# Oracle® Communications Network Charging and Control

## CAP3 SMS Gateway Technical Guide

Release 15.2

January 2026

# Copyright

# Contents

# About This Document

## Scope

The scope of this document includes all the information required to install, configure and administer the Oracle Communications Network Charging and Control CAP3 SMS Gateway application.

## Audience

This guide was written primarily for system administrators and persons installing, configuring and administering the CAP3 SMS Gateway application. However, sections of the document may be useful to anyone requiring an introduction to the application.

## Prerequisites

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

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

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

## Related Documents

The following documents are related to this document:

- *Service Logic Execution Environment Technical Guide*
- *Short Message Charging Bundle User's & Technical 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. |
| *Italics* | Name of a document, chapter, topic or other publication. |
| | Emphasis within text. |
| **Button** | The name of a button to click or a key to press. |
| | **Example:**   To close the window, either click **Close**, or press **Esc**. |
| **Key+Key** | Key combinations for which the user must press and hold down one key and then press another. |
| | Example: **Ctrl+P** or **Alt+F4**. |
| `Monospace` | Examples of code or standard output. |
| **`Monospace Bold`** | Text that you must enter. |
| *variable* | Used to indicate variables or text that should be replaced with an actual value. |
| **menu option > menu option >** | Used to indicate the cascading menu option to be selected. |
| | Example: **Operator Functions > Report Functions** |
| [hypertext link](#) | Used to indicate a hypertext link. |

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

# System Overview

## CAP3 SMS Gateway

### Description

The Oracle Communications Network Charging and Control CAP3 SMS Gateway (capgw) is a SLEE application that translates CAP v3 SMS operations to the equivalent CAP non-SMS operations used by Advanced Control Services ACS. It is used in conjunction with the Short Message Charging Bundle SMCB application. For a detailed description, refer to *CAP3 SMS Gateway* (on page 13) background process.

### CAP3 SMS Gateway context

This diagram illustrates the logical message flow through capgw.
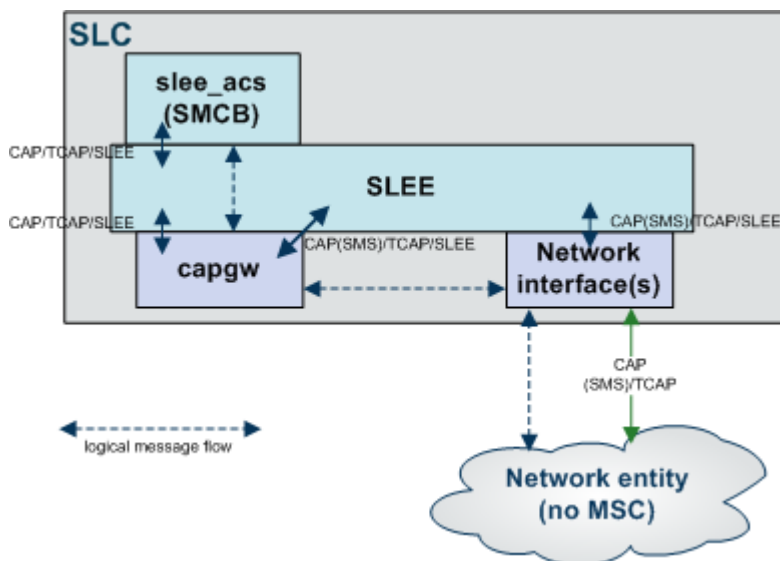
# Message Flows

