Copyright

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

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, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 Xeon 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, Opteron, the AMD logo, and the AMD Opteron 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 on 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. 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.
## Contents

About This Document ........................................................................................................v
Document Conventions ...................................................................................................vi

### Chapter 1

**System Overview** ........................................................................................................1

- Overview ....................................................................................................................1
- SMS Interface System Environment .......................................................................1
- smsInterface Connection Management ...............................................................3
- Notification Interface .............................................................................................4

### Chapter 2

**Configuration** ...........................................................................................................7

- Overview ....................................................................................................................7
- Configuring the Environment ................................................................................7
- eserv.config Configuration .....................................................................................8
- Configuring the smscIF.cfg ...................................................................................10

### Chapter 3

**Background Processes** ..........................................................................................17

- Overview ....................................................................................................................17
- notificationIF ............................................................................................................17
- smsInterface ..........................................................................................................22

### Chapter 4

**About Installation and Removal** ...........................................................................23

- Overview ....................................................................................................................23
- Installation and Removal Overview ......................................................................23
- Installation Prerequisites ........................................................................................23
- Post-installation Configuration ...............................................................................24

**NCC Glossary of Terms** .........................................................................................25

**Index** .......................................................................................................................31
About This Document

Scope

The scope of this document includes all functionality a user must know in order to effectively install and configure the SMS Interface.

The purpose of this document is to describe how to use the SMS Interface on an intelligent Network platform.

The document is not intended to detail the technical design of the SMS Interface or to advise on the network implications of operating the SMS Interface.

Audience

This guide was written primarily for SMS Interface installers and System Administrators. 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 Intelligent Network architectures would be an advantage.

As well as an understanding of the Short Message Service center EMI-UCP Interface specification Version 3.5.

Related documents

The following documents are related to this document:

• [1] EMI – UCP Interface Specification 3.5
Document Conventions

Typographical Conventions

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

<table>
<thead>
<tr>
<th>Formatting convention</th>
<th>Type of information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Special Bold</strong></td>
<td>Items you must select, such as names of tabs. Names of database tables and fields.</td>
</tr>
<tr>
<td><strong>Italics</strong></td>
<td>Name of a document, chapter, topic or other publication. Emphasis within text.</td>
</tr>
<tr>
<td><strong>Button</strong></td>
<td>The name of a button to click or a key to press. <strong>Example:</strong> To close the window, either click <strong>Close</strong>, or press <strong>Esc</strong>.</td>
</tr>
<tr>
<td><strong>Key+Key</strong></td>
<td>Key combinations for which the user must press and hold down one key and then press another. <strong>Example:</strong> <strong>Ctrl+P</strong>, or <strong>Alt+F4</strong>.</td>
</tr>
<tr>
<td><strong>Monospace</strong></td>
<td>Examples of code or standard output.</td>
</tr>
<tr>
<td><strong>Monospace Bold</strong></td>
<td>Text that you must enter.</td>
</tr>
<tr>
<td><strong>variable</strong></td>
<td>Used to indicate variables or text that should be replaced.</td>
</tr>
<tr>
<td><strong>menu option &gt; menu option &gt;</strong></td>
<td>Used to indicate the cascading menu option to be selected, or the location path of a file. <strong>Example:</strong> <strong>Operator Functions &gt; Report Functions</strong> <strong>Example:</strong> <strong>/IN/html/SMS/Helptext/</strong></td>
</tr>
<tr>
<td><strong>hypertext link</strong></td>
<td>Used to indicate a hypertext link on an HTML page.</td>
</tr>
</tbody>
</table>

Specialized terms and acronyms are defined in the **Glossary** at the end of this guide.
Chapter 1

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.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS Interface System Environment</td>
<td>1</td>
</tr>
<tr>
<td>smsInterface Connection Management</td>
<td>3</td>
</tr>
<tr>
<td>Notification Interface</td>
<td>4</td>
</tr>
</tbody>
</table>

SMS Interface System Environment

What is the SMS interface?

SLEE applications can use the smsInterface to send messages to an SMS center. Messages use the UCP protocol defined in EMI-UCP Interface Specification.
Architectural overview

The following diagram shows the smsInterface and sub-system components that surround it.

Sub-system components

This table provides notes about components surrounding the smsInterface.

