Oracle® Communications
Network Charging and Control
Number Portability Service Pack Technical Guide
Release 5.0.1

June 2013
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.
NCC Glossary of Terms ........................................................................................................... 47
Index ...................................................................................................................................... 55
About This Document

Scope
The scope of this document includes all the information required to install, configure and administer the NP Service Pack application.

Audience
This guide is written primarily for system administrators and persons installing, configuring and administrating the NP Service Pack application. However, the overview sections of the document are useful to anyone requiring an introduction to the application.

Prerequisites
Although it is not a prerequisite to 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:

- NP Service Pack User's Guide
- CCS User's Guide
- CCS Technical Guide
- CCS Feature Node User’s Guide
- CPE User's Guide
- ACS User's Guide
- ACS Technical Guide
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><em>Italics</em></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. Example: 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. Example: <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. Example: <strong>Operator Functions &gt; Report Functions</strong>. Example: <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.

Icons

The following icons are used in NCC guides as visual cues to draw attention to important information.

- ![Note](image.png): Indicates useful and complementary information, such as an explanation, comment, or short expansion of the text object that is intended to catch your attention.

- ![Tip](image.png): Indicates practical but non-essential information that makes the solution easier to use or operate (for example, a keyboard shortcut, or alternative way to perform a step in a procedure).

- ![Warning](image.png): Indicates a caution. If this information is ignored, it could possibly cause irreversible damage to the equipment, data, or software.
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.

What is the NP Service Pack

What is the NP Service Pack

Introduction

The Number Portability Service Pack (NP Service Pack), accessed through SMS, provides flexibility and control over call routing of subscribers to the network.

Features

Features of NP Service Pack include:

- The ability to configure ported subscriber information from the screens and/or using the NP Provisioning Interface (PI) commands, and supporting control plan nodes to make services NP/MNP aware
- The ability to configure routing information based on operator assigned number prefixes and supporting control plan nodes to allow services to route to appropriate operators
- Functionality to allow the IN platform to satisfy MNP SRF requirements through a MAP based application (MTA) that can trigger a control plan when a supported message is received. The control plan can then perform MNP and through the appropriate node, instruct the MTA on how to respond (for example, relay, ack or error)
- Simple GTT functionality provided in the MTA for performing relay actions
- Call connection using least cost routing based on predefined carrier selection rule sets
- Home routing for calls within the network
- EDR generation for all calls processed by the network. The EDRs can be used for billing and reporting purposes.

NP macro nodes

The following feature nodes are available for the NP Service within the ACS Control Plan Editor:

- NP Destination Selection
• NP Least Cost Routing
• NP Home Routing
• NP Map Trigger

See *NCC Feature Nodes Reference Guide* for information on these nodes. See *CPE User's Guide* for information on using the ACS Control Plan Editor.

**NP PI commands**

A number of additional Provisioning Interface (PI) commands and utilities are supplied for NP. These supplement the existing PI commands available for CCS.

The NP PI commands can be used to add, modify and delete the following data through batch input:

• DN Ranges
• LCR Rule Sets and Rules
• Home Routing data

For an explanation of these commands, refer to *NP PI Commands Definition* document.

**Note:** For more information on the standard PI commands for CCS, refer to *PI User's & Technical Guide*. 
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 3
- acs.conf Configuration 4
- np_components.cfg Configuration 5
- SLEE.cfg Configuration 9
- Configuring EDR Collection 9

Configuration Overview

Introduction

This topic provides a high level overview of how the NP Service Pack is configured.

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

Configuration components

Number Portability Service Pack is configured by the following components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Locations</th>
<th>Description</th>
<th>Further Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>acs.conf</td>
<td>SMS and SLC</td>
<td>Configures the acsChassis which processes calls.</td>
<td>acs.conf Configuration (on page 4)</td>
</tr>
<tr>
<td>cdrIF.cfg</td>
<td>SLC</td>
<td>Configures the EDR interface.</td>
<td>Configuring EDR Collection (on page 9)</td>
</tr>
<tr>
<td>mta.cfg</td>
<td>SLC</td>
<td>Configures the MAP Trigger application.</td>
<td>mta (on page 16)</td>
</tr>
<tr>
<td>np_components.cfg</td>
<td>SLC</td>
<td>Configures the NP specific macro nodes.</td>
<td>np_components.cfg Configuration (on page 5)</td>
</tr>
<tr>
<td>SLEE.cfg</td>
<td>SLC</td>
<td>The &lt;SLEE _ac&gt;interface is configured to include the NP application, the MTA and the cdrIF interface.</td>
<td>SLEE.cfg Configuration (on page 9) and SLEE Technical Guide.</td>
</tr>
</tbody>
</table>
Chapter 2

Configuration file format

The NP Service Pack configuration files are located in the `/IN/service_packages/NP_SERVICE_PACK/etc` directory.