## CAP3 SMS Gateway CapContinueSMS

Here is an example message flow where CAP3 SMS Gateway returns a CapContinueSMS.
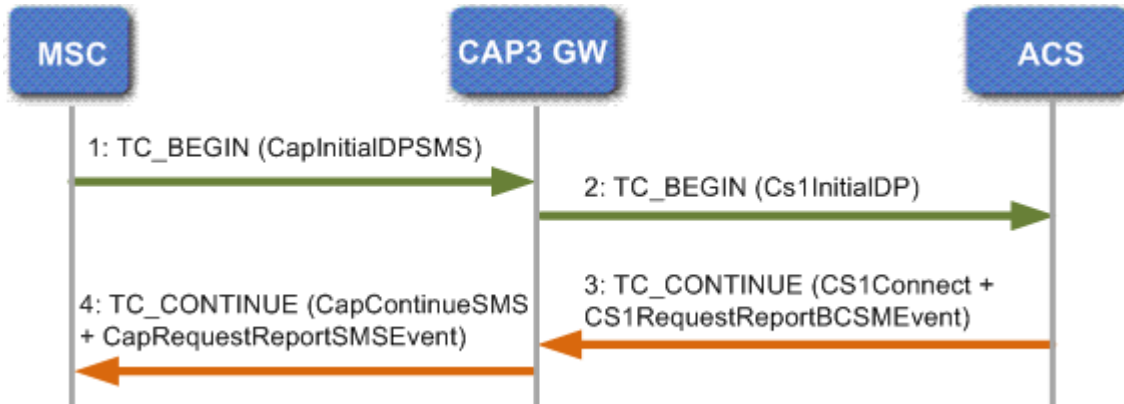


**Notes:**

1   Network element triggers the SLC using IDPSMS containing calling and destination numbers.
2   Message is routed to the CAP3 SMS Gateway which triggers the ACS application by mapping parameters in the IDPSMS to an IDP.
3   Control plan is triggered. The control does not change the calling and/or destination numbers. This is sent in a Connect operation back to CAP3 SMS Gateway.
4   CAP3 SMS Gateway returns a ContinueSMS operation back to the network element.

## CAP3 SMS Gateway CapConnectSMS

For some deployments, the triggered application (for example, Messaging Manager) has the capability to be able to change the calling and destination numbers of SMS messages (because of, number portability, for example) before they are delivered to the SMSC. In this case a connect will be sent.

Here is an example message flow where CAP3 SMS Gateway returns a CapConnectSMS.
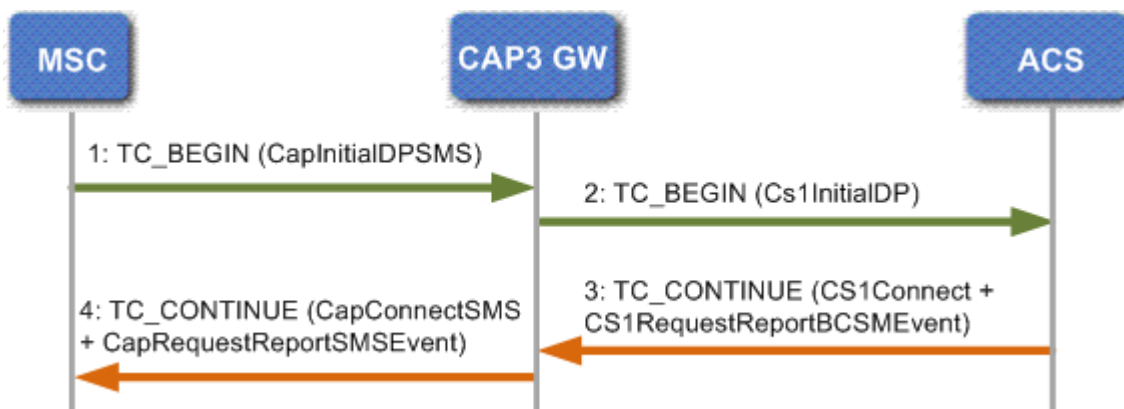


**Notes:**

1   Network element triggers the SLC using IDPSMS containing calling and destination numbers.
2   Message is routed to the CAP3 SMS Gateway which triggers the ACS application by mapping parameters in the IDPSMS to an IDP.
3   Control plan is triggered. The service designer elects to change the calling and/or destination numbers. These are sent in a Connect operation back to CAP3 SMS Gateway.

