

Oracle® Communications Network Charging and Control

Virtual Private Network Technical Guide



Release 15.2

January 2026

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

Copyright

Copyright © 2026, Oracle and/or its affiliates.

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

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

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

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

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

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

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

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

Contents

About This Document	v
Document Conventions	vi
Chapter 1	
System Overview	1
Overview	1
What is the Virtual Private Network Service?	1
Main Components of VPN	3
Statistics	7
EDRs	9
About Secure SSL Connection to the Database	10
Chapter 2	
Configuration.....	11
Overview	11
Configuring the SLEE.cfg	11
Configuring acs.conf for the SLC	12
eserv.config Configuration.....	15
VPN service loader configuration	16
VPN Specific Announcements	20
Defining the VPN Screen Labels Language	25
Defining the VPN Help Screen Language	25
Setting up the Screens	26
Chapter 3	
Background Processes	35
Overview	35
libvpnService	35
libvpnMacroNodes.....	36
libvpnChassisActions.....	36
Chapter 4	
Tools and Utilities	39
Overview	39
Examining Network and Station Profiles	39
Chapter 5	
About Installation and Removal	43
Overview	43
Installation and Removal Overview	43
Restarting the Stats Daemon	44

About This Document

Scope

The scope of this document includes all the information required to install, configure and administer the Virtual Private Network (VPN) application.

Audience

This guide was written primarily for system administrators and persons installing, configuring and administering the Virtual Private Network (VPN) application. However, sections of the document may be useful to anyone requiring an introduction to the application.

Prerequisites

Although there are no prerequisites for using this guide, familiarity with the target platform would be an advantage.

A solid understanding of Unix and a familiarity with IN concepts are an essential prerequisite for safely using the information contained in this technical guide. Attempting to install, remove, configure or otherwise alter the described system without the appropriate background skills, could cause damage to the system; including temporary or permanent incorrect operation, loss of service, and may render your system beyond recovery.

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:

- Service Management System Technical Guide
- Advanced Control Services Technical Guide
- Virtual Private Network User's Guide

Document Conventions

Typographical Conventions

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

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

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

System Overview

Overview

Introduction

This chapter provides a high-level overview of the application. It explains the basic functionality of the system and lists the main components.

It is not intended to advise on any specific Oracle Communications Network Charging and Control (NCC) network or service implications of the product.

In this Chapter

This chapter contains the following topics.

<i>What is the Virtual Private Network Service?</i>	1
<i>Main Components of VPN</i>	3
<i>Statistics</i>	7
<i>EDRs</i>	9
<i>About Secure SSL Connection to the Database</i>	10

What is the Virtual Private Network Service?

Introduction

The Virtual Private Network (VPN) service provides a fully IN-based, feature-rich VPN solution with intuitive user interfaces, available on industry-standard platforms. Whilst being simple and easy to use, it also provides enhanced functionality for more experienced users.

The basic Virtual Private Network (VPN) service connects multiple locations together. Each VPN network has its own private numbering plan for mapping numbers in the private plan to the numbers required to correctly route the call through the PSTN (or mobile network). In addition to this simple number translation service, you can perform additional processing to add further value to the offered service.

Individual phone numbers can be provided with profiles that specify the available phone operations. For example, users can implement outgoing call barring to prevent people from making certain types of calls.

The VPN service includes:

- *Originating features*
- *Terminating features*
- *Management hotline features*
- *Global service features*
- *IN features*

The facility to access a corporate VPN from points outside the VPN (off-net) is provided, effectively giving access to the corporate telephone network from any phone. Access to this very powerful feature requires rigorous security to prevent fraudulent use.

Users of off-net access can also inherit the features of the service they have in their office, such as speed-dial codes, and the ability to divert calls from their office phone, and so on.

Originating features

Here is the list of VPN originating features:

- *Called party options: on-net, off-net, speed dial*
- *Station PIN*
- *Station "Roaming" feature*
- *Account code feature (un-validated entry, or explicit list)*
- *Network and station level speed dials*
- *Network and station level destination filtering lists*
- *Station off-net call bar (optional PIN override)*
- *Optional filtering bypass for Network speed dials*
- *Tariffing*
- *Network site code (inter-network dialing)*
- *Mapped network prefixes*
- *Closed user groups (restricted and non-restricted)*

Terminating features

Here is the list of VPN terminating features:

- *Incoming barring options (all, off-net, none)*
- *Network and station incoming off-net filtering lists*
- *Follow me number*
- *Time of day, time of week and day of year hunting scheduling*
- *Hunt on busy/no-answer/unconditional flags*
- *Sequential hunting list*
- *Routing failure alternate number*
- *Administrator restriction of forwarding addresses*
- *Choice of addressing mechanisms*
 - On-net address
 - PSTN address
 - Virtual DDI address
 - Global virtual numbering system (GVNS) addressing
- *Simple network-level carrier pre-select*
 - National prefix replacement
 - International prefix replacement
- *CLI presentation options*
 - Calling line display: extension address or full address
 - CLI restricted
- *Tariffing*
- *Closed user groups*
 - Restricted
 - Non-restricted
 - PIN access requirement

Management hotline features

Here is the list of VPN management hotline features:

- *Access management hotline through private or public address*
- *Per-station management feature restriction*
- *Off-net calling through management hotline*
- *Set follow-me/routing failure number*
- *Set new PIN*
- *Set station speed dials*

Global service features

Here is the list of VPN global service features:

- *Optional/configurable help line for VPN members*
- *Optional/configurable help announcements*

IN features

Here is the list of VPN IN features:

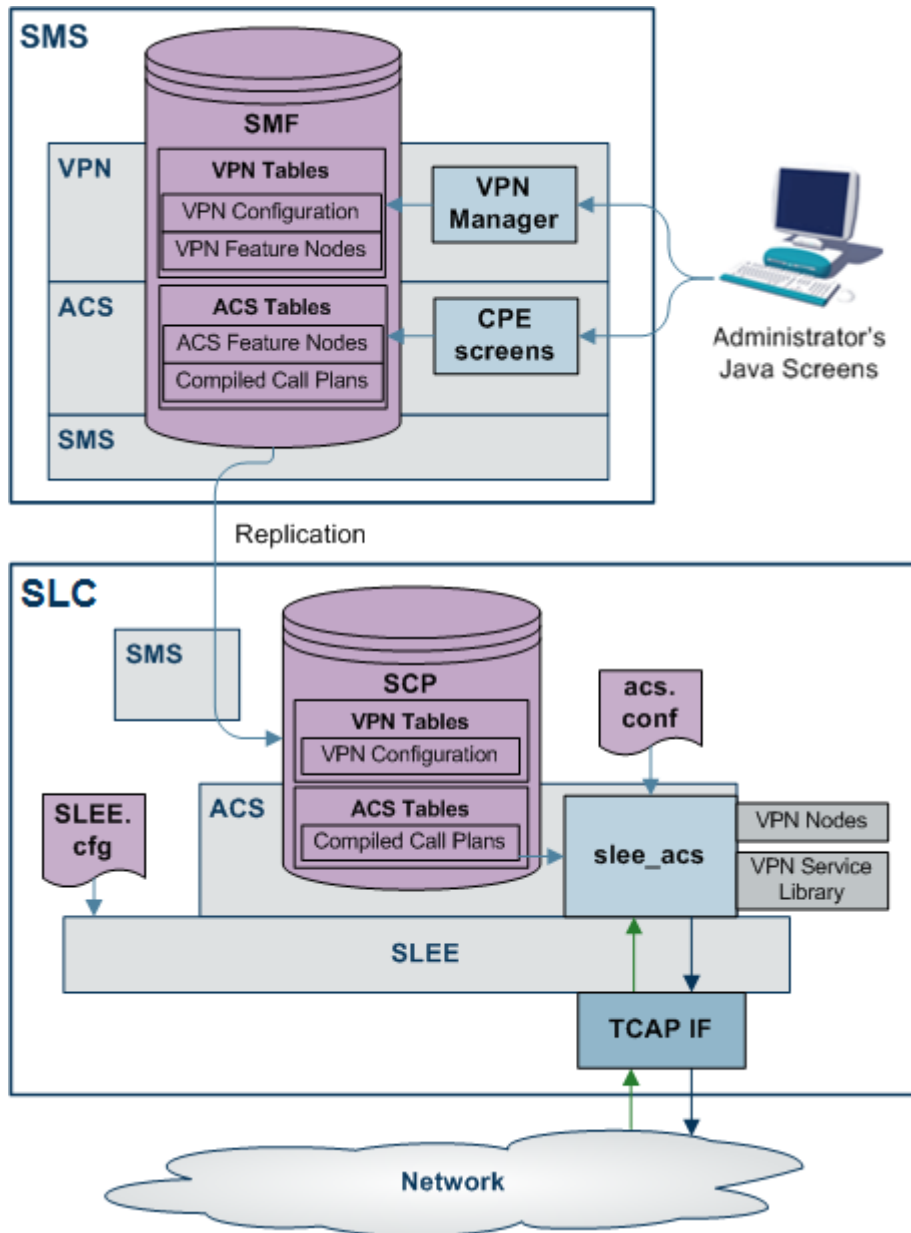
- *Option A. Re-trigger for originating/terminating processing*
- *Option B. Internal service handover without re-trigger*
- *Configurable digit prefixing for termination numbers*
- *Generic number normalization*
- *Service bypass for emergency numbers*

Main Components of VPN

Diagram

The VPN system components are split between the Service Management System (SMS) and the Service Logic Controller (SLC).

This diagram shows the relationship between the major components on each system, and the ACS framework, in the context of NCC components.



Description

The component diagram shows how the VPN service integrates with, and interacts with, the layered ACS and SMS components.

The VPN service:

- Has a specific ACS service library. This is implemented as a shared library using the standard ACS mechanism described in ACS Technical Guide.
- Uses service specific feature nodes and service specific chassis actions. These are implemented as shared libraries using the standard ACS pluggable feature node/chassis action mechanisms.
- Uses the SMS for two-way replication.
- Runs all processes as the "acs_oper" user, created by ACS.