Editing configuration files

Open the configuration file on your system using a standard file editor. Do not use file editors such as Microsoft Word that attach Microsoft DOS or Windows line termination characters (that is, ^M) at the end of each row, as this will cause file errors when the application tries to read the configuration file.

Always keep a backup of your current configuration before making any changes to it, to ensure that you always have a working copy.

Loading configuration changes

If you change a configuration file, then you must send a signal (SIGHUP) to the relevant process, or restart the SLEE, to enable the new options to take effect.

acs.conf Configuration

Introduction

The `acs.conf` file must be configured on the SLC to enable NP to work.

The `acs.conf` file will be automatically updated during the installation of the npScp package.

Refer to ACS Technical Guide for more information on `acs.conf` configuration.

acsChassis configuration

Once the NP packages have been installed, you should check that the following lines have been added to the `acsChassis` section of `acs.conf`:

```
MacroNodePluginFile libNpSpecificMacroNodeLoader.so  # Inserted by npScp
ChassisPlugin libNpCpuChassisActions.so              # Inserted by npScp
```

Optionally this line can be added to the `acsChassis` section:

```
extensionNumber 0 101 inapNumber digits
```

When the MAP Trigger Application (MTA) receives a MAP message it builds an IDP using information from the received MAP message. If the above configuration is specified the first extension number in the IDP will be set to the original MSISDN (before normalization) contained in the received MAP message.

Optionally this service library can also be added to `acs.conf`:

```
acsServiceLibrary
    ExtraPORPlugin
    /IN/service_packages/NP_SERVICE_PACK/lib/libnpAcsSvcExtra.so:
```

This will enable the "Append URI" functionality in the Destination Selection Macro Node. See NP Service Pack User's Guide for more information.
np_components.cfg Configuration

Introduction

The np_components.cfg file must be configured to enable the NP feature nodes to work. An example np_components.cfg showing the configuration required is provided in /IN/service_packages/NP_SERVICE_PACK/etc/np_components.cfg.

np_components.cfg configuration

Once the NP packages have been installed, you must edit np_components.cfg to ensure it contains the following lines:

```plaintext
appID:NP
DNMinimal:6
DNMaximal:15
InternalDestination:Internal_Destination
DefaultDestination:Default_Destination
```

Example configuration file

Here is an example np_components.cfg file.

```plaintext
appID:NP
DNMinimal:6
DNMaximal:15
InternalDestination:Internal_Destination
DefaultDestination:Default_Destination
DNNoaPrefix:
#AddStopDigit
#UseCutAndPaste
CCSDeployment:no
SendCarrierCode:no
FormatCLI:no
#GenLCRCDR:no
ISUPTrunkInCDR:no
OPInRanges
#DoPQYZAfterRanges
#PQYZNumType:M
```

Parameters

This topic describes the parameters in np_components.cfg.

AddStopDigit

- **Syntax:** AddStopDigit
- **Description:** Determines whether a stop digit is added to the end of the pending termination number when the carrier-available branch is activated.
- **Allowed:** set (exists in config), not set (not in config)
- **Default:** False (not set)
- **Notes:** This parameter is used by the NP Least Cost Routing feature node.
- **Example:** AddStopDigit

appID

- **Syntax:** appID:value
- **Description:** Defines the NP application.
Chapter 2

Default: NP
Example: appID:NP

CCSDeployment
Syntax: CCSDeployment:yes|no
Description: Determines whether the NP_SERVICE_PACK will be installed and used in combination with the NCC CCS application.
Allowed: yes, no
Default: no
Example: CCSDeployment:yes

DefaultDestination
Syntax: DefaultDestination:value
Description: Defines the default routing destination name for NP Service Pack.
Default: Default_Destination
Notes: This parameter is used to initialize the NP Destination Selection feature node.
Example: DefaultDestination:Default_Destination

DNMaximal
Syntax: DNMaximal:value
Description: Defines the maximum length for the dialed number.
Allowed: A numeric value
Default: 15
Notes: The DNMaximal value must be less than the DNMinimal value. This parameter is used to initialize the Destination Selection feature node.
Example: DNMaximal:15