<table>
<thead>
<tr>
<th>Component</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLEE</td>
<td>The service logic execution environment. The SLEE:</td>
</tr>
<tr>
<td></td>
<td>- Starts and stops the smsInterface</td>
</tr>
<tr>
<td></td>
<td>- Directs communication to the smsInterface from other NCC applications.</td>
</tr>
<tr>
<td></td>
<td>The SLEE must be installed before the smsInterface.</td>
</tr>
<tr>
<td>smsclF.cfg file</td>
<td>A configuration file, smsclF.cfg.example, is supplied in the form of an example. You can find it at /IN/service_packages/SLEE/etc</td>
</tr>
<tr>
<td></td>
<td>If you use this file you must rename it to:</td>
</tr>
<tr>
<td></td>
<td>smsclF.cfg</td>
</tr>
<tr>
<td>Log file</td>
<td>The log file, smsclF.log, is created at run time. You can find it at /IN/service_packages/SLEE/tmp</td>
</tr>
<tr>
<td>Trace log</td>
<td>The trace log file is created at run time. You can find it either at /IN/service_packages/UCP/tmp or, if the UCP directory does not exist, at /tmp</td>
</tr>
<tr>
<td>Prefix file</td>
<td>The prefix file, smsclPrefix.cfg, is created as part of the user’s configuration process. You can find it at /IN/service_packages/SLEE/etc</td>
</tr>
<tr>
<td>SMS message structures</td>
<td>SMS message structures supported are:</td>
</tr>
<tr>
<td></td>
<td>- Submit Short Message Operation(51)</td>
</tr>
<tr>
<td></td>
<td>- Session Management Operation(60)</td>
</tr>
<tr>
<td></td>
<td>See EMI-UCP Interface Specification.</td>
</tr>
</tbody>
</table>
**smsInterface Connection Management**

**Purpose**

The smsInterface can queue and manage the sending of SMS messages to one or more SMSCs.

**Message queuing**

The message events are held in one of two queues.

<table>
<thead>
<tr>
<th>Queue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Live</td>
<td>This queue contains those messages currently awaiting acknowledgment, up to a maximum of 100 messages. Messages are only placed on this queue if there is a live connection to an SMSC and if there is room for more messages. Messages are removed from the live queue when they have been acknowledged or if they time out.</td>
</tr>
<tr>
<td>Holding</td>
<td>If all 100 'slots' in the live queue are occupied by messages that have not completed or if there is no live connection the messages are placed in this queue. This queue has a configurable (queueMaximum (on page 13)) size. A message may be removed from the holding queue if the message is older than msgLife (on page 13). Any remaining messages will be re-sent upon successful re-connection to a SMSC. This occurs as part of the close connection process called whenever messages are being transmitted to the SMSC and a failure is detected.</td>
</tr>
</tbody>
</table>

**Connection process**

The IP addresses of two SMSCs can be configured.

The IP addresses are to be read from a configuration file.

During the connection process an attempt is first made to connect to the current (or initial) SMSC. Upon program startup the Primary machine will be the first one that a connection attempt is made to.

Should this fail then a count of sequential failures is made. If this number is greater than the configuration file parameter (retry number) then a connection is attempted to the other SMSC (there is a timed delay set by the configuration file parameter (retry interval) between each new connection attempt).

Should the attempt to this second SMSC also fail (retry number) times a configured delay of (fail over interval) is made before a re-connect attempt is made to the OTHER SMSC.

A user signal (USR2) can be sent that will trigger a change back to the primary IP. This will call the close connection process resulting in all those messages that are in the 100 element live queue being resent when a new connection is achieved.

**Connection failure**

If a connection fails, remaining messages are moved from the live queue to the holding queue and the connection is terminated as part of the close connection / connect process.

The connection will be considered failed if any of the following conditions are met:

- Acknowledgment timeout (acknowledge timeout) - No response from the SMSC.
- Idle timeout - No communication for period of time defined by idleTimeout (on page 11).
- SMS negative acknowledgment error - Failure message is received from the SMSC.
- Read / Write failure of connection.
When messages are placed in the live queue they are removed from the holding queue.

**Login event**

A close connection / re-connect may also arise from a failure of the login message.

The login message is sent after a successful connection but before any attempt to send SMS messages is made. The login message expects a result back from the SMSC. If it does not get one within the configuration file parameter (login response time) or an error is returned then a close connection process is initiated followed by the reconnection process.

**Notification Interface**

**Architectural overview**

The following diagram shows the notificationIF and sub-system components that surround it. It also identifies notificationIF message flows.

![Diagram showing notificationIF and sub-system components](image)

**Message flows**

The Notification interface uses the message flows identified in the diagram as FL1 through FL 8.

<table>
<thead>
<tr>
<th>Flow Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FL1</td>
<td>Java management screens access the SMF database to edit the list of SMS templates.</td>
</tr>
<tr>
<td>FL2</td>
<td>A client application sends a request, via a FIFO, to the ccsSSMDispatcher running on the SMS. An example of a client application would be a smsTrigDaemon.</td>
</tr>
<tr>
<td>FL3</td>
<td>The ccsSSMDispatcher forwards requests to the ccsSSMMaster running on the SLC platform. Requests are forwarded through a TCP/IP socket.</td>
</tr>
</tbody>
</table>
| FL4 | ccsSSMMaster constructs a SLEE notification event with:  
  - Application name  
  - Notification type  
  - Language  
  - Destination MSISDN  
  - Optional variable parts  
  SLEE notification events are sent to the Notification interface. |
| FL5 | notificationIF is able to forward requests to smsInterface. |
| FL6 | The smsInterface sends requests it receives from the notificationIF to the SMSC. Requests are forwarded through a TCP/IP network. |
| FL7 | notificationIF is also able to forward requests to Messaging Manager’s xmsTrigger process. |
| FL8 | xmsTrigger sends requests it receives from the notificationIF directly to the destination. It uses the TCAP FDA interface. |
Chapter 2
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.