**4** CAP3 SMS Gateway returns a ConnectSMS operation back to the network element.

## CAP3 SMS Gateway CapReleaseSMS

Here is an example message flow where CAP3 SMS Gateway returns a CapReleaseSMS.

This may occur in cases such as insufficient funds.
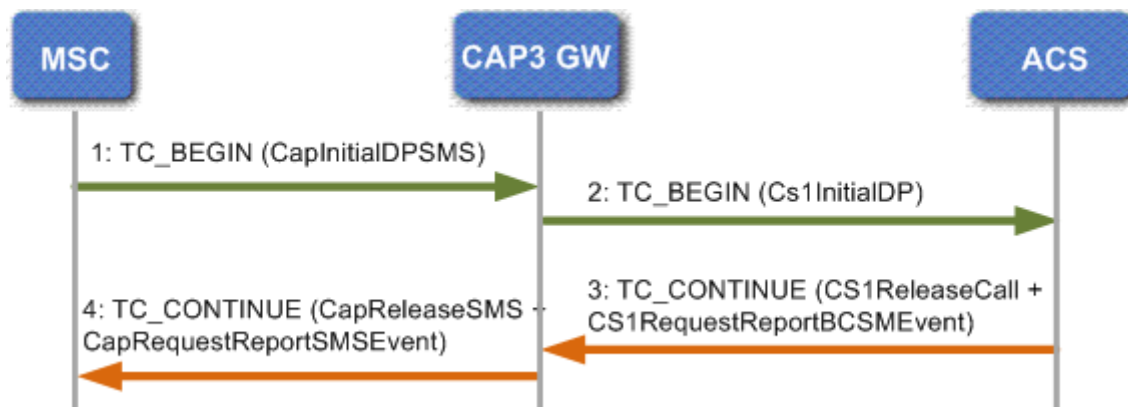


# Incoming Session Data

The CAP3 SMS Control Agent determines incoming data values from ACS profile tag values. Profile tags are stored in profile blocks and configured in the ACS UI. See the discussion on ACS configuration in *ACS User's Guide* for more information.

## IDP Extensions Parameter

The CAP3 SMS Gateway passes a profile block to ACS in the `InitalDP.extensions` parameter as extension type 701. The slee_acs process makes the profile tags from this profile block available to a feature node through the **Session data**, **Incoming Session Data** option in the feature node configuration window.

### IDP Profile Tag

The following table lists the information written into IDP extension type 701 by the CAP3 SMS Gateway.

| Profile Tag Number | Name |
|---|---|
| 8126464 | SMSC Address |

# 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.

## Configuration Overview

### Introduction

This chapter provides a high level overview of how the Oracle Communications Network Charging and Control CAP3 SMS Gateway application, provided in the capgwScp package, is configured.

There are configuration options which are added to the configuration files that are not explained in this chapter. These configuration options are required by the applications and should not be changed.

### Configuration process overview

This table describes the steps involved in configuring the applications for the first time.

| Stage | Description |
|---|---|
| 1 | The environment the applications will run in must be configured correctly. This includes if the directory the packages were installed into was not the recommended directory, setting the root directory. |
| 2 | The **SLEE.cfg** file must be configured for the SLCs. This file is automatically updated during installation. |
| 3 | The **eserv.config** file must be configured for the SLCs. This file is automatically updated during installation. |

### Configuration components

The CAP v3 SMS Gateway is configured by the following components:

| Component | Locations | Description | Further Information |
|---|---|---|---|
| **eserv.config** | All SLCs | In the **eserv.config** file, CAP3 SMS Gateway is configured by the `capgw` section | **Example capg***w parameters* (on page 6) |

| Component | Locations | Description | Further Information |
|-----------|-----------|-------------|---------------------|
| **SLEE.cfg** | All SLCs | Sets up the capgw service, service keys and applications on the SLEE. | *SLEE configuration changes* (on page 6) |

# Configuring capgw

## Introduction

The CAP3 SMS Gateway (capgw) is a SLEE application, therefore it must be configured in the **SLEE.cfg** configuration file and also in the **eserv.config** configuration file.