- Database tables are owned by the "acs_admin" Oracle user, created by ACS.
- Provides seven sample control plans that the operator can save on the SMS. These provide the operator with a sample set of services. To use the sample control plans and services, they must first be copied on to the client machine, and then compiled using the standard ACS tools. The sample control plans are:
 - Fixed Line Originating
 - Mobile Originating
 - Terminating
 - Management
 - Originating Alternative
 - Terminating Alternative
 - Management Alternative

They can be selected as the default control plans to be used by new VPN networks for originating, terminating, and management functions.
- The system administrator can modify the sample control plans to give site specific behavior. In addition, individual VPNs can be given their own dedicated call plans to replace any of the default call plans.

VPN Profile Block list

Here are the profile blocks available when VPN is installed.

Name	Description
VPN Network Profile	Contains most of the information you can specify in the VPN edit network, for example: <ul style="list-style-type: none"> • Account code maximum length • Outgoing barred/allowed list type • Incoming barred/allowed list type • VPN network SD no check • VPN present private address
VPN Station Profile	Contains most of the information you can specify in the VPN edit station, for example: <ul style="list-style-type: none"> • Outgoing barred/allowed list type • Incoming barred/allowed list type • VPN bar all incoming • VPN bar off network incoming

VPN primary tags

Here is a list of the profile primary tags used in the VPN service.

Note: These tags are only present for VPN and are displayed on the ACS Configuration screen, **Profile Tag Details** tab.

Description	Hex	Decimal
Network SD No Check	0x30001	196609
Present Private Address	0x30002	196610
Bar All Incoming	0x30003	196611
Bar Off Network Incoming	0x30004	196612
PIN Prefix	0x30005	196613

Description	Hex	Decimal
Account Code Prefix	0x30006	196614
Alternate Station Prefix	0x30007	196615
Off Network Prefix	0x30008	196616
Speed Dial Prefix	0x30009	196617
PIN Length	0x3000a	196618
Account Code Length	0x3000b	196619
Station Length	0x3000c	196620
Off Network Call Barred	0x3000d	196621
Station Is Manager	0x3000e	196622
Restrict Calling Address	0x3000f	196623
Allow Short Extensions	0x30010	196624
Hunting List 1	0x30011	196625
Hunting List 2	0x30012	196626
Hunting List 3	0x30013	196627
Hunting List 4	0x30014	196628
Hunting List 5	0x30015	196629
Hunting List Default	0x30016	196630
Hunting To List 1	0x30017	196631
Hunting To List 2	0x30018	196632
Hunting To List 3	0x30019	196633
Hunting To List 4	0x3001a	196634
Hunting To List 5	0x3001b	196635
Hunting To List Default	0x3001c	196636
Send Identical CPN	0x3001d	196637
Match Undefined Extensions	0x3001e	196638
Hunting Configuration	0x30020	196640
Hunting Scheduling	0x30060	196704
SCI ID	0x30100	196864
SCI Data	0x30101	196865
Dialing Prefix Length	0x30200	197120
Calling On Network List	0x30310	197392
Calling On Network List Type	0x30311	197393
Calling Off Network List	0x30320	197408
Calling Off Network List Type	0x30321	197409
Calling PIN Always List	0x30330	197424
Calling PIN Always List Type	0x30331	197425
Calling PIN Never List	0x30340	197440
Calling PIN Never List Type	0x30341	197441
Called On Network LIST	0x30350	197456
Called On Network List Type	0x30351	197457
Called Off Network List	0x30360	197472
Called Off Network List Type	0x30361	197473

Description	Hex	Decimal
Called PIN Always List	0x30370	197488
Called PIN Always_List Type	0x30371	197489
Called PIN Never List	0x30380	197504
Called PIN Never List Type	0x30381	197505

Statistics

Introduction

Macro nodes are feature nodes that are used by ACS using the ACS Control Plan Editor, but are not part of the ACS product. They are supplied by other NCC applications, but require the presence of ACS for use.

Macro nodes require some configuration to be entered into the **eserv.config** file. The macro node reads the global configuration file (**eserv.config**) on initialization. Should the configuration of a macro node be changed, the configuration files must be re-read.

The following section details the configuration that is necessary for the macro node used for VPN.

Mobile analyze

This table defines the statistics generated in the VPN Mobile Analyze macro node.

Call Type	Description	Calling (cg) - Called (cd) Number Description
MOC	Mobile originating call where the calling number is 'on-net' and the called number is 'on-net'.	on-net (cg) -> on-net (cd)
MOC	Mobile originating call where the calling number is 'on-net' and the called number is 'off-net'.	on-net (cg) -> off-net (cd)
MOC	Mobile originating call where the calling number is 'on-net' and the called number is 'speed_code_on-net'.	on-net (cg) -> speed_code_on-net (cd)
MOC	Mobile originating call where the calling number is 'on-net' and the called number is 'speed_code_off-net'.	on-net (cg) -> speed_code_off-net (cd)
MFC	Mobile forwarded call where the calling number is 'on-net' and the called number is 'on-net'. The forwarded call is determined when the called number does not equal the redirecting party id.	on-net (cg) -> on-net (cd)
MFC	Mobile forwarded call where the calling number is 'on-net' and the called number is 'off-net'. The forwarded call is determined when the called number does not equal the redirecting party id.	on-net (cg) -> off-net (cd)
MFC	Mobile forwarded call where the calling number is 'on-net' and the called number is 'speed_code_on-net'. The forwarded call is determined when the called number does not equal the redirecting party id.	on-net (cg) -> speed_code_on-net (cd)

Call Type	Description	Calling (cg) - Called (cd) Number Description
MFC	Mobile forwarded call where the calling number is 'on-net' and the called number is 'speed_code_off-net'. The forwarded call is determined when the called number does not equal the redirecting party id.	on-net (cg) -> speed_code_off-net (cd)

Analyze

This table defines the statistics generated in the VPN Analyze macro node.

Call Type	Description	Calling (cg) - Called (cd) Number Description
FOC	Fixed line originating call where the calling number is 'on-net' and the called number is 'on-net'.	on-net (cg) -> on-net (cd)
FOC	Fixed line originating call where the calling number is 'on-net' and the called number is 'off-net'.	on-net (cg) -> off-net (cd)
FOC	Fixed line originating call where the calling number is 'on-net' and the called number is 'speed_code'.	on-net (cg) -> speed_code (cd)
FOC	Fixed line originating call where the calling number is 'on-net' and the called number is 'alternate_station'.	on-net (cg) -> alternate_station (cd)
FFC	Fixed line forwarded call where the calling number is 'on-net' and the called number is 'on-net'. The forwarded call is determined when the called number does not equal the redirecting party id.	on-net (cg) -> on-net (cd)
FFC	Fixed line forwarded call where the calling number is 'on-net' and the called number is 'off-net'. The forwarded call is determined when the called number does not equal the redirecting party id.	on-net (cg) -> off-net (cd)
FFC	Fixed line forwarded call where the calling number is 'on-net' and the called number is 'speed_code'. The forwarded call is determined when the called number does not equal the redirecting party id.	on-net (cg) -> speed_code (cd)
FFC	Fixed line forwarded call where the calling number is 'on-net' and the called number is 'alternate_station'. The forwarded call is determined when the called number does not equal the redirecting party id.	on-net (cg) -> alternate_station (cd)

CLI Lookup

This table defines the statistics generated in the VPN CLI Lookup macro node.

Call Type	Description	Calling (cg) - Called (cd) Number Description
MTC	Mobile terminating call where the calling number is 'on-net' and the called number is 'on-net'. The vpn_station.STATION_TYPE is determined as fixed(0) or mobile(1).	on-net (cg) -> on-net (cd)

Call Type	Description	Calling (cg) - Called (cd) Number Description
MTC	Mobile terminating call where the calling number is 'off-net' and the called number is 'off-net'. The vpn_station.STATION_TYPE is determined as fixed(0) or mobile(1).	off-net (cg) -> on-net (cd)
FTC	Fixed line terminating call where the calling number is 'on-net' and the called number is 'on-net'. The vpn_station.STATION_TYPE is determined as fixed(0) or mobile(1).	on-net (cg) -> on-net (cd)
FTC	Fixed line terminating call where the calling number is 'off-net' and the called number is 'on-net'. The vpn_station.STATION_TYPE is determined as fixed(0) or mobile(1).	off-net (cg) -> on-net (cd)

Service Loader

This table defines the statistics generated in the VPN Service Loader macro node.

Control Plan Type Trigger	Description	Calling (cg) - Called (cd) Number Description
Originating	Calling number is physical address.	physical address (cg)
Originating	Calling number is VDDI address.	VDDI address (cg)
Originating	Calling number is VPN station address.	VPN station address (cg)
Terminating	Calling number is physical address.	physical address (cd)
Terminating	Calling number is GVNS address.	GVNS address (cd)
Terminating	Calling number is VDDI address.	VDDI address (cd)
Terminating	Calling number is VPN station address.	VPN station address (cd)

EDRs

EDR tags

Here is the list of EDR tags generated by the ACS service, but used exclusively for the VPN service.

Tag	Description
CGNA	Global Calling Network Address
CPNI	Calling Private Network ID
CS	Connect Status (Status 9 only is VPN)
PCNA	Calling Private Network Address
PTNA	Private Terminating Network Address
TGNA	Global Terminating Network Address

TPNI	Terminating Private Network ID
------	--------------------------------

For a full description, refer to *Event Detail Record Reference Guide*.

About Secure SSL Connection to the Database

Enabling Secure SSL Connection to the Database

NCC supports secure network logins through Secure Socket Layer (SSL) connections from the NCC UI to the database. SSL is the default method for connecting to the database when you install NCC. You can also enable SSL after installing NCC.

For information about enabling SSL connections to the database, see *SMS Technical Guide*.

Enabling SSL for VPN

You can access VPN through the Services menu in the SMS UI or you can access it directly:

- From your internet browser by using the appropriate URL
- From a Java WebStart URL
- From the desktop or start menu by using the VPN shortcut

See *Accessing VPN* (on page 26) for more information.