DNMinimal
Syntax: DNMinimal:value
Description: Defines the minimum length for the dialed number.
Allowed: A numeric value
Default: 6
Notes: The DNMinimal value must be less than the DNMaximal value. This parameter is used to initialize the Destination Selection feature node.
Example: DNMinimal:6

DNNoaPrefix
Syntax: DNNoaPrefix:value
Description: A string that will be prepended to the called number NOA digit that is added to the called number by the Least Cost Routing service.
Allowed: Up to 3 hexadecimal digits
Notes: This parameter is used by the NP Least Cost Routing feature node.
Example: DNNoaPrefix:
DoPQYZAfterRanges
Syntax: DoPQYZAfterRanges
Description: When set, if a matching operator cannot be found in the Dn Range table, then the PQYZ table will be searched.
Type: Boolean
Optionality: Optional.
Allowed: set (present in config), not set (not present in config).
Default: False (not set)
Notes: This parameter is used by the NP Destination Selection node.
Example: DoPQYZAfterRanges

FormatCLI
Syntax: FormatCLI:yes|no
Description: Determines whether the service will apply the preferred number formatting of the selected carrier to the CLI.
Allowed: yes, no
Default: yes
Notes: If set to no, the preferred number formatting of the selected carrier is only applied to the termination number, and the original CLI is preserved.
This parameter is used by the NP Least Cost Routing feature node.
Example: FormatCLI:yes

GenLCRCDR
Syntax: GenLCRCDR:yes|no
Description: Determines whether an EDR should be generated each time the Least Cost Routing functionality is successfully applied during the processing of a running call.
Allowed: yes, no
Default: no
Notes: Setting this parameter to yes will cause multiple EDRs being generated for the same call, one for each LCR attempt.
This parameter is used by the NP Least Cost Routing feature node.
Example: GenLCRCDR:yes

InternalDestination
Syntax: InternalDestination:value
Description: Defines the internal routing destination name for NP Service Pack.
Default: Internal_Destination
Notes: This parameter is used to initialize the NP Destination Selection feature node.
Example: InternalDestination:Internal_Destination

ISUPTrunkInCDR
Syntax: ISUPTrunkInCDR:yes|no
Description: Determines whether the service will include the ORIGTRUNK tag in the EDR.
Allowed: yes, no
Default: yes
Notes: The value of the ORIGTRUNK tag is set to the value of the locationNumber buffer. This parameter is used by the NP Least Cost Routing feature node.
Example: ISUPTrunkInCDR:yes

OPInRange
Syntax: OPInRange
Description: Determines whether operators will be stored in the dn range table
Type: Boolean
Optionality: Optional.
Allowed: set (present in config), not set (not in config)
Default: False (not set)
Notes: This parameter is used by the NP Destination Selection node.
Example: OPInRange

PQYZNumType
Syntax: PQYZNumType:value
Description: Defines whether to exit from the NP Destination Selection node along a fixed or a mobile branch.
Optionality: Optional (default used if not set).
Allowed: M (mobile), F (fixed)
Default: F
Notes: This parameter is used by the NP Destination Selection node.
Example: PQYZNumType:M

SendCarrierCode
Syntax: SendCarrierCode:yes|no
Description: Determines whether the service will include the carrier code in the DRA.
Allowed: yes, no
Default: yes
Notes: This parameter is used by the NP Least Cost Routing feature node.
Example: SendCarrierCode:yes

UseCutAndPaste
Syntax: UseCutAndPaste
Description: Determines whether INAP Cut and Paste is used in a Connect operation that is issued after the “Carrier available” branch of an LCR node has been selected.
Allowed: set (exists in config), not set (not in config)
Default: False (not set)
Notes: If the parameter is set, then Cut and Paste will be set to the length of the original un-normalized called number, so that all digits that were initially dialed are cut. This action will ensure that any subsequent digits in an overlap sending scenario will be preserved.
Example: UseCutAndPaste
SLEE.cfg Configuration

Introduction

The SLEE.cfg file must be configured to enable NP to work and to enable the collection of NP-specific EDRs.

The SLEE.cfg file will be automatically updated during the installation of the npScp package to set up the platform to use the MTA and the NP cdrIF interface.

Refer to:

- *SLEE Technical Guide* for more information on SLEE configuration.
- *TCAP Interfaces Technical Guide* for more information on SERVICEKEY format.