## SLEE configuration changes

The post installation scripts automatically run during the installation of the capgw package will make all necessary configuration changes to the SLEE configuration file to include the capgw application.

The **SLEE.cfg** is located in **/IN/service_packages/SLEE/etc**.

In the `capgw` section of the **SLEE.cfg** file, ensure that the following entries are present - if not, they must be added.

```
APPLICATION=capgw capgw.sh /IN/service_packages/CAP3_GW/bin 1 1
SERVICE=capgw 1 capgw capgw
SERVICEKEY=INTEGER 15 capgw
```
For further details on the configuration of the SLEE application, see *SLEE Technical Guide*.

## eserv.config changes

The post installation scripts that run automatically during the installation of the capgwScp package make all the necessary configuration changes to the **eserv.config** configuration file to include the capgw application.

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

## Example capgw parameters

The following example shows the full list of entries in the `capgw` section of **eserv.config** file. These are created from your responses to the configuration questions asked during the installation of the capgwScp package. Some of these entries may not be present if the default values are being used.

```
capgw = {
    destinationServiceKey = 11
    #calling_noa = 2
    #calling_plan = 1
    #calling_screen = 0
    #calling_present = 2
    #calling_innorni = true
    called_noa_map = "0,4,3,5,1,0,0,0"
    calling_noa_map = "0,4,3,5,1,0,0,0"
    # idps_use_noa_maps = false

    connectSMS = {
        # smscNoa = 2
        # smscPlan = 1
    }

    tracing = {
        enabled = false
        outputFile = "/tmp/smsTrace.trc"
```

```
        callsPerMinute = 2
        outputFileCycle = 1024
        maxFileSizeKB = 0 # unlimited
        origAddress = [
            # "00123456789"
        ]
        destAddress = [
            # "00987654321"
        ]

    }}
}
```

## Parameters

The capgw supports the following parameters.

`called_noa_map`

| | |
|---|---|
| **Syntax:** | `called_noa_map = "a,b,c,d,e,f,g,h"` |
| **Description:** | Associate called party address Type Of Number values in CAP SMS operations with NoA values used by SM-CB. |
| | Value is an array of SM-CB NoA values indexed by Type of Number value. |
| **Type:** | String |
| **Optionality:** | Optional (default used if not set). |
| **Allowed:** | Where *a-h* have values 0-7. |
| **Default:** | No mapping and values passed through CAP gateway transparently |
| **Notes:** | |
| **Example:** | A mapping of "0,4,3,5,1,0,0,0" will map a ToN 1 to SM-CB NoA 4 and used in reverse will map SM-CB NoA 3 to ToN 2. <br> `called_noa_map = "0,4,3,5,1,0,0,0"` |

`calling_innorni`

| | |
|---|---|
| **Syntax:** | `calling_innorni = true|false` |
| **Description:** | The calling innOrNI value in InitialDPs to SM-CB. |
| **Type:** | Boolean |
| **Optionality:** | Optional (default used if not set). |
| **Allowed:** | true, false |
| **Default:** | true |
| **Notes:** | IDP override - If set this will be used instead of what was supplied in the InitialDPSMS. |
| **Example:** | `calling_innorni = false` |

`calling_noa`

| | |
|---|---|
| **Syntax:** | `calling_noa = num` |
| **Description:** | The calling NoA in InitialDPs sent to SM-CB. |
| **Type:** | Integer |
| **Optionality:** | Optional (default used if not set). |
| **Allowed:** | |
| **Default:** | no override |

| | |
|---|---|
| **Notes:** | This overrides the value received in the InitialDPSMS and the `calling_noa_map` (on page 8) mapping. |
| **Example:** | `calling_noa = 2` |