If you access the VPN through the SMS UI and SSL is already enabled, no further action is required to enable SSL for VPN. For information about enabling SSL on the SMS, see *SMS Technical Guide*.

If you access the VPN directly, enable SSL connections to the database:

- Create the Oracle wallet that identifies the database server on the SMS node, and specify its location in the **listener.ora** and **sqlnet.ora** files.
- Change the **listener.ora** file to additionally listen on port 2484 by using the TCPS protocol for secure SSL connections to the database.

Note: The standard Oracle listener TCP port is 1521. However, SSL connections use the standard port for the TCPS protocol, port 2484, instead. If there is a firewall between screen clients and the SMS, open port 2484 in the firewall.

For more information about enabling SSL by configuring the Oracle wallet and updating the **listener.ora** and **sqlnet.ora** files, see *SMS Technical Guide*.

The following additional configuration must be set in the **vpnGui.bat/vpnGui.sh** file:

- The **secureConnectionDatabaseHost** Java application property (on non-clustered systems) or the **secureConnectionClusterDatabaseHost** Java application property (on clustered systems) must specify the database connection in the **CONNECT_DATA** part. In addition the **PROTOCOL** part must be set to **TCPS** and the **PORT** part must be set to **2484**.
- If present, set the **EncryptedSSLConnection** Java application property to **true**. The NCC UI connects to the database by using encrypted SSL connections by default.

Note: If you are using non-SSL connections to the database then you must set **EncryptedSSLConnection** to **false**. When **EncryptedSSLConnection** is set to **false**, the **secureConnectionDatabaseHost** and the **secureConnectionClusterDatabaseHost** properties are ignored.

See *VPN Application Properties for SSL and Non-SSL Database Connections* (on page 30) for more information.

Configuration

Overview

Introduction

This chapter explains how to configure the Oracle Communications Network Charging and Control (NCC) application.

In this chapter

This chapter contains the following topics.

<i>Configuring the SLEE.cfg</i>	<i>11</i>
<i>Configuring acs.conf for the SLC</i>	<i>12</i>
<i>eserv.config Configuration</i>	<i>15</i>
<i>VPN service loader configuration</i>	<i>16</i>
<i>VPN Specific Announcements</i>	<i>20</i>
<i>Defining the VPN Screen Labels Language.....</i>	<i>25</i>
<i>Defining the VPN Help Screen Language.....</i>	<i>25</i>
<i>Setting up the Screens</i>	<i>26</i>

Configuring the SLEE.cfg

Introduction

Apart from the `ServiceEntry` configuration in **acs.conf**, in the SLC, calls are routed to application services by using service keys. You configure service keys, service names, and application services in the **SLEE.cfg** configuration file on the SLC.

The ACS application runs all VPN services. ACS is labeled as `slee_acs` in the **SLEE.cfg**:

```
APPLICATION=slee_acs slee_acs.sh /IN/service_packages/ACS/bin 1 1 1000
```

The VPN services are attached to ACS by including their configuration entries in the **SLEE.cfg**. The structures of these entries are:

```
SERVICE=service_name priority application_name service_handle
SERVICEKEY=key_type key_value service_name
```

Service names are used to link service keys to service handles. The service handles enable the service application to distinguish between the different services it handles.

Note: In VPN, all service names have the same name as their respective service handle.

SLEE.cfg entries

The VPN installation adds the following `SERVICE` and `SERVICEKEY` entries to the **SLEE.cfg**:

```
SERVICE=VPN_Terminating 1 slee_acs VPN_Terminating
SERVICE=VPN_Originating 1 slee_acs VPN_Originating
SERVICE=VPN_Management 1 slee_acs VPN_Management
```

```
SERVICEKEY=INTEGER 112 VPN_Terminating
SERVICEKEY=INTEGER 113 VPN_Originating
SERVICEKEY=INTEGER 114 VPN_Management
```

The **SLEE.cfg** is located in `/IN/service_packages/SLEE/etc`. See *SLEE Technical Guide* for a complete description of **SLEE.cfg**

Note: You may need to change the **SERVICEKEY** numbers if there is a conflict with existing **SERVICEKEY** numbers.

Three control plans per company

You are able to set up three control plans for each company by appending a 4 character company code, in format *Type_of_callCompany_information*.

Example:

```
service key a = service key handle: VPN_OriginatingCMP1
service key a = service key handle: VPN_TerminatingCMP1
service key a = service key handle: VPN_ManagementCMP1
service key a = service key handle: VPN_OriginatingABCD
service key a = service key handle: VPN_TerminatingABCD
service key a = service key handle: VPN_ManagementABCD
```

Use of **VPN_Originating**, **VPN_Terminating**, and **VPN_Management** service key handles, as described in *SLEE.cfg entries*, does not preclude manipulation of number buffers within control plans, using nodes such as Set Buffer, Set Carrier Code, and Set Pending TN.

Configuring acs.conf for the SLC

Introduction

The Virtual Private Network (VPN) tools and processes are integrated within the Advanced Control Services (ACS) application and depend on the ACS configuration file, **acs.conf**. The **acs.conf** file is located in the `$(ACS_ROOT)/etc` directory. For more information about **acs.conf**, see *ACS Technical Guide*.

acs.conf file

The **acs.conf** configuration file consists of several sections named for the executable they control. Each section contains a name value pair representing a single configuration option. Leading '#' characters represent comments and are ignored by the system. Each section must be terminated by a ':' character.

The **acs.conf** configuration file is part of the ACS application. However, it contains some variables that are specific to VPN that are described in this topic. Any configuration options in the **acs.conf** file that are not described here are required by ACS, and should not be changed.

For more information about the **acs.conf** configuration file, see *ACS Technical Guide*.

acsChassis

Service entries are specified in the **acs.conf** file. **ServiceEntry** is a standard **acsChassis** parameter, and is described in detail in *ACS Technical Guide*.

The VPN installation adds the following **ServiceEntry** lines to **acs.conf**:

```
ServiceEntry (VPN_Originating,libvpnService.so)
ServiceEntry (VPN_Terminating,libvpnService.so)
ServiceEntry (VPN_Management,libvpnService.so)
```

Check the **acs.conf** file to ensure that these lines have been added.

acsChassis plugins

The VPN installation automatically adds entries for the VPN-specific feature nodes and chassis actions.

The following lines are added to the `acsChassis` section:

```
MacroNodePluginFile libvpnMacroNodes.so
ChassisPlugin libvpnChassisActions.so
```

VPN section

The VPN-specific section of **acs.conf** supports the following parameters.

Note: The end of each configuration section in **acs.conf** is always marked with a colon that must be appended immediately after the value of the last parameter in the section.

`SendIdenticalCallingPartyNumber`

Syntax: `SendIdenticalCallingPartyNumber 0|1`
Description: Send unchanged `CallingPartyNumber` in Connect for all VPN networks.
Type: Boolean
Optionality: Optional (default used if not set)
Allowed: 0 (false), 1 (true)
Default: 1
Notes:
Example: `SendIdenticalCallingPartyNumber 1`

`MatchUndefinedExtensions`

Syntax: `MatchUndefinedExtensions 0|1`
Description: Match undefined extension numbers.
 If this parameter is set to true (1) and the following conditions apply, the call will be treated as though it is from a station in the network. The station profile will be set to the network profile:

- A call is made from a number for which there is no entry in the `VPN_STATION` table
- The Calling Party Number starts with the site code of a VPN network
- The length of the calling party number = length of this site code + extension length for this network
- The network has the Match Undefined Extensions flag on the Edit VPN Network screen set to true

Type: Boolean
Optionality: Optional (default used if not set)
Allowed: 0 (false), 1 (true)
Default: 0
Notes: 0 means do not do this for any VPN network
Example: `MatchUndefinedExtensions 1`

`LocInfoLocNumExtensionBuffer`

Syntax: `LocInfoLocNumExtensionBuffer value`
Description: Normalize the `IDP.LocationInformation.LocationNumber` (if present) into the given extension buffer.

Type: Integer
Optionality: Optional (default used if not set)
Allowed: 0 through 9
Default: 0
Notes:
Example: `LocInfoLocNumExtensionBuffer 1`

AllowTermFromOrigKey

Syntax: `AllowTermFromOrigKey = 0|1`
Description: Look up the terminating call plan if the originating lookup fails.
Type: Boolean
Optionality: Optional (default used if not set)
Allowed: 0 (false), 1 (true)
Default: 0
Notes: 0 means do not do this for any VPN network.
Example: `AllowTermFromOrigKey 0`

overrideVPNCapabilityWithEventTypeBCSM

Syntax: `overrideVPNCapabilityWithEventTypeBCSM = true | false`
Description: Setting this parameter to true overrides the capability type for VPN calls based on the value provided in eventTypeBCSM of IDP.
Type: Boolean
Optionality: Optional
Allowed: true, false
Default: false
Notes: IDP.eventTypeBCSM == collectedInfo, sets capability type to VPN_MO
IDP.eventTypeBCSM == termAttemptAuthorized, sets capability type to VPN_MT
Example: `overrideVPNCapabilityWithEventTypeBCSM = true`

setVPNNetworkProfiletag

Syntax: `setVPNNetworkProfiletag 0|1`
Description: Enable setting the value into the profile tag CC Private Network ID.

- **0:** Disabled
- **1:** Enabled

Type: Boolean
Optionality: Optional (default used if not set)
Allowed: 0 (false), 1 (true)
Default: 0
Notes:
Example: `setVPNNetworkProfiletag 0`

Example acs.conf for VPN

Here is an example of the VPN-specific section of **acs.conf**.

```
VPN
```

```

SendIdenticalCallingPartyNumber 1
MatchUndefinedExtensions 1
LocInfoLocNumExtensionBuffer 1
AllowTermFromOrigKey 0
setVPNNetworkProfiletag 0

```

eserv.config Configuration

Introduction

The **eserv.config** file is a shared configuration file, from which many Oracle Communications Network Charging and Control (NCC) applications read their configuration. Each NCC machine (SMS, SLC, and VWS) has its own version of this configuration file, containing configuration relevant to that machine. The **eserv.config** file contains different sections; each application reads the sections of the file that contains data relevant to it.