- Configuring the Environment .................................................. 7
- eserv.config Configuration .................................................... 8
- Configuring the smscIF.cfg ..................................................... 10

Configuring the Environment

Startup configuration

notificationIF and smsInterface are started by shell scripts. They can be used to set environmental parameters. For more information about smscIF.sh, see SLEE.cfg (on page 8).

SMSC supports the following environmental parameters:

LOG_FILE
Syntax: LOG_FILE=PathFile
Description: The name of the file to which alarms raised by the smsInterface will be logged.
Type: String
Optionality: Optional (default used if not set).
Allowed: None
Default: /IN/service_packages/SLEE/tmp/notificationIF.log
Notes: For more information about SMSC alarms, see System Alarms.
Example: LOG_FILE=/IN/service_packages/SLEE/tmp/smsscIF.log
LOG_FILE=/IN/service_packages/SLEE/tmp/notificationIF.log

CONF_FILE
Syntax: CONF_FILE=PathFile
Description: The directory smsInterface will look for smscIF.sh configuration file in.
Type: String
Optionality: Optional (default used if not set).
Allowed: None
Default: /IN/service_packages/SLEE/etc/smsscIF.sh
Notes: None
Location of eserv.config

By default, notificationIF will read its configuration from the notificationIF section of /IN/service_packages/eserv.config.

To override the default location, use the ESERV_CONFIG_FILE environmental variable.

**ESERV_CONFIG_FILE**

**Syntax:**

```
ESERV_CONFIG_FILE = "path/file"
```

**Description:**

The directory eserv.config configuration file will be read from.

**Type:** String

**Optionality:** Optional (default used if not set).

**Allowed:**

Default: /IN/service_packages/eserv.config

**Notes:**

Example:

```
CONF_FILE=/IN/service_packages/SLEE/etc/smscIF.cfg
```

SLEE.cfg

The SLEE is responsible for starting and stopping the smsInterface and notificationIF.

The SLEE also directs all communication to the smsInterface from other Oracle components.

A row in the SLEE.cfg file tells the SLEE which executable to run and gives it an identifier for other applications. For more information about this line, see Startup (on page 22).

Multiple instance configuration

Multiple occurrences of the smsInterface (on page 22) may be configured within the SLEE.cfg file.

This can allow SMS messages to be sent to more than one SMSC by creating multiple smscIF.cfg files (one for each unique instance of the smsInterface) and within these configuration files, defining different smsInterface IPs.

Each instance defined within the SLEE.cfg file must have a unique handle. See Single Instance Configuration above.

Thus two entries within the SLEE.cfg file might be of the form:

```
INTERFACE=smscIF smscIF.sh /IN/service_packages/SLEE/bin UDG
INTERFACE=smscIF2 smscIF2.sh /IN/service_packages/SLEE/bin UDG
```

**Note:** There must always be one instance with the handle smscIF.

For more information about SLEE.cfg and INTERFACE entries, see SLEE Technical Guide.

eserv.config Configuration

**Introduction**

The eserv.config file is a shared configuration file, from which many 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 up 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 may be used, as in this example:

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

or

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

or

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

**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 will cause 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 will ensure you have a working copy to which you can return.

**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.

**Warning:** This file is not intended to be changed by the user. Please contact Oracle support with your queries.
Chapter 2

Example eserv.config file

To see an example of a configured eserv.config file, see Example notificationIF section (on page 21).

Configuring the smsclF.cfg

Overview

The smsInterface (on page 22) needs to be set up to connect to the required SMSC. You do this by setting the values of parameters in the smsclF.cfg configuration file.

Configuration file structure

The structure of smsclF.cfg is summarized below.

- The configuration file consists of a number of lines, each of which defines a parameter name and associated value or values.
- Each line can contain characters totaling no more than 1024 bytes.
- In any line, the parameter name and value or values are separated from each other by a single space. This means that parameter names and values must not contain spaces.
- The parameter name is the first group of characters in the line.
- All parameter lines are case-sensitive.
- Free-form comments are allowed. Lines containing free-form comments are preceded by the ASCII 35 (#) symbol.

Configuration file location

smsclF.cfg is located in /IN/service_packages/SLEE/etc.