## calling_noa_map

| | |
|---|---|
| **Syntax:** | `calling_noa_map = "a,b,c,d,e,f,g,h"` |
| **Description:** | Associate calling party address NoA values in SMS operations with NoA values used by SM-CB. |
| | Value is array of SM-CB NoA values indexed by SMS NoA. |
| **Type:** | String |
| **Optionality:** | Optional (default used if not set). |
| **Allowed:** | Where *a-h* have values 0-7. |
| **Default:** | No mapping and values passed through CAP gateway transparently |
| **Notes:** | |
| **Example:** | A mapping of "0,4,3,5,1,0,0,0" will map SMS NoA 1 to SM-CB NoA 4 and used in reverse will map SM-CB NoA 3 to SMS NoA 2. |
| | `calling_noa_map = "0,4,3,5,1,0,0,0"` |

## calling_plan

| | |
|---|---|
| **Syntax:** | `calling_plan = num` |
| **Description:** | The calling party numbering plan from the original InitialDPs to SM-CB. |
| **Type:** | Integer |
| **Optionality:** | Optional (default used if not set). |
| **Allowed:** | |
| **Default:** | 1 |
| **Notes:** | IDP override - If set this will be used instead of what was supplied in the InitialDPSMS. |
| **Example:** | `calling_plan = 3` |

## calling_present

| | |
|---|---|
| **Syntax:** | `calling_present = num` |
| **Description:** | The calling present value in InitialDPs to SM-CB |
| **Type:** | Integer |
| **Optionality:** | Optional (default used if not set). |
| **Allowed:** | |
| **Default:** | 2 |
| **Notes:** | IDP override - If set this will be used instead of what was supplied in the InitialDPSMS. |
| **Example:** | `calling_present = 3` |

## calling_screen

| | |
|---|---|
| **Syntax:** | `calling_screen = num` |
| **Description:** | Calling party screening indicator. |
| **Type:** | Integer |
| **Optionality:** | Optional (default used if not set). |
| **Allowed:** | |

| | |
|---|---|
| **Default:** | 0 |
| **Notes:** | IDP override - If set this will be used instead of what was supplied in the InitialDPSMS. |
| **Example:** | `calling_screen = 1` |

`connectSMS`

| | |
|---|---|
| **Syntax:** | `connectSMS = {connect_parameters}` |
| **Description:** | The connect SMS operation configuration to support sending a CAP3 ConnectSMS operation back to the network element that originated the dialog. |
| **Type:** | Parameter section |
| **Optionality:** | Optional (default used if not set). |
| **Allowed:** | |
| **Default:** | |
| **Notes:** | |
| **Example:** | `connectSMS = {`<br>`    smscNoa = 2`<br>`    smscPlan = 1`<br>`}` |

`destinationServiceKey`

| | |
|---|---|
| **Syntax:** | `destinationServiceKey = skey` |
| **Description:** | Incoming IDP SMS messages are converted to IDPs and sent to this service key. |
| **Type:** | Integer |
| **Optionality:** | Optional (default used if not set). |
| **Allowed:** | |
| **Default:** | 11 |
| **Notes:** | Normally SMCB is on this service key. |
| **Example:** | `destinationServiceKey = 1200` |

`idps_use_noa_maps`

| | |
|---|---|
| **Syntax:** | `idps_use_noa_maps = true\|false` |
| **Description:** | Map the NOAs in InitialDPs sent to SM-CB using the `called_noa_map` (on page 7) and `calling_noa_map` (on page 8) mappings. |
| **Type:** | Boolean |
| **Optionality:** | Optional (default used if not set). |
| **Allowed:** | true, false |
| **Default:** | false |
| **Notes:** | |
| **Example:** | `idps_use_noa_maps = false` |

`smscNoa`

| | |
|---|---|
| **Syntax:** | `smscNoa = NoA` |
| **Description:** | Specifies the NoA value to use for outgoing SMSC Address sent in connectSMS. |
| **Type:** | Integer (value 0-7) |
| **Optionality:** | Optional (default used if not set). |

| Allowed: | Nature of Address values |
|---|---|
| **Default:** | 2 |
| **Notes:** | This is a member of the `connectSMS` (on page 9) parameter group. |
| **Example:** | `smscNoa = 2` |

smscPlan

| **Syntax:** | `smscPlan = NoA` |
|---|---|
| **Description:** | Specifies the Numbering Plan value to use for outgoing SMSC Address sent in connectSMS. |
| **Type:** | Integer (value 0-15) |
| **Optionality:** | Optional (default used if not set). |
| **Allowed:** | |
| **Default:** | 1 |
| **Notes:** | This is a member of the `connectSMS` (on page 9) parameter group. |
| **Example:** | `smscPlan = 1` |