The **eserv.config** file is located in the `/IN/service_packages/` directory.

The **eserv.config** file format uses hierarchical groupings, and most applications make use of this to divide the options into logical groupings.

Configuration File Format

To organize the configuration data within the **eserv.config** file, some sections are nested within other sections. Configuration details are opened and closed using either `{ }` or `[]`.

- *Groups of parameters are enclosed with curly brackets – `{ }`*
- *An array of parameters is enclosed in square brackets – `[]`*
- *Comments are prefaced with a `#` at the beginning of the line*

To list things within a group or an array, elements must be separated by at least one comma or at least one line break. Any of the following formats can be used, as in this example:

```

{ name="route6", id = 3, prefixes = [ "00000148", "0000473" ] }
{ name="route7", id = 4, prefixes = [ "000001049" ] }

```

or

```

{ name="route6"
  id = 3
  prefixes = [
    "00000148"
    "0000473"
  ]
}
{ name="route7"
  id = 4
  prefixes = [
    "000001049"
  ]
}

```

or

```

{ name="route6"
  id = 3
  prefixes = [ "00000148", "0000473" ]
}
{ name="route7", id = 4
  prefixes = [ "000001049" ]
}

```

eserv.config Files Delivered

Most applications come with an example **eserv.config** configuration in a file called **eserv.config.example** in the root of the application directory, for example, `/IN/service_packages/eserv.config.example`.

Editing the File

Open the configuration file on your system using a standard text editor. Do not use text editors, such as Microsoft Word, that attach control characters. These can be, for example, Microsoft DOS or Windows line termination characters (for example, ^M), which are not visible to the user, at the end of each row. This causes file errors when the application tries to read the configuration file.

Always keep a backup of your file before making any changes to it. This ensures you have a working copy to which you can return.

Loading eserv.config Changes

If you change the configuration file, you must restart the appropriate parts of the service to enable the new options to take effect.

Restarting

The following processes are restarted after configuration changes as indicated.

Process	Restart Action
slee_acs	Restart of slee_acs or restart of the SLEE. For more information about how to restart the SLEE, see <i>SLEE Technical Guide</i> .

VPN service loader configuration

Introduction

libvpnService is configured in the VPN section of the **eserv.config** file. When the service loader is started, it checks the **eserv.config** file and loads the plug-ins specified in the PLUGINS array.

Note: It is not recommended to change the values of this section. Please contact Oracle before modifying this configuration.

Parameters

Here are the VPN service loader configuration parameters.

stripPrefix

Syntax:	<code>stripPrefix = true false</code>
Description:	Use the Speed Dial Prefix or Off Net Prefix profiles to strip the prefix before matching on the terminating number.
Type:	Boolean
Optionality:	Optional (default used if not set)
Allowed:	<ul style="list-style-type: none"> <code>true</code> – Use prefix stripping <code>false</code> – Do not alter the prefix
Default:	false
Notes:	Used when checking for station and network speed dials and for on-net to off-net calls.

Example: `stripPrefix = true`

vpnPluginExtend

The `vpnPluginExtend` section of the VPN service loader configuration configures additional plug-in libraries, and enables the service loader to use the additional functionality provided.

These are the `vpnPluginExtend` parameters:

SERVICE_HANDLE

Syntax: `SERVICE_HANDLE = "name"`

Description: Calls using this SLEE service handle will be processed using the configuration in this `Services` section.

Type: String

Optionality: Optional (default used if not set)

Allowed:

Default: Empty string

Notes: When a new call is presented to the application, the service handle indicates the particular service for which the call is intended.
If this string does not match a service handle in `SLEE.cfg`, no calls will be processed using this service configuration.

Example: `SERVICE_HANDLE = "VPN_Originating_V"`
`SERVICE_HANDLE = "EAX_MO"`

PLUGINS

Syntax: `PLUGINS = [`
 `{LIBRARY = lib_config}`
 `...`
`]`

Description: List of plug-ins to load after loading the main service.

Type: Array

Optionality: Optional (no plug-ins used if not set)

Allowed:

Default: None

Notes: A plug-in will be loaded only if it is configured here. Therefore, this parameter must include any plug-ins that are needed to supply application functions.
Applies to calls that use the service handle defined in `SERVICE_HANDLE` (on page 17).
An example of a plug-in library is the MM plug-in, which is used to extract SMS information for `slee_acs`.

Example:

LIBRARY

Syntax: `LIBRARY =`
 `"lib,acsChassisFn,call_acsChassis_fn_with_null,acsPrePORFn"`

Description: Configuration for the service plug-ins.

Type: Array

Optionality: Optional

Allowed:

Default:

Notes:

Example: `LIBRARY =
"libxmsSvcLibCallContext.so,xmsSvcLibCallContextCreate,False
,xmsSvcLibCallContextExtend"`

`lib`

Syntax: `lib.so`

Description: The filename of the service loader plug-in library.

Type: String

Optionality: Required (must be present to load the plug-in)

Allowed:

Default: None

Notes: This is the first parameter in the *LIBRARY* (on page 17) parameter array.

Example: `libChargingPlugin.so`

`acsChassisFn`

Syntax: `acsChassisFn`

Description: Name of the `acsChassisLoadService` function used to create a call context to the required application.

Type: String

Optionality: Required

Allowed: Any valid function name

Default:

Notes: This is the second parameter in the *LIBRARY* (on page 17) parameter array.
For more information about the functions which can be used with a library, see the technical guide for the application that provides the plug-in library.

Example: `xmsSvcLibCallContextCreate`

`call_acsChassis_fn_with_null`

Syntax: `call_acsChassis_fn_with_null`

Description: Defines whether the `acsChassisLoadService` function should accept a null value as input or not.

Type: Boolean

Optionality: Required

Allowed:

- *True – Accept null*
- *False – Do not accept null*

For more information about whether to set this parameter to False or True, see the technical guide for the application that provides the plug-in library.

Default:

Notes: This value only controls whether the plug-in `acsChassisLoadService` function should be called with a null value parameter or not.
If a non-null value is passed, the VPN service loader will always call the `acsChassesLoadService` function.

This is the third parameter in the *LIBRARY* (on page 17) parameter array.

Example: `False`

acsPrePORFn

Syntax:	<i>acsPrePORFn</i>
Description:	Name of the acsPrePORFn function used to extend the MM call context.
Type:	String
Optionality:	Required
Allowed:	<ul style="list-style-type: none"> • <i>function name</i> – The name of the library's function • <i>none</i> – Do not call a PrePOR function <p>For more information about the functions which can be used with a library, see the technical guide for the application which provides the plug-in library.</p>
Default:	
Notes:	This is the second parameter in the <i>LIBRARY</i> (on page 17) parameter array.
Example:	xmsSvcLibCallContextExtend

Example configuration

Here is an example of the VPN section in the **eserv.config**.

```
VPN = {
    StripPrefix = true
    vpnPluginExtend = [
        {
            SERVICE_HANDLE = "VPN_Originating"
            PLUGINS = [
                {LIBRARY = "libeaxChargingPlugin.so,createEAXCallContext,True,None"}
                {LIBRARY =
                    "libxmsSvcLibCallContext.so,xmsSvcLibCallContextCreate,False,xmsSvcLi
                    bCallContextExtend"}
            ]
        }
        {
            SERVICE_HANDLE = "VPN_Originating_V"
            PLUGINS = [
                {LIBRARY = "libeaxChargingPlugin.so,createEAXCallContext,True,None"}
            ]
        }
        {
            SERVICE_HANDLE = "VPN_Originating_M"
            PLUGINS = [
                {LIBRARY =
                    "libxmsSvcLibCallContext.so,xmsSvcLibCallContextCreate,False,xmsSvcLi
                    bCallContextExtend"}
                {LIBRARY = "libeaxChargingPlugin.so,createEAXCallContext,True,None"}
            ]
        }
        {
            SERVICE_HANDLE = "VPN_Terminating_V"
            PLUGINS = [
                {LIBRARY = "libeaxChargingPlugin.so,createEAXCallContext,True,None"}
            ]
        }
        {
            SERVICE_HANDLE = "VPN_Terminating_M"
            PLUGINS = [
                {LIBRARY = "libeaxChargingPlugin.so,createEAXCallContext,True,None"}
                {LIBRARY =
                    "libxmsSvcLibCallContext.so,xmsSvcLibCallContextCreate,False,xmsSvcLi
                    bCallContextExtend"}
            ]
        }
    ]
}
```

```

    ]
  }
  {
    SERVICE_HANDLE = "VPN_Terminating"
    PLUGINS = [
      {LIBRARY = "libeaxChargingPlugin.so,createEAXCallContext,True,None"}
      {LIBRARY =
        "libxmsSvcLibCallContext.so,xmsSvcLibCallContextCreate,False,xmsSvcLi
        bCallContextExtend"}
    ]
  }
  {
    SERVICE_HANDLE = "VPN_Management_V"
    PLUGINS = [
      {LIBRARY = "libeaxChargingPlugin.so,createEAXCallContext,True,None"}
    ]
  }
  {
    SERVICE_HANDLE = "VPN_Management_M"
    PLUGINS = [
      {LIBRARY = "libeaxChargingPlugin.so,createEAXCallContext,True,None"}
      {LIBRARY =
        "libxmsSvcLibCallContext.so,xmsSvcLibCallContextCreate,False,xmsSvcLi
        bCallContextExtend"}
    ]
  }
  {
    SERVICE_HANDLE = "VPN_Management"
    PLUGINS = [
      {LIBRARY = "libeaxChargingPlugin.so,createEAXCallContext,True,None"}
      {LIBRARY =
        "libxmsSvcLibCallContext.so,xmsSvcLibCallContextCreate,False,xmsSvcLi
        bCallContextExtend"}
    ]
  }
}
]
}

```

VPN Specific Announcements

Introduction

Before the VPN control plans can access and use the VPN announcements, the announcements must be recorded on an IP available to the network, and the IP announcement IDs must be entered into the **Announcements** tab on the ACS Configuration screen.