NP SLEE configuration

Once the npScp package has been installed, check that the following lines have been added to the SLEE.cfg file:

```plaintext
INTERFACE=cdrIF cdrIF /IN/service_packages/NP_SERVICE_PACK/bin EVENT # Inserted by npScp
APPLICATION=mtaApplication mta.sh /IN/service_packages/NP_SERVICE_PACK/bin 1 1 # Inserted by npScp
SERVICEKEY=INTEGER 0x10800000016 MTA # Inserted by npScp
SERVICEKEY=INTEGER 0x1080000002D MTA # Inserted by npScp
SERVICEKEY=INTEGER 0x10500000047 MTA # Inserted by npScp (GSM ATI)
SERVICEKEY=INTEGER 0x1050000090F MTA # Inserted by npScp (IS41 Location Request)
SERVICEKEY=INTEGER 0x10500000937 MTA # Inserted by npScp (IS41 SMS Request)
SERVICE=MTA 1 mtaApplication MTA # Inserted by npScp
```

Note:

- It is essential for the correct operation of the CDR interface that the SLEE Interface Type for cdrIF is always set to EVENT.

SERVICEKEY values are generated from the subsystem number and base key specified during the npScp installation process and therefore may be different to those shown here.

Configuring EDR Collection

Introduction

NP can be configured to produce EDRs for use in post processing as required. All EDR configuration is done in the cdrIF.cfg file. The EDRs are saved to file in a location specified in cdrIF.cfg.

EDR collection

Each call processed can produce a single EDR, or multiple EDRs, depending on the type and outcome of the call. As a minimum, each call invokes either an ACS or a CCS service, producing one ACS/CCS EDR for every termination attempt.

Where Least Cost Routing (LCR) is invoked, an LCR EDR is produced for every carrier selected for termination as part of the LCR service logic, in addition to the ACS/CCS EDR produced for every termination attempt. This means that the number of LCR EDRs and the number of ACS/CCS EDRs produced for the call is the same.

Format

EDRs are saved to file in tag/value pairs, separated by "|", in the following form:
Note: For ACS/CCS EDRS, the first value in the EDR is not a tag/value pair. It contains the name of the service that created the EDR (either ACS or CCS) only.

Parameters

The parameters used to configure EDR collection in the application are contained in the following sections in cdrIF.cfg:

- Defaults
- RecordDef

Defaults section

Here is an example of Defaults section configuration in cdrIF.cfg. This section defines default values for the EDR files.

```
Defaults {
    TempDirectory="/IN/service_packages/NP_SERVICE_PACK/cdr/temp"
    FileSize=4096
    FileDirectory="/IN/service_packages/NP_SERVICE_PACK/cdr/Closed"
}
```

**FileDialog**

**Syntax:** 
FileDialog = "dir"

**Description:** Specifies the final directory for storing the EDR files.

**Allowed:**

**Default:** 
"/IN/service_packages/NP_SERVICE_PACK/cdr/default/"

**Notes:** This is also the default final directory location for LCR EDRs if this parameter is not set in the RecordDef section.

**Example:**
FileDialog = 
"/IN/service_packages/NP_SERVICE_PACK/cdr/default/"

**FileSize**

**Syntax:** 
FileSize = size

**Description:** Specifies the maximum size (in bytes) for the EDR files.

**Default:** 
4096

**Example:**
FileSize = 4096

**TempDirectory**

**Syntax:** 
TempDirectory = "dir"

**Description:** Specifies the temporary directory where the EDR files are stored before being moved to their final location.

**Allowed:**

**Default:** 
"/IN/service_packages/NP_SERVICE_PACK/cdr/temp"

**Notes:**

**Example:**
TempDirectory = 
"/IN/service_packages/NP_SERVICE_PACK/cdr/temp"
RecordDef section

Here is an example of the RecordDef section in cdrIF.cfg. This section defines the location and characteristics of the LCR EDRs.

RecordDef "NPLCRCDR" {
  FileDesc {
    FileDirectory="/IN/service_packages/NP_SERVICE_PACK/cdr/Closed/
    FileName="LCR_%C(%y%M%d%h%m%s).cdr"
    FileHeader=""
    FileFooter=""
    RowHeader=""
   FileSize=1024000
    CdrFileMaxSize=1000
    CdrFileMaxAge=3600
    RowTrailer="\n"
    ColumnSepator="|
    RemoveNullColumns=true
  }
  ColumnDef {
    PID "%s"
    CID "%s"
    CUST "%s"
    SN "%s"
    TNNUM "%s"
    TNNOA "%s"
    CLI "%s"
    SK "%s"
    CPN "%s"
    PTI "%s"
    TIME "%s"
    CALLINGNUM "%s"
    CALLINGNOA "%s"
    ROUTEDEST "%s"
    CARRIERNAME "%s"
    CARRIERPOS "%s"
    ORIGTRUNK "%s"
  }
}