Parameters

smsclF.cfg contains the following parameters.

cdrInterfaceName

Syntax: cdrInterface name
Description: The handle of the EDR Interface process to use to write EDRs.
Type: String
Optionality: Optional (not used if not set).
Allowed: Must match the handle for the EDR Interface set in SLEE.cfg.
Default:
Notes:
Example: cdrInterface cdrIF

cdrLibrary

Syntax: cdrLibrary path bin
Description: The UPC library to use to generate EDRs based on SMS messages sent from smsInterface.
Type: String
Optionality: Optional (not used if not set).
Allowed:
Default:
Notes: For more information about libsmsCdr, see *USSD GW Technical Guide*.
Example: cdrLibrary /IN/service_packages/UPC/lib/libsmsCdr.so

classDefault
Syntax: classDefault 0|1|2|3
Description: When smsInterface receives an SMS message over the SLEE and the message doesn't have a MessageClass set, set the class parameter to this value in the message to the SMSC.
Type: Integer
Optionality: Optional (default used if not set).
Allowed: -1 Do not set a message class (leave blank).
       0
       1
       2
       3
Default: -1
Notes: Example: classDefault 0

connectTimeout
Syntax: connectTimeout seconds
Description: After a connection attempt, the maximum number of seconds that smsInterface (on page 22) will wait for a response from the SMSC before deciding that the attempt has failed.
Type: Integer
Optionality: Mandatory
Allowed: Default: 60
Notes: Example: connectTimeout 60

failoverInterval
Syntax: failoverInterval seconds
Description: After a failed connection to the SMSC's primary IP address (ip1 on page 12), the maximum number of seconds that smsInterface (on page 22) will wait before attempting to reconnect.
Type: Integer
Optionality: Mandatory
Allowed: Default: 60
Notes: Example: failoverInterval 60

idleTimeout
Syntax: idleTimeout seconds
Description: The maximum number seconds, that smsInterface (on page 22) will remain...
inactive.
Type: Integer
Optionality: Mandatory
Allowed: 
Default: 86400
Notes: 86400 seconds is one day.
Example: idleTimeout 60

ip1
Syntax: ip1 ip port
Description: The primary address of the SMSC.
The port number associated with the primary address of the SMSC.
Type: ip is an Internet protocol number in dotted-decimal notation. For example, 125.3.57.155.
port is an integer.
Optionality: Mandatory
Allowed: 
Default: None
Notes: 
Example: ip1 127.0.0.1 6666

ip2
Syntax: ip2 ip port
Description: The secondary address of the SMSC.
The port number associated with the secondary address of the SMSC.
Type: ip is an Internet protocol number in dotted-decimal notation. For example, 125.3.57.155.
port is an integer.
Optionality: Optional
Allowed: 
Default: None
Notes: 
Example: ip2 127.0.0.15 6666

loginTimeout
Syntax: loginTimeout seconds
Description: The maximum number of seconds, that smsInterface (on page 22) will wait for a logon message from the SMSC.
Type: Integer
Optionality: Mandatory
Allowed: 
Default: 60
Notes: 
Example: loginTimeout 60
maxSmsPerSecond

Syntax: maxSmsPerSecond max

Description: The maximum number of SMS sent to the SMS Center in one second.

Type: Integer

Optionality: Optional (default used if not set).

Default: 2147483647 (unlimited)

Notes: The throttling limit is calculated by measuring the number of messages sent in each 10th of a second. If the number of messages sent in the last 10 deciseconds adds up to more than maxSmsPerSecond, we queue the message instead of sending it.

Example: maxSmsPerSecond 2000

msgLife

Syntax: msgLife seconds

Description: The maximum number of seconds that smsInterface (on page 22) will keep messages in the queue.

Type: Integer

Optionality: Mandatory

Default: 3600

Notes: 3600 seconds is one hour.

Example: msgLife 3600

password

Syntax: password passwd

Description: The password sent from smsInterface (on page 22) to the SMSC.

Type: String

Optionality: Mandatory

Default: None

Notes: Example: password ALPHA@NUM

queueMaximum

Syntax: queueMaximum max

Description: The maximum number of elements that smsInterface (on page 22) will queue in the holding queue.

Type: Integer

Optionality: Mandatory

Default: 5000

Notes: For more information about message queuing, see Message queuing (on page 3).

Example: queueMaximum 5000
retryInterval
Syntax: retryInterval seconds
Description: The maximum number of seconds after a failed connection attempt that smsInterface (on page 22) will wait before attempting another connection.
Type: Integer
Optionality: Mandatory
Allowed: 
Default: 1
Notes: 
Example: retryInterval 60

retryNumber
Syntax: retryNumber int
Description: After a failed attempt to connect to connect to an IP address, the maximum number of times that smsInterface (on page 22) will attempt to connect to the same address again.
Type: Integer
Optionality: Mandatory
Allowed: 
Default: 1
Notes: 
Example: retryNumber 1

tracingDirectory
Syntax: tracingDirectory dir
Description: The directory to write the tracing log to.
Type: String
Optionality: Optional (default used if not set).
Allowed: 
Default: /IN/service_packages/UCP/tmp
If that directory is not available, defaults to /tmp.
Notes: New logging is appended to the end of any previous file.
File is named in the following format:
ucpTrace%04u%02u%02u.log
Example: tracingDirectory /IN/service_packages/SMSC/tmp

tracingEnabled
Syntax: tracingEnabled true|false
Description: Whether or not to enable tracing.
Type: Boolean
Optionality: 
Allowed: true, false
Default: 
Notes: 
Example: tracingEnabled false
**userID**