## Tracing parameters

The capgw configuration supports the following tracing parameters.

```
tracing = {
    enabled = false
    outputFile = "/tmp/smsTrace.trc"
    callsPerMinute = 2
    outputFileCycle = 1024
    maxFileSizeKB = 0 # unlimited
    origAddress = [
        # "00123456789"
    ]
    destAddress = [
        # "00987654321"
    ]

}
```

callsPerMinute

| **Syntax:** | `callsPerMinute = num` |
|---|---|
| **Description:** | The maximum number of calls per minute to trace. |
| **Type:** | Integer |
| **Optionality:** | |
| **Allowed:** | |
| **Default:** | 2 |
| **Notes:** | |
| **Example:** | `callsPerMinute = 2` |

destAddress

| **Syntax:** | `destAddress = ["add1","add2"]` |
|---|---|
| **Description:** | Array of destination addresses for which calls will be traced. |
| **Type:** | |
| **Optionality:** | |
| **Allowed:** | |

| | |
|---|---|
| **Default:** | You can specify "0" for all numbers. |
| **Notes:** | Prefixes in this list (*TonNpi* format) must be enclosed in double quotes; that is,"tttnnn" |
| **Example:** | destAddress = [<br>  "00987654321"<br>] |

## enabled

| | |
|---|---|
| **Syntax:** | enabled = *true*\|*false* |
| **Description:** | Determines whether or not to collect tracing details of numbers whose prefixes match those specified in the origAddress (on page 11) and destAddress (on page 10) parameter lists. |
| **Type:** | Boolean |
| **Optionality:** | Optional (default used if not set). |
| **Allowed:** | true, false |
| **Default:** | true |
| **Notes:** | |
| **Example:** | enabled = true |

## maxFileSizeKB

| | |
|---|---|
| **Syntax:** | maxFileSizeKB = *size* |
| **Description:** | The maximum file size, in KB. |
| **Type:** | Integer |
| **Optionality:** | |
| **Allowed:** | |
| **Default:** | 0 (unlimited) |
| **Notes:** | |
| **Example:** | maxFileSizeKB = 0 |

## origAddress

| | |
|---|---|
| **Syntax:** | origAddress = ["*add1*","*add2*"] |
| **Description:** | Array of originating addresses for which calls will be traced. |
| **Type:** | |
| **Optionality:** | |
| **Allowed:** | |
| **Default:** | You can specify "0" for all numbers. |
| **Notes:** | Prefixes in this list (*TonNpi* format) must be enclosed in double quotes; that is,"tttnnn" |
| **Example:** | origAddress = [<br>  "00987654321"<br>] |

## outputFile

| | |
|---|---|
| **Syntax:** | outputFile = "*file*" |
| **Description:** | The file that all tracing details are to be written to. The primary output file. |
| **Type:** | |

| | |
|---|---|
| **Optionality:** | |
| **Allowed:** | |
| **Default:** | "/tmp/smsTrace.trc" |
| **Notes:** | Once established, capgw cannot remove or rename the tracing output file. |
| **Example:** | `outputFile = "/tmp/smsTrace.trc"` |

## outputFileCycle

| | |
|---|---|
| **Syntax:** | `outputFileCycle = num` |
| **Description:** | Close and re-open the file every *N* calls. Checks for file size exceeded at this time also. |
| **Type:** | |
| **Optionality:** | |
| **Allowed:** | |
| **Default:** | 1024 |
| **Notes:** | Tracing buffered data is flushed to the output file before closing. |
| **Example:** | `outputFileCycle = 1024` |