All announcements for each language must have correctly mapped IP announcements IDs for that language.

Originating announcements

Here is the list of announcements used by the VPN Originating control plan. You can customize these announcements if required.

Announcement name and Announcement	Diagram Code
Originating Decode Fail "Cannot decode entered digits, or speed dial out of valid range"	101
Originating Invalid Alternate Station ID "The alternate [roaming] station address you specified in your dialing string does not exist on this network"	102

Announcement name and Announcement	Diagram Code
Originating Roaming Not Allowed "The alternate [roaming] station address you specified in your dialing string is not authorized to use the roaming feature"	103
Originating Unknown Speed Dial "The station or network speed dial dialed is not defined for your station or network"	104
Originating Cannot Complete On-Net "The dialed on-net address does not exist on your network"	105
Originating OffNet Access Barred "This station has barred off-net access, and the user does not have permission to override, or could not enter the PIN for this station after multiple attempts"	106
Originating Number is Barred "The dialed off-net address is barred, or not allowed for this station or this network"	107
Originating Invalid Account Code "A correct account code was not entered after multiple attempts"	108
Originating Chain Count Exceeded "I'm Sorry, your call cannot be processed due to a system error"	109
Note: Too many chained call plans - this should not be possible.	
Originating Roaming PIN Required "A PIN is required for station roaming, please enter it now"	110
Originating Roaming PIN Wrong "Your PIN is not correct, please re-enter it now"	111
Originating Call Bar Override PIN Required "A PIN is required in order to override the off-net call bar on this station, please enter it now"	112
Originating Call Bar Override PIN Wrong "Your PIN is not correct, please re-enter it now"	113
Originating Account Code Required "A PIN is required for station roaming"	114
Originating Account Code Reprompt "The account code you entered is not valid, please enter a valid code"	115

Terminating announcements

Here is the list of announcements used by the VPN Terminating control plan. You can customize these announcements, if required.

Announcement name & Announcement	Diagram Code
Terminating Manager Unavailable "I'm sorry, your call cannot be processed due to a system error" Note: This probably indicates that the management call plan has been deleted, or not correctly built. If you wish to remove Management functionality, you should remove the ServiceHandover node in the Terminating call plan.	201
Terminating Incoming Barred "This station cannot receive calls from your location"	202
Terminating Invalid Forward Number "This station has been forwarded to another station, however the remote station is not valid"	203
Terminating Too Many Forwards "A call forwarding loop has been detected" Note: A call forwarding loop is the only realistic explanation for this event.	204

Management announcements

Here is the list of announcements used by the VPN Management control plan. You can customize these announcements if required.

Announcement name & Announcement	Diagram Code
Management Station ID Required "Please enter your station ID" Note: You only see this prompt if the call is received from off-net. Calls from on-net will not need to prompt for a caller station ID.	301
Management Station ID Reprompt "That is not a known station, please enter your station ID" Note: Only required for manager calls from off-net.	302
Management Authorization Required "A PIN is required to use the management functions, please enter your PIN now" Note: PIN entry is not required if the user is on-net, and has previously entered a PIN.	303
Management Authorization Reprompt "Your PIN is not correct, please re-enter it now" Note: Only required for manager calls from off-net.	304

Announcement name & Announcement	Diagram Code
Management Main Menu Prompt "Please select a function. Press "1" to store a speed code Press "2" to set your forwarding address for Network Failure Press "5" to set your follow me address, all calls will be forwarded to the specified destination. Press "6" to change your PIN Press "9" to make a call now Press "0" to hear a more detailed help message"	305
Management Main Menu Bad Digit "I'm Sorry, but that option is not known"	306
Management Goodbye "Thank you, have a nice day"	307
Management Not Allowed "I'm sorry, this station is not authorized to use the station management feature. Please contact your network administrator".	308
Management Main Please make a selection from the list (Same as 305, except read more slowly, and LOUDER!)	309
Management Feature Prohibited "I'm Sorry, you are not authorized to use this feature" Note: This is played if the user fails to enter a PIN at this point (if they haven't already entered one), or if the user does not have the appropriate permission flag.	310
Management Destination Number Prompt "Enter the number you wish to dial. This may include an off-net prefix"	311
Management Destination Number Reprompt "Re-prompt for destination. Input time-out, too few, too many digits"	312
Management Invalid Address "The destination number you wish to store is not valid, it must be an on-net address, or an off-net number with the off-net prefix" Note: Speed dials, account codes, PINs and alternate stations are not permitted.	313
Management Processing Failed "I'm sorry, your call cannot be processed due to a system error" Note: This should not happen under normal circumstances.	317
Management Alternate Prompt "Please enter the new destination number to store" Note: This announcement is used for both Speed Dial and Forwarding numbers. The announcement is very poorly named.	321
Management Alternate Reprompt "Invalid entry, please enter the new destination number to store" Note: Timeout or insufficient digits. This announcement is used for both Speed Dial and Forwarding numbers. The announcement is very poorly named.	322

Announcement name & Announcement	Diagram Code
Management Invalid Destination "The destination number you wish to store is not valid, it must be an on-net address, or an off-net number with the off-net prefix" Note: Speed dials, account codes, PINs and alternate stations are not permitted.	323
Management Cannot Forward "I'm sorry, the entered destination is not a permitted forwarding destination" Note: This could fail on either the network or station Barred/Allowed list.	324
Management Cannot Store "I'm sorry, the number you entered cannot be saved"	325
Management Update Confirmed "Your station profile has been changed successfully"	326
Management Speed Code Prompt "Please enter the Speed Dial Code you wish to update, in the range 100-199"	327
Management Speed Code Reprompt "That Speed Dial Code is not in the valid range 100-199, please enter the Speed Dial Code you wish to update"	328
Management PIN Entry "Please enter your new PIN"	329
Management PIN Entry Reprompt "Invalid number of digits, please re-enter your new PIN"	330
Management Unknown Station "The number you have entered as your station ID is not known on this network" Note: This message is played after all permitted (possibly multiple) attempts to enter your station address have failed.	331
Management PIN Required "A PIN is required to access this feature, please enter your PIN"	332
Management PIN Required Reprompt "Your PIN is not correct, please re-enter it now"	333

Non control plan specific announcements

Here is the list of announcements that may be used by any control plan. You can customize these announcements if required.

Announcement name & Announcement	Diagram Code
Collect PIN Introduction "A PIN is required to access this feature, please enter your PIN"	401
Invalid PIN Entered "Your PIN is not correct, please re-enter it now"	402
Maximum Allowed PIN Attempts Reached "You have reached the maximum number of PIN attempts for this feature"	403
Silence Announcement "Your PIN is not correct, please re-enter it now" Note: This is an empty announcement.	404

Note

These announcement mappings are not installed automatically because in most cases, the required values are customer specific.

If you wish to configure these values, you can enter them manually in the ACS or VPN announcement screens or you can run the provided configuration script as `acs_oper`.

```
/IN/service_packages/ACS/db/install/install_vpn/vpn_language_mappings.sh
```

Defining the VPN Screen Labels Language

Introduction

The system administrator uses a default language file to define the "default" language of the system. This determines what language is used to display the login screen.

A file called **Default.lang** is created in the language directory. This contains a soft-link to the actual language file.

Example:

If Russian is the default language, then create a soft-link from the **Default.lang** file to the **Russian.lang** file.

If a **Default.lang** file is not specified or has not been created, then **English.lang** is used.

Once a **Default.lang** file is created, a user must explicitly set their language to English (or other required language), otherwise the default language is used.

Procedure

Follow these steps to modify the default language file.

Step	Action
1	Go to the <code>/IN/html/Vpn_Service/language</code> directory.
2	Is the Default.lang soft-link resident? If Yes, enter the following to determine which language file it points to: <pre>ls -l</pre> Note: The default is "English.lang".
3	Do you wish to modify this? If Yes, enter the following: <pre>rm Default.lang</pre>
4	Replace it with the following command: <pre>ln -s language.lang Default.lang</pre> where <i>language</i> is your language.

Defining the VPN Help Screen Language

Introduction

You define the default language of the help system in the default helpset file, **Default_service_name.hs**.

Procedure

Follow these steps to create a default helpset file.

Step	Action
1	Go to the <code>/IN/html/Vpn_Service/helptext/</code> directory.
2	Ensure that the required file is resident: <code>language_servicename.hs</code> Where: <ul style="list-style-type: none"> • <code>language</code> is your language • <code>servicename</code> is the name of the service you are installing Example: Russian_vpn_service.hs
3	Enter the following command: <code>ln -s language_servicename.hs Default_servicename.hs</code>

Setting up the Screens

Accessing VPN

To access the VPN user interface (UI), do the following:

- Ensure the Java SE Runtime Environment version 21 is installed on your machine.
- If required, obtain, and install the trusted certificate for the database connection into your keystore.
- Obtain the application zip file containing jars and other files (**vpnGui.bat** or **vpnGui.sh**).
- In Windows, run **vpnGui.bat** to start the application.
- In other machines:
 - Change the permission of **vpnGui.sh** using `chmod 755 vpnGui.sh` command.
 - Run the application using `bash vpnGui.sh` command.

For more information about the VPN UI, see *VPN User's Guide*.

About customizing the UI

You can customize the VPN UI by setting application properties in the **vpnGui.bat/vpnGui.sh** file, which is located in the `/IN/html/` directory. You set Java application properties by using the following syntax:

```
-Dproperty="value"
```

Where:

- `property` is the name of the Java application property
- `value` is the value of the specified property

Java Application Properties

The following application properties are available to customize the UI:

```
jnlp.sms.host
```

Syntax: `-Djnlp.sms.host="IPaddress"`

Description: Specifies the Internet Protocol (IP) address for the SMS host machine that is set at installation.

Type: String

Optionality: Required

Allowed:

- *IP version 4 (IPv4) addresses*
- *IP version 6 (IPv6) addresses*

Default: No default

Notes: You can use the industry standard for omitting zeros when specifying IP addresses.