**Syntax:** userID name

**Description:** The logon name sent from smsInterface (on page 22) to the SMSC.

**Type:** String

**Optionality:** Mandatory

**Allowed:**

**Default:** None

**Notes:**

**Example:** userID ALPHA@NUM

---

**Configuration file example**

Listed below are lines of a typical smscIF.config file. Comment lines have been removed.

```
loginTimeout 60
idleTimeout 60
retryInterval 60
failoverInterval 60
connectTimeout 60
ip1 127.0.0.1 6666
ip2 127.0.0.15 6666
retryNumber 1
queueMaximum 5000
msgLife 3600
userID ALPHA@NUM
password ALPHA@NUM
maxSmsPerSecond 2000
```

**Note:** An example smscIF.cfg file called smscIF.cfg.example is provided when SMSC is installed.
Overview

Introduction

This chapter explains the processes which run automatically as part of the application. These processes are started automatically by one of the following:

- inittab
- crontab
- 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.

<table>
<thead>
<tr>
<th>notificationIF</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>smsInterface</td>
<td>22</td>
</tr>
</tbody>
</table>

notificationIF

Purpose

notificationIF is a SLEE application that sends preformed text messages from NCC applications to customer's mobile telephones using smsInterface.

notificationIF receives commands in the form of SLEE events. These events determine:

- MSISDN
- Application
- Type
- Language
- A list of substitution parameters

The application, type and language are used to source the message.

The MSISDN routes the message.

The substitution parameters customize the message.

Location

This binary is located on SLCs.

Startup

This task is started by the SLEE, by the following lines in SLEE.cfg:
Chapter 3

INTERFACE=Notification notificationIF.sh /IN/service_packages/SLEE/bin

UDG

Notes:
- notificationIF.sh is a shell script which starts the notificationIF process.
- The above are defaults and may vary.

Configuration

notificationIF reads the notificationIF section of the eserv.config file. For more general information about this file, see eserv.config Configuration (on page 8).

Parameters

Parameters of the notificationIF section of the eserv.config file are defined below.

fromAddress

Syntax: fromAddress = "OriginatingAddress"
Description: The originating address in outbound SMSs.
Type: String
Optionality: Optional
Allowed: None
Default: None
Notes:
Example: fromAddress = "441473289900"

oracleLogin

Syntax: oracleLogin = "UserName/Password"
Description: The user name and password that the Notification interface must use when it connects to the Oracle database.
Type: String
Optionality: Mandatory
Allowed: /
Default: /
Notes:
Example: oracleLogin = "/"

smsIF

Syntax: smsIF = "handle"
Description: The SLEE interface handle of the SMS interface.
Type: String
Optionality: Mandatory
Allowed: smscIF
Default: smscIF
Notes:
Example: smsIF = "smsIF"
xmsDestNPI
Syntax: xmsDestNPI = indicator
Description: The XMS destination numbering plan indicator.
Type: Integer
Optionality: Optional
Allowed: Refer to the xmsOrigNPI parameter for permitted values for this parameter.
Default: 0
Notes:
Example: xmsDestNPI = 0

xmsDestTON
Syntax: xmsDestTON = type
Description: The XMS destination number type
Type: Integer
Optionality: Optional
Allowed: Refer to the xmsOrigTON parameter for permitted values for this parameter.
Default: 0
Notes:
Example: xmsDestTON = 0

xmsDirectFromPrefix
Syntax: xmsDirectFromPrefix = "prefix"
Description: The prefix that precedes the originating address when using MM Direct.
Type: String
Optionality: Optional
Allowed:
Default: None
Notes:
Example: xmsDirectFromPrefix = "2"

xmsFDAFromPrefix
Syntax: xmsFDAFromPrefix = "prefix"
Description: The prefix that precedes the originating address when using MM FDA.
Type: String
Optionality: Optional
Allowed:
Default: None
Notes:
Example: xmsFDAFromPrefix = "1"

xmsIWrapperIfName
Syntax: xmsIWrapperIfName = "handle"
Description: The SLEE interface handle of the XMSI wrapper interface.
Type: String
Optionality: Optional
Allowed: None
Notes: Messaging Manager cannot be used if this parameter is missing or left blank.
Example: xmsiWrapperIfName = "xmsIP"