CdrFileMaxAge

Syntax: CdrFileMaxAge= seconds
Description: Specifies the maximum number of seconds that new records may be added to the EDR file before it is closed and moved to the directory location specified by the FileDirectory parameter.
Allowed: A numeric value.
Default: 3600
Notes: The file will already have been closed and moved, if CdrFileMaxSize is reached first.
Example: CdrFileMaxAge= 3600

CdrFileMaxSize

Syntax: CdrFileMaxSize= num
Description: Specifies the maximum number of records in the EDR file.
Allowed: A numeric value.
Default: 1000
Notes:
Example: CdrFileMaxSize= 1000

**ColumnDef**
This section specifies the LCR EDR fields (tags). For a description of each, see LCR EDRs.

**ColumnSeperator**
Syntax: ColumnSeperator = "char"
Description: Specifies the character to use to separate the EDR fields.
Allowed: Default: "|"
Notes: Example: ColumnSeperator = "|"

**FileDesc**
Identifies the file description section for the LCR EDRs.

**FileDirectory**
Syntax: FileDirectory = "dir"
Description: Specifies the directory where the LCR EDR files are finally stored.
Allowed: Default: "/IN/service_packages/NP_SERVICE_PACK/cdr/LCR/"
Notes: Example: FileDirectory = "/IN/service_packages/NP_SERVICE_PACK/cdr/LCR/"

**FileFooter**
Syntax: FileFooter = "text"
Description: Specifies the text to be appended to the end of each EDR file.
Allowed: Default: ""
Notes: Example: FileFooter = ""

**FileHeader**
This parameter defines the text to be appended to the beginning of each EDR file.
Default: ""
Allowed: -

**FileName**
Syntax: FileName = "name"
Description: Specifies the name format for the LCR EDR files.
Allowed: Default: "LCR_%C(%y%M%d%H%m%S).cdr"
Notes: %C(%y%M%d%h%m%s) represents the time stamp for when the EDR file was created.

Example file name: LCR_20060118105515.cdr

Example: FileName = "LCR_%C(%y%M%d%h%m%s).cdr"

### FileSize

**Syntax:** FileSize = size

**Description:** Specifies the maximum size, in bytes, for the LCR EDR files.

**Allowed:**

**Default:** 1024000

**Notes:**

**Example:** FileSize = 1024000

### RecordDef

Identifies the LCR EDR section.

**Default:** "NPLCRCDR"

**Allowed:**

**Notes:**

### RemoveNullColumns

**Syntax:** RemoveNullColumns = true|false

**Description:** Determines whether or not to include empty fields in the EDR record.

**Allowed:** true, false

**Default:** true

**Notes:**

**Example:** RemoveNullColumns = true

### RowHeader

**Syntax:** RowHeader = "text"

**Description:** Specifies the text to be appended to the beginning of each record in the EDR file.

**Allowed:**

**Default:** ""

**Notes:**

**Example:** RowHeader = ""

### RowTrailer

**Syntax:** RowTrailer = "\char"

**Description:** Specifies the character to use to separate the EDR records.

**Allowed:**

**Default:** "n"

**Notes:**

**Example:** RowTrailer = "\n"
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.

cdrIF 15  
mta 16  
npMfileCarrierDaemon 27  
npMfilePQYZDaemon 28  
npMfileRoutingDestinationDaemon 29  
npMfileRuleDaemon 30

cdrIF

Purpose

cdrIF is a SLEE interface. It manages the EDRs generated from calls processed by your network.

Startup

This process is started automatically by the SLEE. The following line must be included in the SLEE.cfg to start cdrIF:

```
INTERFACE=cdrIF cdrIF /IN/service_packages/NP_SERVICE_PACK/bin EVENT
```

For more information see SLEE.cfg Configuration (on page 9).

Note: If you do not specify the NP cdrIF interface in SLEE.cfg, then the standard cdrIF interface for ACS and CCS will be used for processing the EDRs.

Location

This process is located on the SLC.
Configuration

For details on the cdrIF configuration file and the available parameters, see *Configuring EDR Collection* (on page 9).

Failure

If cdrIF fails, no EDRs will be processed.

**mta**

Purpose

The MAP Trigger Application (MTA) receives a MAP message from the TCAP Interface. The MAP and SCCP parameters are matched against the configured trigger rules and the first rule to match causes an IDP to be sent to the service key defined in the trigger. An appropriate alarm is raised if there is a problem processing the MAP message.