Examples:

```
-Djnlp.sms.host="192.0.2.0"
-Djnlp.sms.host="2001:db8:0000:1050:0005:0600:300c:326b"
-Djnlp.sms.host="2001:db8:0:0:0:500:300a:326f"
-Djnlp.sms.host="2001:db8::c3"
```

jnlp.sms.port

Syntax: `-Djnlp.sms.port="num"`

Description: Specifies the SQL*Net port for connecting to the SMS host machine.

Type: Integer

Optionality: Optional (default used if not set)

Allowed:

Default: 1521

Notes: Set at installation

Example: `-Djnlp.sms.port="1521"`

jnlp.sms.database

Syntax: `-Djnlp.sms.database="SMF"`

Description: Specifies the Oracle SID for the SMF database.

Type: String

Optionality: Optional (default used if not set)

Allowed:

Default: SMF

Notes: Set at installation.

Example: `-Djnlp.sms.database="SMF"`

jnlp.vpn.INProtocol

Syntax: `-Djnlp.vpn.INProtocol="name"`

Description: Specifies the IN protocol for VPN screens.

Type: String

Optionality: Required

Allowed:

- *AIN – Hides settings not relevant to AIN. Only customers using Advanced Intelligent Network (AIN) should set the property to AIN.*
- *Not set – All settings are shown.*

Default: Not set

Notes: Set at installation.

Example: `-Djnlp.vpn.INProtocol="AIN"`

jnlp.sms.dbUser

Syntax:	-Djnlp.sms.dbUser="user"
Description:	Specifies the database user name. This is a special database user that the ACS Logon screen uses before the user logs in. This property is set during installation and is then not changed.
Type:	String
Optionality:	Optional (default used if not set)
Allowed:	
Default:	acs_public
Notes:	Do not change this value.
Example:	-Djnlp.sms.dbUser="acs_public"

jnlp.sms.dbPassword

Syntax:	-Djnlp.sms.dbPassword="password"
Description:	Specifies the database password. This password is for a special database user that the ACS Logon screen uses before the user logs in. This property is set during installation and is then not changed.
Type:	String
Optionality:	Optional (default used if not set)
Allowed:	
Default:	acs_public
Notes:	Do not change this value.
Example:	-Djnlp.sms.dbPassword="acs_public"

JNLP Resources and Application Properties

Here is an example of the `vpnGui.bat/vpnGui.sh` file showing the application property settings at installation.

vpnGui.bat

```
@echo off

REM Copyright (c) 2025 Oracle. All rights reserved.
REM
REM This material is the confidential property of Oracle Corporation or its
REM licensors and may be used, reproduced, stored or transmitted only in
REM accordance with a valid Oracle license or sublicense agreement.
REM
REM
REM vpnGui.bat: runs the VPN application
REM Run: execute vpnGui.bat

REM Set JAR_PATH to the current directory
set JAR_PATH=%~dp0

REM Enable delayed expansion
setlocal enabledelayedexpansion

REM Set classpath
```

```

set
CLASSPATH=%JAR_PATH%;%JAR_PATH%sms.jar.sig;%JAR_PATH%vpn.jar.sig;%JAR_PATH%acs.jar.s
ig;%JAR_PATH%common.jar.sig;%JAR_PATH%ojdbc11.jar.sig;%JAR_PATH%oraclepki.jar.sig;%J
AR_PATH%ohj.jar.sig;%JAR_PATH%help-
share.jar.sig;%JAR_PATH%oracle_ice.jar.sig;%JAR_PATH%jwt.jar.sig;%JAR_PATH%share.ja
r.sig;

REM Starting GUI....
java ^
    -Djava.util.Arrays.useLegacyMergeSort=true ^
    -Djnlp.packEnabled=true ^
    -Djnlp.sms.host=OUI_HOSTNAME ^
    -Djnlp.sms.port=LPORT ^
    -Djnlp.sms.database=OUI_ORACLE_SID ^
    -Djnlp.sms.databaseHost=NCC_DBHOST:LPORT:OUI_ORACLE_SID ^
    -Djnlp.sms.secureConnectionDatabaseHost="(DESCRIPTION= (ADDRESS_LIST=
(ADDRESS=(PROTOCOL=TCPS) (HOST=NCC_DBHOST) (PORT=SECPORT))) (CONNECT_DATA=
(SERVICE_NAME=OUI_ORACLE_SID)))" ^
    -Djnlp.sms.EncryptedSSLConnection=true ^
    -Djnlp.sms.sslCipherSuites=(TLS_RSA_WITH_AES_128_CBC_SHA) ^
    -Djnlp.vpn.INProtocol=IN_PROTOCOL ^
    -cp "%CLASSPATH%" ^
com.g8labs.vpn.VpnApplication

REM End delayed expansion
Endlocal

```

vpnGui.sh

```

#!/bin/bash

#####
#
# Copyright (c) 2025 Oracle. All rights reserved.
#
# This material is the confidential property of Oracle Corporation or its
# licensors and may be used, reproduced, stored or transmitted only in
# accordance with a valid Oracle license or sublicense agreement.
#
#
# vpnGui.sh : runs the VPN application
# Run: bash vpnGui.sh
#
#####

JAR_PATH=$(pwd)

CLASSPATH=$JAR_PATH/:$JAR_PATH/sms.jar.sig:\
$JAR_PATH/vpn.jar.sig:\
$JAR_PATH/acs.jar.sig:\
$JAR_PATH/common.jar.sig:\
$JAR_PATH/ojdbc11.jar.sig:\
$JAR_PATH/oraclepki.jar.sig:\
$JAR_PATH/ohj.jar.sig:\
$JAR_PATH/help-share.jar.sig:\
$JAR_PATH/oracle_ice.jar.sig:\
$JAR_PATH/jwt.jar.sig:\
$JAR_PATH/share.jar.sig:

```

```

echo "Starting GUI...."
exec ${JAVA_HOME}/bin/java \
-Djnlp.sms.host=OUI_HOSTNAME \
-Djnlp.sms.port=LPORT \
-Djnlp.sms.database=OUI_ORACLE_SID \
-Djnlp.sms.databaseHost=NCC_DBHOST:LPORT:OUI_ORACLE_SID \
-Djnlp.sms.secureConnectionDatabaseHost="(DESCRIPTION= (ADDRESS_LIST=
(ADDRESS=(PROTOCOL=TCPS) (HOST=NCC_DBHOST) (PORT=SECPORT))) (CONNECT_DATA=
(SERVICE_NAME=OUI_ORACLE_SID)))" \
-Djnlp.sms.EncryptedSSLConnection=true \
-Djnlp.sms.sslCipherSuites=(TLS_RSA_WITH_AES_128_CBC_SHA) \
-Djnlp.vpn.INProtocol=IN_PROTOCOL \
-cp "$CLASSPATH" \
com.g8labs.vpn.VpnApplication

```

VPN Application Properties for SSL and Non-SSL Database Connections

The following Java application properties in the `vpn.jnlp` file are used for SSL and non-SSL connections to the database:

`jnlp.sms.databaseHost`

Syntax:	<code>-Djnlp.sms.databaseHost = "ip:port:sid"</code>
Description:	<p>Sets the IP address and port to use for non-SSL connections to the SMF database, and the database SID.</p> <ul style="list-style-type: none"> • <i>To use non-SSL connections to the database, set port to 1524 and the <code>jnlp.sms.EncryptedSSLConnection</code> property to false.</i> • <i>To use SSL connections to the database, set the <code>jnlp.sms.EncryptedSSLConnection</code> property to true and set either the <code>jnlp.sms.secureConnectionDatabaseHost</code> property or the <code>jnlp.sms.secureConnectionClusterDatabaseHost</code> property appropriately. When the <code>jnlp.sms.EncryptedSSLConnection</code> property is set to true or is undefined, <code>jnlp.sms.databaseHost</code> is ignored.</i>
Type:	String
Optionality:	Optional
Allowed:	
Default:	Not set. Secure SSL connection is enabled at installation by default.
Notes:	Internet Protocol version 6 (IPv6) addresses must be enclosed in square brackets []; for example: <code>[2001:db8:n:n:n:n:n:n]</code> where <i>n</i> is a group of 4 hexadecimal digits. The industry standard for omitting zeros is also allowed when specifying IP addresses.
Examples:	<pre> -Djnlp.sms.databaseHost = "192.0.2.1:2484:SMF" -Djnlp.sms.databaseHost = "[2001:db8:0000:1050:0005:0600:300c:326b]:2484:SMF" -Djnlp.sms.databaseHost = "[2001:db8:0:0:0:500:300a:326f]:2484:SMF" -Djnlp.sms.databaseHost = "[2001:db8::c3]:2484:SMF" </pre>

`jnlp.sms.databaseID`

Syntax:	<code>-Djnlp.sms.databaseID="port:sid"</code>
Description:	Specifies the SQL*Net port for connecting to the database, and the database SID.
Type:	String
Optionality:	Required

Allowed:

Default: 1521:SMF

Notes:

- To use non-SSL connections to the database, set port to 1521 and the `jnlp.sms.EncryptedSSLConnection` property to false.
- To use SSL connections to the database, set the `jnlp.sms.EncryptedSSLConnection` property to true and set either the `jnlp.sms.secureConnectionDatabaseHost` property or the `jnlp.sms.secureConnectionClusterDatabaseHost` property appropriately. When the `jnlp.sms.EncryptedSSLConnection` property is set to true or is undefined, `jnlp.sms.databaseID` is ignored.

Example: `-Djnlp.sms.databaseID="1521:SMF"`

`jnlp.sms.clusterDatabaseHost`

Syntax: `-Djnlp.sms.clusterDatabaseHost = "(DESCRIPTION=(LOAD_BALANCE=YES) (FAILOVER=ON) (ENABLE=BROKEN) (ADDRESS_LIST=(ADDRESS=(PROTOCOL=type) (HOST=name) (PORT=port)) (ADDRESS=(PROTOCOL=type) (HOST=name) (PORT=port))) (CONNECT_DATA=(SERVICE_NAME=SMF) (FAILOVER_MODE=(TYPE=SESSION) (METHOD=BASIC) (RETRIES=5) (DELAY=3)))"`

Description: Specifies the connection string (including a host and an alternative host address, in case the first IP address is unavailable) for non-SSL cluster-aware connection to the database.