**xmsOrigNPI**

**Syntax:**

```
xmsOrigNPI = indicator
```

**Description:** The XMS originating numbering plan indicator.

**Type:** Integer

**Optionality:** Optional

**Allowed:** Numbering plan bits allocated to the Global Title Indicator 0011.

<table>
<thead>
<tr>
<th>Bits</th>
<th>type</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
<td>Unknown</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
<td>ISDN and telephony numbering plan.</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
<td>Generic numbering plan.</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
<td>Data numbering plan.</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
<td>Telex numbering plan.</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
<td>Maritime mobile numbering plan.</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
<td>Land mobile numbering plan.</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
<td>ISDN and mobile numbering plan.</td>
</tr>
<tr>
<td>1110</td>
<td>14</td>
<td>Private network or network-specific numbering plan.</td>
</tr>
</tbody>
</table>

**Default:** 0

**Notes:** For information about Global Title Indicator 0011, refer to ITU-T Q.713, *Specifications of Signalling System No. 7 — Signalling Connection Control part (SCCP)*.

**Example:** `xmsOrigNPI = 0`

**xmsOrigTON**

**Syntax:**

```
xmsOrigTON = type
```

**Description:** The XMS originating number type.

**Type:** Integer

**Optionality:** Optional

**Allowed:** Encoding scheme bits allocated to Global Title Indicator 0011.

<table>
<thead>
<tr>
<th>Bits</th>
<th>type</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
<td>Unknown</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
<td>BCD, odd number of digits.</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
<td>BCD, even number of digits.</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
<td>National specific.</td>
</tr>
</tbody>
</table>

**Default:** 0

**Notes:** For information about Global Title Indicator 0011, refer to ITU-T Q.713,
Specifications of Signalling System No. 7 — Signalling Connection Control part (SCCP).

Example: 
\[ \text{xmsOrigTON} = 0 \]

**xmsPC**

**Syntax:** 
\[ \text{xmsPC} = \text{destination} \]

**Description:** The XMS destination PC.

**Type:** Integer

**Optionality:** Optional

**Allowed:**

**Default:** 1

**Notes:**

Example: 
\[ \text{xmsPC} = 1 \]

**xmsSSN**

**Syntax:** 
\[ \text{xmsSSN} = \text{destination} \]

**Description:** The XMS destination SSN.

**Type:** Integer

**Optionality:** Optional

**Allowed:**

**Default:** 1

**Notes:**

Example: 
\[ \text{xmsSSN} = 1 \]

### Example notificationIF section

An example of the notificationIF section of the eserv.config file is listed below. Comment lines have been removed.

```
notificationIF = {
    oracleLogin = "/",
    smsIF = "smscIF",
    xmsIWrapperIfName = "xmsIIf",
    fromAddress = "441473289900",
    xmsFDAFromPrefix = "1",
    xmsDirectFromPrefix = "2",
    xmsPC = 1
    xmsSSN = 1
    xmsOrigTON = 0
    xmsOrigNPI = 0
    xmsDestTON = 0
    xmsDestNPI = 0
}
```

### Output

notificationIF logs errors to the file specified by LOG_FILE (on page 7).
smsInterface

Purpose

smsInterface provides an interface to SMS Centers to send SMS messages. It is one of two main processes in the SMSC component.

Location

This binary is located on SLCs.

Startup

This task is started by the SLEE, by the following lines in SLEE.cfg:

```
INTERFACE=smscIF smscIF.sh /IN/service_packages/SLEE/bin UDG
```

Notes:

- smscIF.sh is a shell script which starts the smsInterface process.
- The above are defaults and may vary.
- In a standard configuration, only a single instance of the smsInterface will be started. Multiple instances of the smsInterface process can be run. For more information about multi-interface configuration, see Multiple instance configuration (on page 8).

Configuration

smsInterface is configured by the smscIF.cfg file. For more information about this configuration, see Configuring the smscIF.cfg (on page 10).

Output

smsInterface logs errors to the file specified by LOG_FILE (on page 7).
Overview

Introduction

This chapter provides details of the installation and removal process for the application.

In this chapter

This chapter contains the following topics.

Installation and Removal Overview 23
Installation Prerequisites 23
Post-installation Configuration 24

Installation and Removal Overview

Introduction

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

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

SMS Interface component package

An installation of the SMS Interface component package includes the following, on the SLC:

- SMSC

Installation Prerequisites

Set SMSC database privilege

Before the installation of the SMSC a privilege in the Oracle DB must exist or, if it does not exist, must be granted.