## tracing

| | |
|---|---|
| **Syntax:** | `tracing = {tracing_parameters}` |
| **Description:** | The tracing feature allows individual calls to be identified and traced. |
| | To activate tracing, the tracing section of the capgw configuration must be enabled, and the prefixes to be traced must be defined. Tracing output is written to file at a specified interval. |
| **Type:** | |
| **Optionality:** | Optional |
| **Allowed:** | |
| **Default:** | |
| **Notes:** | |
| **Example:** | See *Tracing parameters* (on page 10). |

# 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.

## CAP3 SMS Gateway

### Purpose

The Oracle Communications Network Charging and Control CAP3 SMS Gateway (capgw) is a SLEE application that translates CAP v3 SMS INAP operations to the CAP v2 equivalent. It is intended for use in conjunction with the Short Message Charging Bundle application.

**Example:** When the capgw application receives an InitialDPSMS, it uses the parameters from the InitialDPSMS to construct an InitialDP. The InitialDP is then sent to the required destination.

### Startup

The capgw application is started using the **/IN/service_packages/CAP3_GW/bin/capgw.sh** startup script.

### Configuration

capgw accepts the following command line arguments.

**Usage:**

```
capgw -c configuration_section
```
The available parameter is:

| Parameters | Default | Description |
|---|---|---|
| -c *config_section* | capgw | Defines the configuration section in **eserv.config** to use for capgw, if you do not want to use the default. See *Configuring capgw* (on page 6) for details. |
| | | **Note:** The configuration section for capgw is automatically populated with the required configuration parameters and values by the installation process. |

## Signal handling

The capgw application accepts signal requests from the watchdog to shut down. It only handles the signal once. This means that the capgw reverts to the default signal handling after the signal is received.

**Example:** If the watchdog sends a SIGABORT to the capgw, the capgw tries to shutdown gracefully. In the mean time, the default SIGABORT handling is set so that if a second SIGABORT is sent, then the default behavior (exit with core) is used.

### Signals table

This table shows the signals that are accepted by the capgw application.

**Note:** Default signal handling specified by the operating system is used if any other signals are sent to capgw.

| Signal | Effect |
|--------|--------|
| SIGTERM | Shuts down the capgw application gracefully. |
| SIGHUP | The capgw application re-reads the configuration file. |
| SIGABORT | Shuts down the capgw application gracefully. |
| SIGUSR1 | Shuts down the capgw application gracefully. |
| SIGUSR2 | Shuts down the capgw application gracefully. |

## Failure

If the CAP3 SMS Gateway fails, then CAP v3 SMS INAP operations will not be translated to their CAP v2 equivalent, and no further calls will be received.

## Output

The capgw application writes error messages to the system log file.

## Number tracing

You can activate a number tracing facility for calls passing through the CAP3 SMS Gateway using the tracing parameter in the capgw section of the **eserv.config** file.

The tracing parameter is a number prefix. When it is present in the configuration file, calling party numbers that match the prefix are traced. The trace is placed in the following file: **/tmp**/*calling_party_number*.

## Switching tracing on and off

Follow these steps to switch tracing on or off.

| Step | Action |
|------|--------|
| 1 | Do one of the following:<br>• To turn tracing on, add the tracing parameter and the desired number prefix to the capgw section of the **eserv.config** file.<br>• To turn tracing off, remove the tracing parameter line from the capgw section of **eserv.config**. |
| 2 | Type kill -HUP *pid_of_capgw*<br><br>**Result:** Sends a SIGHUP to capgw, causing it to re-read the configuration file. |

# 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.

## 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

### CAP3 SMS Gateway packages

An installation of Oracle Communications Network Charging and Control CAP3 SMS Gateway includes the following packages, on the:

* SMS:
    * capgwSms
* SLC:
    * capgwScp

### Checking the installation

When the capgwScp package is installed correctly, and the SLEE is restarted, the following process will be run on the SLEE:

* capgw