRY

115	`TAG does not exist` *`tag_name`*
	Where *`tag_name`* is an invalid name for the specified profile tag.
		ACSPFL=CHG
		ACSPFL=QRY

116	`TYPE is not valid` *`type_name`*
	Where *type_name* is an invalid profile type.
		ACSPFL=CHG
		ACSPFL=QRY

117	`VALUE is not valid` *`value`*
	Where *value* is an invalid value for the specified profile tag.
		ACSPFL=CHG
		ACSPFL=QRY

Code	Message	Commands

118	`Customer does not exist` *`customer_name`*
	Where *customer_name* is not a valid ACS customer.
		ACSCST=QRY
		ACSCPL=EXP
		ACSCPL=IMP
		ACSCPL=DEL
		ACSSNM=ADD
		ACSSNM=CHG
		ACSSNM=QRY
		ACSSNM=DEL
		ACSTNM=ADD
		ACSTNM=QRY
		ACSTNM=DEL

120	`FILENAME does not exist` *`file_name`*	ACSCPL=IMP
121	`FILENAME has insufficient permissions` *`file_name`*	ACSCPL=IMP
130	`Invalid termination prefix`	ACSTNM=ADD
131	`Invalid minimum termination length`	ACSTNM=ADD
132	`Invalid maximum termination length`	ACSTNM=ADD
133	`Overlapping termination ranges not allowed`	ACSTNM=ADD
134	`Termination number or range does not exist`	ACSTNM=DEL
135	`Service number not found for customer`	ACSSNM=DEL
136	`Parent customer not found`	ACSCST=ADD
137	`Reseller cannot have a parent customer`	ACSCST=ADD
138	`Reseller cannot have a parent customer`	ACSCST=ADD
139	`Minimum termination number length greater than maximum termination number length`	ACSTNM=ADD

140	`Duplicate service numbers not allowed`	ACSCST=ADD
		ACSSNM=ADD
		ACSSNM=QRY
		ACSSNM=DEL
		ACSTNM=ADD
		ACSTNM=QRY
		ACSTNM=DEL

141	`Parameter too long`	ACSSNM=CHG
142	`Customer already exists`	ACSCST=ADD

143	`Invalid service number`	ACSSNM=ADD
		ACSSNM=CHG

144	`Invalid customer name`	ACST=ADD
		ACSPL=IMP
		ACSPL=EXP
		ACSPL=DEL

145	`Parameter too short`	ACSSNM=CHG
146	`Invalid date`	ACSSNM=CHG