Follow these steps to set the SMSC database privilege.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Find out which OS user(s) run the SLEE instance(s) that run the SMSC.</td>
</tr>
</tbody>
</table>
| 2    | Grant execution privileges to the corresponding Oracle user(s): Enter on the command line:  
  `su - oracle -c "ORACLE_SID=SCP sqlplus '/ as sysdba'"` |
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 3    | In the sqlplus session, enter on the command line: grant execute on sys.dbms_alert to ops$ccs_oper;  
**Note:** ccs_oper is the OS user. |
| 4    | Repeat steps 2 and 3 for each OS user. |
| 5    | Restart the relevant SLEE instance. |
| 6    | Repeat steps 2 through 5 for each SLEE instance. |

**Post-installation Configuration**

**Configuring the installation**

When running the installation, the configuration is done automatically. However, the configuration file for the smsInterface **smsIF.cfg** is not put in place from the package installation but is instead left with a suffix **.example**

This will not be picked up by the smsInterface. The customer must edit this file to configure the operation for the smsInterface to their requirements. When the file is correct it may be copied to the a new file with the name **smsclF.cfg**
NCC Glossary of Terms

AAA

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

Connection
Transport level link between two peers, providing for multiple sessions.

cron
Unix utility for scheduling tasks.

crontab
File used by cron.

DB
Database

Diameter
A feature rich AAA protocol. Utilises SCTP and TCP transports.

DTMF
Dual Tone Multi-Frequency - system used by touch tone telephones where one high and one low frequency, or tone, is assigned to each touch tone button on the phone.

EDR
Event Detail Record

Note: Previously CDR. The industry standard for CDR is EDR (Event Detail Record). Over time EDR will replace CDR in the NCC documentation.

EMI
Exchange Message Interface protocol

FDA
First Delivery Attempt - the delivery of a short message directly to the SME rather than relaying it through the MC.

GSM
Global System for Mobile communication.
It is a second generation cellular telecommunication system. Unlike first generation systems, GSM is digital and thus introduced greater enhancements such as security, capacity, quality and the ability to support integrated services.

**HLR**

The Home Location Register is a database within the HPLMN (Home Public Land Mobile Network). It provides routing information for MT calls and SMS. It is also responsible for the maintenance of user subscription information. This is distributed to the relevant VLR, or SGSN (Serving GPRS Support Node) through the attach process and mobility management procedures such as Location Area and Routing Area updates.

**HTML**

HyperText Markup Language, a small application of SGML used on the World Wide Web. It defines a very simple class of report-style documents, with section headings, paragraphs, lists, tables, and illustrations, with a few informational and presentational items, and some hypertext and multimedia.

**IN**

Intelligent Network

**IP**

1) Internet Protocol

2) Intelligent Peripheral - This is a node in an Intelligent Network containing a Specialized Resource Function (SRF).

**IP address**

Internet Protocol Address - network address of a card on a computer

**ISDN**

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

**ISUP**

ISDN User Part - part of the SS7 protocol layer and used in the setting up, management, and release of trunks that carry voice and data between calling and called parties.

**ITU**

International Telecommunication Union

**MAP**

Mobile Application Part - a protocol which enables real time communication between nodes in a mobile cellular network. A typical usage of the protocol would be for the transfer of location information from the VLR to the HLR.

**MC**

Message Centre. Also known as SMSC.
**Messaging Manager**

The Messaging Manager service and the Short Message Service components of Oracle Communications Network Charging and Control product. Component acronym is MM (formerly MMX).

**MM**

Messaging Manager. Formerly MMX, see also XMS (on page 29) and Messaging Manager (on page 27).

**MS**

Mobile Station

**MSC**

Mobile Switching Centre. Also known as a switch.

**MSISDN**

Mobile Station ISDN number. Uniquely defines the mobile station as an ISDN terminal. It consists of three parts; the country code (CC), the national destination code (NDC) and the subscriber number (SN).

**MT**

Mobile Terminated

**MTP**

Message Transfer Part (part of the SS7 protocol stack).

**Oracle**

Oracle Corporation

**PC**

Point Code. The Point Code is the address of a switching point.

**SCCP**

Signalling Connection Control Part (part of the SS7 protocol stack).

**SCTP**

Stream Control Transmission Protocol. A transport-layer protocol analogous to the TCP or User Datagram Protocol (UDP). SCTP provides some similar services as TCP (reliable, in-sequence transport of messages with congestion control) but adds high availability.

**Session**

Diameter exchange relating to a particular user or subscriber access to a provided service (for example, a telephone call).
SGML

SLC
Service Logic Controller (formerly UAS).

SLEE
Service Logic Execution Environment

SME
Short Message Entity - an entity which may send or receive Short Messages. It may be located in a fixed network, a mobile, or an SMSC.

SMS
Depending on context, can be:
- Short Message Service
- Service Management System platform
- NCC Service Management System application

SMSC
Short Message Service Centre - stores and forwards a short message to the indicated destination subscriber number.

SN
Service Number

SRF
Specialized Resource Function - This is a node on an IN which can connect to both the SSP and the SLC and delivers additional special resources into the call, mostly related to voice data, for example play voice announcements or collect DTMF tones from the user. Can be present on an SSP or an Intelligent Peripheral (IP).

SS7
A Common Channel Signalling system used in many modern telecoms networks that provides a suite of protocols which enables circuit and non circuit related information to be routed about and between networks. The main protocols include MTP, SCCP and ISUP.

SSN
Subsystem Number. An integer identifying applications on the SCCP layer.
For values, refer to 3GPP TS 23.003.