If a MAP Trigger node is defined then the action configured in the node will be performed. For example the MTA could respond with a MAP response message or relay the MAP message at the SCCP level.

Startup

The mta is started automatically by the SLEE. The following lines must be included in the *SLEE.cfg* to start the mta:

```bash
APPLICATION=mtaApplication mta.sh /IN/service_packages/NP_SERVICE_PACK/bin 1 1
SERVICEKEY=INTEGER 0x10800000016 MTA
SERVICEKEY=INTEGER 0x1080000002D MTA
SERVICEKEY=INTEGER 0x10500000047 MTA
SERVICEKEY=INTEGER 0x10500000037 MTA
SERVICE=MTA 1 mtaApplication MTA
```

Note: The service key details are generated automatically during installation and therefore may be different from this example.

For more information see *SLEE.cfg Configuration* (on page 9).

Location

The mta is located on the SLC.

**mta.cfg configuration file**

The *mta.cfg* configuration file defines the list of triggers for the MAP messages received by the mta.

Here is an example *mta.cfg* file.

```json
mta = {
    triggers = [
        { name="OLONoNP", servicekey=20 },
        { name="VoiceNP", servicekey=100 },
        { name="ACS", servicekey=111 }
    ]
    triggerRules = [
        { msg="MAP_SRI", msisdn="99952044,99952,99904452,88", cdpn="99952,88",
          cdpnoa=4, cgpn="88852,77", cgpnoa=4, msisdnSuffixMinLength=2, msisdnSuffixMaxLength=10, trigger="VoiceNP" },
        { msg="MAP_SRI_SM", trigger="OLONoNP" }
    ]
}
```
{
  msg="MAP_SRI,MAP_SRI_SM", trigger="ACS" }
{
  msg="MAP_ATI", imsi="333, 444", trigger="ACS" }
{
  msg="LOCREQ", dgtsdial="18852044, 19052",
    dgtsDialSuffixMinLength=2, dgtsDialSuffixMaxLength=10,
    trigger="ACS" }
{
  msg="SMSREQ", min="788044, 6752", trigger="ACS" }
{
  msg="SMSREQ", imsi="55555", trigger="ACS" }
{
  msg="SMSREQ", mdn="123456", mdnSuffixMinLength=2,
    mdnSuffixMaxLength=10, trigger="ACS" }
}

pc_format = "ITU-TS"
local_pc = "7-243-3"
local_ssn = 8
local_gt_digits = "441473123456"
local_gt_noa = 2
local_numplan = 255
local_ri = "GT"

gttRules = [
  { prefix="44", pc = "7-243-3", ssn = 6,
    ni = true, numPlan = 255, transType = 255,
    minLength=2, maxLength=10,
    remove_chars = 1, add_chars = "44" }
  { prefix="55", pc = "1-255-2", ssn = 2 }
  { prefix="*", pc = "7-243-3", ssn = 8 }
]

map_response_imsi = [
  { prefix="44", imsi="441473123456" }
  { prefix="*", imsi="441473000000" }
]

normalisationRules = [
  { prefix = "88", noa = 4, remove_chars = 1, add_chars = "44" }
  { prefix = "44", remove_chars = 2, add_chars = "0" }
  { remove_chars = 2 , add_chars = "0" }
]
in_timeout = 10
sri_sm_dra_location = "NETWORKNODE_NUMBER"
map_sri_sm_dra_location= "MSC_NUMBER"

abort_code_mapping = [
  { code = "0-10", message = "A Critical Error Occurred", severity = "CRITICAL" }
  { code = "11", message = "An error occurred", severity = "ERROR" }
  { message = "A minor error occurred", severity = "WARNING" }
]

hop_count = 0
scscp_loop_compare_digits_only = true
scscp_loop_compare_pc_only = false
map_version = 1
msisdnn_sri_noa_override = 1
msisdnn_srism_noa_override = 1
msisdnn_ati_noa_override = 1
digitsdialled_locreq_noa_override = 1
mdn_smsreq_noa_override = 1
msisdnn_sri_plan_override = 1
msisdnn_srism_plan_override = 1
msisdnn_ati_plan_override = 1
prefer_imsi = true
Parameters

The mta accepts the following parameters.

abort_code_mapping

Syntax: abort_code_mapping = [ { code = "alarm_code", message = "alarm_message", severity = "severity_level" } ]

Description: List of abort code number mappings (returned by the Map Trigger Node) to alarm messages and severity levels.