To use non-SSL connections to the database, set the `jnlp.sms.EncryptedSSLConnection` property to false.

Type: String

Optionality: Optional

Allowed:

Default: By default, `port` is set to 1521.

Notes: If present, this property is used instead of the `jnlp.sms.databaseID` property.

Example: `-Djnlp.sms.clusterDatabaseHost = "(DESCRIPTION=(LOAD_BALANCE=YES) (FAILOVER=ON) (ENABLE=BROKEN) (ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP) (HOST=smsphysnode1) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=smsphysnode2) (PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=SMF) (FAILOVER_MODE=(TYPE=SESSION) (METHOD=BASIC) (RETRIES=5) (DELAY=3)))"`

`jnlp.sms.EncryptedSSLConnection`

Syntax: `-Djnlp.sms.EncryptedSSLConnection = "value"`

Description: Specifies whether connections to the client UI use encrypted SSL.

Type: Boolean

Optionality: Optional (default used if not set)

Allowed: true – Use encrypted SSL connections to access the client UI.
false – Use non-SSL connections to access the client UI.

Default: true

- Notes:**
- To use SSL connections to the database, set the `jnlp.sms.EncryptedSSLConnection` property to true and set either the `jnlp.sms.secureConnectionDatabaseHost` property or the `jnlp.sms.secureConnectionClusterDatabaseHost` property appropriately.
 - To use non-SSL connections to the database, set the `jnlp.sms.EncryptedSSLConnection` property to false.

Example: `-Djnlp.sms.EncryptedSSLConnection = "true"`

`jnlp.sms.port`

Syntax: `-Djnlp.sms.port="num"`

Description: Specifies the SQL*Net port for connecting to the SMS host machine.

Type: Integer

Optionality: Optional (default used if not set)

Allowed:

Default: 1521

Notes: Set at installation

Example: `-Djnlp.sms.port="1521"`

`jnlp.sms.sslCipherSuites`

Syntax: `-Djnlp.sms.sslCipherSuites="(TLS_RSA_WITH_AES_128_CBC_SHA)"`

Description: Specifies the cipher suites to use for SSL encryption. You must set this property if you are using encrypted SSL for connecting to the SMS database.

Type: String

Optionality: Optional (default used if not set)

Allowed: (TLS_RSA_WITH_AES_128_CBC_SHA)

Default: (TLS_RSA_WITH_AES_128_CBC_SHA)

Notes: You must also set the SSL_CIPHER_SUITES property to (TLS_RSA_WITH_AES_128_CBC_SHA) in the `listener.ora` and `sqlnet.ora` files.

Example: `-Djnlp.sms.sslCipherSuites="(TLS_RSA_WITH_AES_128_CBC_SHA)"`

`jnlp.sms.secureConnectionDatabaseHost`

Syntax: `-Djnlp.sms.secureConnectionDatabaseHost = "(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=type) (HOST=IPaddress) (PORT=port)))) (CONNECT_DATA=(SERVICE_NAME=servicename))"`

Description: Specifies the connection string (including host address and port) for encrypted SSL connections to the SMF database on a non-clustered system.

To use SSL connections to the database, set `port` to 2484 and set the `jnlp.sms.EncryptedSSLConnection` property to true.

Type: String

Optionality: Optional (default used if not set)

Allowed:

Default:

Notes: If present, this property is used instead of the `jnlp.sms.databaseID` property.

Example: `-Djnlp.sms.secureConnectionDatabaseHost = "(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCPS) (HOST=192.0.1.1) (PORT=2484)))) (CONNECT_DATA=(SERVICE_NAME=SMF))"`

`jnlp.sms.secureConnectionClusterDatabaseHost`

Syntax:	<code>-Djnlp.sms.secureConnectionClusterDatabaseHost = "(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=<i>type</i>) (HOST=<i>IPaddress</i>) (PORT=<i>port</i>)) (ADDRESS=(PROTOCOL=<i>type</i>) (HOST=<i>IPaddress</i>) (PORT=<i>port</i>))) (CONNECT_DATA=(SERVICE_NAME=<i>servicename</i>)))"</code>
Description:	Specifies the connection string (including host address and port) for encrypted SSL connections to the SMF database on a clustered system. To enable secure SSL connections to the database, set <i>port</i> to 2484 and set the <code>jnlp.sms.EncryptedSSLConnection</code> property to true.
Type:	String
Optionality:	Optional (default used if not set)
Allowed:	
Default:	
Notes:	If present, this property is used instead of the <code>jnlp.sms.secureConnectionDatabaseHost</code> property.
Example:	<code>-Djnlp.sms.secureConnectionClusterDatabaseHost = "(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCPS) (HOST=192.0.1.1) (PORT=2484)) (ADDRESS=(PROTOCOL=TCP) (HOST=192.0.2.1) (PORT=2484))) (CONNECT_DATA=(SERVICE_NAME=SMF)))"</code>

`jnlp.trace`

Syntax:	<code>-Djnlp.trace="value"</code>
Description:	Specifies whether to enable tracing for the Control Plan Editor. The output is displayed in the Java Console.
Type:	Boolean
Optionality:	Optional (default used if not set)
Allowed:	on off, true false, yes no, 1 0, enabled disabled
Default:	Off
Notes:	
Example:	<code>-Djnlp.trace="on"</code>

`jnlp.vpn.vpnMaxNumOfHL`

Syntax:	<code>-Djnlp.vpn.vpnMaxNumOfHL="num"</code>
Description:	Specifies the maximum number of hunting lists per station.
Type:	Integer
Optionality:	Optional (default used if not set)
Allowed:	
Default:	10
Notes:	A hunting list is a terminating call feature where a subscriber may request a list of alternate destination addresses. If their mobile station is not attached, or does not answer a call, the service logic attempts to reach the supplied alternate destinations in sequence.
Example:	<code>-Djnlp.vpn.vpnMaxNumOfHL="15"</code>

`jnlp.vpn.vpnMaxNumOfHLEntries`

Syntax:	<code>-Djnlp.vpn.vpnMaxNumOfHLEntries="num"</code>
Description:	Specifies the maximum number of entries in a hunting list.
Type:	Integer
Optionality:	Optional (default used if not set)
Allowed:	
Default:	20
Notes:	
Example:	<code>-Djnlp.vpn.vpnMaxNumOfHLEntries="25"</code>

Background Processes

Overview

Introduction

This chapter explains the processes that are started automatically by Service Logic Execution Environment (SLEE).

Note: This chapter also includes some plug-ins to background processes which do not run independently.

In this chapter

This chapter contains the following topics.

<i>libvpnService</i>	35
<i>libvpnMacroNodes</i>	36
<i>libvpnChassisActions</i>	36

libvpnService

Purpose

libvpnService is the VPN service library plug-in for `slee_acs` which handles initial set up of VPN calls. It:

- *Sets up VPN call processing (including populating the call context from the IDP)*
- *Uses the configuration to determine the correct control plan to load and run from cache*

Location

This library is located on SLCs.

Startup

If libvpnService is configured in **acs.conf**, it is made available to `slee_acs` when `slee_acs` is initialized. It is included in the `acsChassis` section of **acs.conf** by the `ServiceEntry` property(s).

For more information ServiceEntries for libvpnService in **acs.conf**, see *acsChassis* (on page 12).

Configuration

libvpnService supports the global VPN configuration in **acs.conf**. For more information about this configuration, see *VPN section* (on page 13).

libvpnService is also configured in the `VPN` section of **eserv.config**. For more information about this configuration, see *VPN service loader configuration* (on page 16).

libvpnMacroNodes

Purpose

This `slee_acs` plug-in provides the VPN macro nodes. There are no configuration file settings for these macro nodes, as they are all configured in the Control Plan Editor node configuration screens.

For more information about:

- *The Nodes*, see VPN User's Guide
- *Macro node libraries*, see ACS Technical Guide
- *CPE*, see CPE User's Guide

Location

This library is located on SLCs.

Startup

If `libvpnMacroNodes` is configured in **acs.conf**, it is made available to `slee_acs` when `slee_acs` is initialized. It is included in the `acsChassis` section of **acs.conf** in a `MacroNodePluginFile` entry as follows:

```
acsChassis
MacroNodePluginFile libvpnMacroNodes.so
```

For more information about this entry in **acs.conf**, see *acsChassis plugins* (on page 13).

Configuration

`libvpnChassisActions` are configured in **acs.conf**. For more information about configuring the VPN chassis actions, see *VPN section* (on page 13).

libvpnChassisActions

Purpose

This `slee_acs` plug-in implements the chassis actions which are used by the VPN macro nodes when they need to interact with entities outside `slee_acs`.

Location

This library is located on SLCs.

Startup

If `libvpnChassisActions` is configured in **acs.conf**, it is made available to `slee_acs` when `slee_acs` is initialized. It is included in the `acsChassis` section of **acs.conf** in a `ChassisPlugin` entry.

```
acsChassis
ChassisPlugin libvpnChassisActions.so
```

For more information about this entry in **acs.conf**, see *acsChassis plugins* (on page 13).

Configuration

`libvpnChassisActions` are configured in `acs.conf`. For more information about configuring the VPN chassis actions, see *VPN section* (on page 13).

Tools and Utilities

Overview

Introduction

This chapter explains the procedures for maintaining the VPN application.

In this chapter

This chapter contains the following topics.

Examining Network and Station Profiles 39

Examining Network and Station Profiles

Introduction

You can view network and station profile details using the vpnProfile utility.

vpnProfile

vpnProfile is shipped with both the vpnSms and vpnScp packages. It is located in `/IN/service_packages/ACS/bin`.

Refer to Technical Support for more information about using this tool, and for details on the format of VPN profiles for networks and stations, and how to interpret them.