SSP
Service Switching Point
**System Administrator**

The person(s) responsible for the overall set-up and maintenance of the IN.

**TCAP**

Transaction Capabilities Application Part – layer in protocol stack, message protocol.

**TCP**

Transmission Control Protocol. This is a reliable octet streaming protocol used by the majority of applications on the Internet. It provides a connection-oriented, full-duplex, point to point service between hosts.

**UPC**

USSD Portal Components

**USSD**

Unstructured Supplementary Service Data - a feature in the GSM MAP protocol that can be used to provide subscriber functions such as Balance Query and Friends and Family Access.

**VLR**

Visitor Location Register - contains all subscriber data required for call handling and mobility management for mobile subscribers currently located in the area controlled by the VLR.

**VWS**

Oracle Voucher and Wallet Server (formerly UBE).

**XMS**

Three letter code used to designate some components and path locations used by the Oracle Communications Network Charging and Control Messaging Manager (on page 27) service and the Short Message Service. The published code is MM (on page 27) (formerly MMX).
## Index

### A
- AAA • 25
- About Installation and Removal • 23
- About This Document • v
- Architectural overview • 2, 4
- Audience • v

### B
- Background Processes • 17

### C
- CC • 25
- cdrInterfaceName • 10
- cdrLibrary • 11
- classDefault • 11
- CONF_FILE • 7
- Configuration • 7, 18, 22
- Configuration file example • 15
- Configuration file format • 9
- Configuration file location • 10
- Configuration file structure • 10
- Configuring the Environment • 7
- Configuring the installation • 24
- Configuring the smmclF.cfg • 10, 22
- Connection • 25
- Connection failure • 3
- Connection process • 3
- connectTimeout • 11
- Copyright • ii
- cron • 25
- crontab • 25

### D
- DB • 25
- Diameter • 25
- Document Conventions • vi
- DTMF • 25

### E
- Editing the file • 9
- EDR • 25
- EMI • 25
- eserv.config Configuration • 9, 18
- eserv.config files delivered • 10
- ESERV_CONFIG_FILE • 8
- Example eserv.config file • 10
- Example notificationIF section • 10, 21

### F
- failoverInterval • 11
- FDA • 25
- fromAddress • 18

### G
- GSM • 25

### H
- HLR • 26
- HTML • 26

### I
- idleTimeout • 4, 12
- IN • 26
- Installation and Removal Overview • 23
- Installation Prerequisites • 23
- Introduction • 9, 23
- IP • 26
- IP address • 26
- ip1 • 11, 12
- ip2 • 12
- ISDN • 26
- ISUP • 26
- ITU • 26

### L
- Location • 17, 22
- Location of eserv.config • 8
- LOG_FILE • 7, 22
- Login event • 4
- loginTimeout • 13

### M
- MAP • 26
- maxSmsPerSecond • 13
- MC • 26
- Message flows • 4
- Message queuing • 3, 14
- Messaging Manager • 27, 29
- MM • 27, 29
- MS • 27
- MSC • 27
- msgLife • 3, 13
- MSISDN • 27
- MT • 27
- MTP • 27
- Multiple instance configuration • 8, 22

### N
- Notification Interface • 4
- notificationIF • 17

### O
- Oracle • 27
- oracleLogin • 18
- Output • 22
- Overview • 1, 7, 10, 17, 23
NCC SMS Center Technical Guide

P
Parameters • 10, 18
password • 13
PC • 27
Post-installation Configuration • 24
Prerequisites • v
Purpose • 3, 17, 22

Q
queueMaximum • 3, 14

R
Related documents • v
retryInterval • 14
retryNumber • 14

S
SCCP • 27
Scope • v
SCTP • 27
Session • 27
Set SMSC database privilege • 23
SGML • 28
SLC • 28
SLEE • 28
SLEE.cfg • 7, 8
SME • 28
SMS • 28
SMS Interface component package • 23
SMS Interface System Environment • 1
SMSC • 28
smsIF • 18
smsInterface • 8, 10, 11, 12, 13, 14, 15, 22
smsInterface Connection Management • 3
SN • 28
SRF • 28
SS7 • 28
SSN • 28
SSP • 28
Startup • 8, 17, 22
Startup configuration • 7
Sub-system components • 2
System Administrator • 29
System Overview • 1

T
TCAP • 29
TCP • 29
tracingDirectory • 14
tracingEnabled • 15
Typographical Conventions • vi

U
UPC • 29
userId • 15
USSD • 29

W
What is the SMS interface? • 1

X
XMS • 27, 29
xmsDestNPI • 19
xmsDestTON • 19
xmsDirectFromPrefix • 19
xmsFDAFromPrefix • 19
xmsIWrapperIfName • 20
xmsOrigNPI • 20
xmsOrigTON • 20
xmsPC • 21
xmsSSN • 21

V
VLR • 29
VWS • 29