Type: Parameter array.

Optionality: Optional.

Allowed: Default:

Notes:

- code - The alarm code to match. If the code is not defined then the rule will match any alarm. A range of values can also be specified.
- message (mandatory) - The message to print in the alarm.
- severity (mandatory) - The severity of the alarm. Valid values are: NOTICE, WARNING, ERROR, and CRITICAL.

Example:

```
abort_code_mapping = [
{ code = "0-10", message = "A critical error occurred", severity = "CRITICAL" }]
```

digitsdialled_locreq_noa_override

Syntax: digitsdialled_locreq_noa_override = noa

Description: Set the nature of address value to use for the digits dialed when forwarding an ANSI-41 LocationRequest with an SCCP relay.

Type: Integer

Optionality: Optional

Allowed: 0 - 255

Default:

Notes:

Example:

digitsdialled_locreq_noa_override = 4

gttRules

Syntax: gttRules = [ { prefix = "number", pc = "DRA_pointcode", ssn = number, remove_chars = number, add_chars = "number" } ]

Description: The global title translation rules. At least one GTT rule must be defined. GTT is performed on the digits returned in the destination routing address contained in the CONNECT message from ACS (sent in response to the IDP from the MTA).

Type: Parameter group

Optionality: Mandatory.

Allowed: Default:

Notes:

The following parameters from the group are mandatory:

- prefix - the rule containing the longest prefix match will be fired
- pc - the point code of the DRA will be updated with the defined point code (in pc_format), and
ssn - defines the subsystem number for the DRA.

The following parameters are optional:

- remove_chars - defines the number of characters to remove from the start of the DRA, and
- add_chars - list of characters to add to the start of the DRA after remove_chars has been applied.
- ni - set or unset the national indicator in the outgoing message.
- ri - set the routing indicator on the outgoing message (this will override the routing indicator value set by the call plan).
- numPlan - set the numbering plan of the global title in the outgoing message.
- trans_Type - set the translation type of the global title in the outgoing message.

Example:

```plaintext
gttRules = [
    { prefix="44", pc = "7-243-3", ssn = 6, ni = true, numPlan = 255, transType = 255, remove_chars = 1, add_chars = "44" }
]```

hop_count

Syntax: 

```
hop_count = hop_count
```

Description: 
The SUA hop counter will be set to this value if it is not already defined.

Type: 

```
Integer
```

Optionality: 

```
Optional (default used if not set).
```

Default: 

```
0
```

Notes: 

```
If the value of this parameter is zero (default) then a hop counter will not be added.
```

Example: 

```
hop_count = 70
```

in_timeout

Syntax: 

```
in_timeout = seconds
```

Description: 
The number of seconds to wait for a response from ACS after sending an IDP.

Type: 

```
Integer.
```

Optionality: 

```
Optional (default used if not set).
```

Default: 

```
10
```

Example: 

```
in_timeout = 10
```

local_gt_digits

Syntax: 

```
local_gt_digits = "global_title_digits"
```

Description: 
The global title digits of the NCC platform.

Type: 

```
String
```

Optionality: 

```
Mandatory unless local_pc and local_ssn is defined.
```

Default: 

```
None
```

Notes: 

```
Either both local_pc and local_ssn, or local_gt_digits must be defined. If required you can define all three parameters.
```

Example: 