Station profile

Follow these steps to determine the contents of a station profile.

Step	Action
1	Log on as acs_oper.
2	Enter the following command: <code>./vpnProfile -S N</code> where <i>N</i> is the station ID. This will be a numeric value, for example, 1. Note: You may need to set your path to the script location. Result: The profile for the requested station appears.

Example station profile

Here is an example of the output from vpnProfile.

```
$ ./vpnProfile -S 2
Connecting as smf/smf
Station '2000' on Network 'network2'
Profile is (1216 bytes) and (21 tags).
```

Tag	Desc	Len	HexData...
0x000009	Out BA Typ	1	02
0x00000a	Out BA Lst	12	00 00 00 01 00 00 00 31 00 00 00 01
0x00000b	Inc BA Typ	1	02
0x00000c	Inc BA Lis	12	00 00 00 01 00 00 00 31 00 00 00 01
0x000010	Divert RSF	6	02 36 37 38 39 30
0x000013	Divt FlwMe	6	02 31 32 33 34 35
0x000015	PIN Digits	4	31 32 33 34
0x000017	OffNet Bar	1	00
0x000019	Stn is Mgr	1	01
0x00001a	Speed List	24	
SubTags (1):			
0x000000	5	01 31 30 30 30	
0x00001b	Dvt BA Typ	1	01
0x00001c	Dvt BA Lst	12	00 00 00 01 00 00 00 33 00 00 00 01
0x000025	Language	4	00 00 00 01
0x000034	HuntOnBusy	1	00
0x000035	HuntOnNoAn	1	00
0x000036	HuntAlways	1	00
0x030003	V.BarAllIn	4	00 00 00 01
0x030004	V.BarOffIn	4	00 00 00 01
0x030020	V.HuntConf	824	
SubTags (2):			
0x000001	V.HntLtTg	544	
SubTags (2):			
0x000001	V.ListNo1	228	
SubTags (4):			
0x000001	V.TermNo1	56	
SubTags (3):			
0x000001	V.TermNum	5	31 30 30 30 31
0x000002	V.OnNetFg	4	00 00 00 00
0x000003	V.TimeOut	4	00 00 00 01
0x000002	V.TermNo2	52	
SubTags (3):			
0x000001	V.TermNum	4	31 30 30 32
0x000002	V.OnNetFg	4	00 00 00 00
0x000003	V.TimeOut	4	00 00 00 02
0x000003	V.TermNo3	52	
SubTags (3):			
0x000001	V.TermNum	4	31 30 30 33
0x000002	V.OnNetFg	4	00 00 00 00
0x000003	V.TimeOut	4	00 00 00 03
0x00ffff	V.Default	14	68 75 6e 74 69 6e 67 20 6c 69 73 74 20 31
0x000002	V.ListNo2	288	
SubTags (5):			
0x000001	V.TermNo1	52	
SubTags (3):			
0x000001	V.TermNum	4	32 30 30 31
0x000002	V.OnNetFg	4	00 00 00 00
0x000003	V.TimeOut	4	00 00 00 01
0x000002	V.TermNo2	52	
SubTags (3):			
0x000001	V.TermNum	4	32 30 30 32
0x000002	V.OnNetFg	4	00 00 00 00
0x000003	V.TimeOut	4	00 00 00 02
0x000003	V.TermNo3	52	
SubTags (3):			
0x000001	V.TermNum	4	32 30 30 33
0x000002	V.OnNetFg	4	00 00 00 00
0x000003	V.TimeOut	4	00 00 00 03
0x000004	V.TermNo4	52	
SubTags (3):			

```

0x000001 V.TermNum 4      32 30 30 30
0x000002 V.OnNetFg 4      00 00 00 02
0x000003 V.TimeOut 4      00 00 00 00
0x00ffff V.Default 14    68 75 6e 74 69 6e 67 20 6c 69 73 74 20 32
0x000002 V.HntPlan 252
SubTags (3):
0x000000 V.PlanNo0 108
SubTags (6):
0x000001 V.PlLocT 5      5e 32 30 30 30
0x000002 V.PlCliT 5      5e 32 30 30 30
0x000003 V.TimType 4      00 00 00 01
0x000004 V.StartTm 4      00 00 00 64
0x000005 V.EndTime 4      00 00 00 c8
0x000006 V.ListIdT 4      00 00 00 02
0x000001 V.PlanNo1 100
SubTags (6):
0x000001 V.PlLocT 1      5e
0x000002 V.PlCliT 1      5e
0x000003 V.TimType 4      00 00 00 01
0x000004 V.StartTm 4      00 00 00 c8
0x000005 V.EndTime 4      00 00 04 b0
0x000006 V.ListIdT 4      00 00 00 01
0x00ffff V.Default 4      00 00 00 01
0x030100 V.Unknown! 4      00 00 00 01
0x030101 V.Unknown! 7      28 53 43 49 29 31 31
Total Size: 1216

```

Network profile

Follow these steps to view the contents of a network profile.

Step	Action
1	Log on as acs_oper.
2	Enter the following command: <pre>./vpnProfile -N N</pre> where <i>N</i> is the network ID. This will be a numeric value, for example, 1. Note: You may need to set your path to the script location. Result: The requested network profile is displayed.

Example network profile

Here is an example of the output for a VPN network profile:

```

$ ./vpnProfile -N 2
Connecting as smf/smf
Network 'networ2'
Profile is (536 bytes) and (27 tags).

Tag      Desc      Len  HexData...
-----
0x000004 A/C Max Le 1      04
0x000009 Out BA Typ 1      01
0x00000a Out BA Lst 12     00 00 00 01 00 00 00 31 00 00 00 01
0x00000b Inc BA Typ 1      01
0x00000c Inc BA Lis 12     00 00 00 01 00 00 00 31 00 00 00 01
0x00000d A/C Values 52     00 00 00 01 00 00 32 32 ff ff ff ff 00 00 60 02 ff ff ff ff
30 30 00 00 00 00 00 03 00 00 00 30 00 00 00 01 00 00 00 31 00 00 00 01 00 00 00 32
00 00 00 01

```

Chapter 4

```
0x00000e A/C Policy 1      01
0x000016 PIN Rights 4      00 00 03 f7
0x00001a Speed List 24
  SubTags (1):
    0x000000 5      02 32 30 30 30
0x00001e Break Limt 1      04
0x00001f LCR OldNat 4      31 31 31 31
0x000020 LCR NewNat 5      31 31 31 31 31
0x000021 LCR OldInt 5      32 32 32 32 32
0x000022 LCR NewInt 6      32 32 32 32 32 32
0x000032 SilentDisc 1      00
0x000033 HlpLinAddr 5      02 31 31 31 31
0x030001 V.SdNoChck 4      00 00 00 00
0x030002 V.PrsntPrv 4      00 00 00 01
0x030005 V.PinPrefix 2      39 38
0x030006 V.AcctPrfx 2      39 37
0x030007 V.AltStPrf 2      39 39
0x030008 V.OfNetPrf 2      39 35
0x030009 V.SpdDlPrf 2      39 36
0x03000a V.PinLngth 4      00 00 00 04
0x03000c V.StnLngth 4      00 00 00 04
0x03000f V.RstrAddr 4      00 00 00 01
0x030010 V.AllSrtXt 4      00 00 00 01
Total Size: 536
$
```


About Installation and Removal

Overview

Introduction

This chapter provides information about the installed components for the Oracle Communications Network Charging and Control (NCC) application described in this guide. It also lists the files installed by the application that you can check for, to ensure that the application installed successfully.

In this Chapter

This chapter contains the following topics.

<i>Installation and Removal Overview</i>	43
<i>Restarting the Stats Daemon</i>	44

Installation and Removal Overview

Introduction

For information about the following requirements and tasks, see *Installation Guide*:

- *NCC system requirements*
- *Pre-installation tasks*
- *Installing and removing NCC packages*

VPN packages

An installation of VPN includes the following packages, on the:

- *SMS*:
 - vpnSms
- *SLC*:
 - vpnScp

Environment variables

During the installation of the acsSms package, you are prompted for cluster related information. This information determines which environment variables are written to the **variables_sms.sh** file. The environment variables are used during the installation of vpnSms.

The table below shows which environment variables are created, and when they are created.

Install acsSMS prompt	Result	Environment Variables Created	Value
1. Install ACS SMF schema on database SMF (on this node) ?	If you reply 'Y', then prompt 2, below, appears.	INSTALL_DATABASE	Y

Install acsSMS prompt	Result	Environment Variables Created	Value
	If you reply 'N', the database is not installed on this node, for example, because it is the secondary node in the cluster.	INSTALL_DATABASE	N
2. Install ACS SMF schema on database SMF in clustered mode ?	If you reply 'Y', you are prompted to populate the <code>acs_devices.sh</code> file.	CLUSTER_MODE	Y
	If you reply 'N', you are prompted to enter the OSF mount points. You must enter the full path to the data files.	CLUSTER_MODE DATAFILEPATH1 DATAFILEPATH2 DATAFILEPATH3	N

Environment variables for vpnScp

The following environment variables are created by the acsScp package installation process. They are defined in the `variables_scp.sh` file:

- `DATAFILEPATH1`
- `DATAFILEPATH2`
- `DATAFILEPATH3`

These environment variables are required for the installation of the vpnScp package.

Note: The variables must contain the full data file path.

Restarting the Stats Daemon

Introduction

VPN defines a number of statistics installed on the SMS package.

Enabling the statistics

To enable the new statistic definitions, you must restart the `smsStatsDaemon` on the SLC.

Procedure

Follow these steps to restart the `smsStatsDaemon`.

Step	Action
1	Log into the SCP as <code>smf_oper</code> .
2	Enter the following command: <code>ps -ef grep smsStatsDaemon</code> Result: The PID of the <code>smsStatsDaemon</code> appears.
3	Enter a command in the following format: <code>Kill -9PID_of_the_smsStatsDaemon</code> Result: The <code>smsStatsDaemon</code> is restarted by the <code>init</code> tab.