```
local_gt_digits = "441473123456"
```
local_gt_noa
Syntax: \texttt{local_gt_noa = nature_of_address}
Description: The nature of address of the global title.
Type: Integer
Optionality: Optional (uses default if not defined).
Allowed: 
Default: 4
Notes: 
Example: \texttt{local_gt_noa = 4}

local_ni
Syntax: \texttt{local_ni = true\textbar false}
Description: The national indicator of the NCC platform.
Type: Boolean.
Optionality: Optional (default used if not set).
Allowed: true, false
Default: true (national indicator bit is set)
Example: \texttt{local_ni = true}

local_numplan
Syntax: \texttt{local_numplan = numbering_plan}
Description: The numbering plan of the global title.
Type: Integer.
Optionality: Optional.
Default: None
Example: \texttt{local_numplan = 10}

local_pc
Syntax: \texttt{local_pc = "local_point_code"}
Description: The local point code (that is, the PC value of the NCC platform).
Type: String
Optionality: Mandatory unless either \texttt{local_ssn} or \texttt{local_gt_digits} is defined. If \texttt{local_pc} is defined, then \texttt{local_snn} must also be defined.
Default: None.
Notes: The format used will depend on the \texttt{pc_format} configuration option.
Either both \texttt{local_pc} and \texttt{local_ssn}, or \texttt{local_gt_digits} must be defined.
If required you can define all three parameters.
Example: \texttt{local_pc = "7-243-3"}

local_ri
Syntax: \texttt{local_ri = "routing_indicator"}
Description: The routing indicator of the NCC platform.
Type: String.
Optionality: Optional (default used if not set).
Allowed: PC or GT
Default: GT
Example: \texttt{local\_ri = "PC"}

\textbf{local\_ssn}

\textbf{Syntax:} \texttt{local\_ssn = ssn\_number}
\textbf{Description:} The subsystem number of the NCC platform.
\textbf{Type:} Integer
\textbf{Optionality:} Mandatory unless \texttt{local\_gt\_digits} is defined. If \texttt{local\_ssn} is defined, then \texttt{local\_pc} must also be defined.
\textbf{Default:} None.
\textbf{Notes:} Either both \texttt{local\_pc} and \texttt{local\_ssn}, or \texttt{local\_gt\_digits} must be defined. If required you can define all three parameters.
\textbf{Example:} \texttt{local\_ssn = 8}

\textbf{local\_transtype}

\textbf{Syntax:} \texttt{local\_transtype = translation\_type}
\textbf{Description:} The translation type of the global title.
\textbf{Type:} Integer.
\textbf{Optionality:} Optional.
\textbf{Default:} None.
\textbf{Example:} \texttt{local\_transtype = 10}

\textbf{map\_response\_imsi}

\textbf{Syntax:} \texttt{map\_response\_imsi = [ { prefix="prefix\_number", imsi="imsi\_number" } ]}
\textbf{Description:} List of IMSI values for MAP responses. Maps the SCCP calling party number in the original MAP message (if present) to an IMSI value that will be present in any MAP responses sent by the mta.
\textbf{Type:} Parameter array
\textbf{Optionality:} Mandatory.
\textbf{Default:} None
\textbf{Notes:}
- \texttt{prefix} - The rule containing the longest prefix match will be fired. Prefix matching is performed on the SCCP calling party number in the original MAP message (if present). A default prefix of "*" will match any SCCP CgPN and will also match a message which does not have an SCCP CgPN defined. Only one default prefix can be defined.
- \texttt{imsi} - The imsi value to use in any generated MAP responses.
\textbf{Example:} \texttt{map\_response\_imsi = [ { prefix="44", imsi="441473123456" } ]}

\textbf{map\_version}

\textbf{Syntax:} \texttt{map\_version = map\_version\_number}
\textbf{Description:} Sets the MAP version to be used if a message is received with no application context.
\textbf{Type:} Integer
\textbf{Optionality:} Optional (default used if not set).
Allowed: 1, 2, or 3
Default: 1
Notes:
Example: map_version = 1

map1_sri_sm_dra_location
Syntax: map1_sri_sm_dra_location = "DRA_fields"
Description: Defines the field in the v1 MAP_SRI_SM response in which to place the
destination routing address from the CONNECT message received.
Type: String
Optionality: Optional (default used if not set).
Allowed:
• "ROAMING_NUMBER"
• "MSC_NUMBER"
• "FORWARDED_TO_NUMBER"
Default: MSC_NUMBER
Notes:
Example: map1_sri_sm_dra_location = "MSC_NUMBER"

mdn_smsreq_noa_override
Syntax: mdn_smsreq_noa_override = noa
Description: Set the nature of address value to use for the Mobile Directory Number when
forwarding an ANSI-41 SMSRequest with an SCCP relay.
Type: Integer
Optionality: Optional
Allowed: 0 - 255
Default:
Notes:
Example: mdn_smsreq_noa_override = 4

msisdn ATI noa override
Syntax: msisdn_ati_noa_override = noa
Description: Set the nature of address value to use for the MSISDN when forwarding a MAP
ATI with an SCCP relay.
Type: Integer
Optionality: Optional
Allowed: 0 - 255
Default:
Notes:
Example: msisdn_ati_noa_override = 4

msisdn ATI plan override
Syntax: msisdn_ati_plan_override = noa
Description: Set the numbering plan value to use for the MSISDN when forwarding a MAP ATI
with an SCCP relay.
Type: Integer
Optionality: Optional
msisdn_ati_plan_override

Syntax: msisdn_ati_plan_override = noa

Description: Set the nature of address value to use for the MSISDN when forwarding a MAP SRI with an SCCP relay.

Type: Integer

Optionality: Optional

Allowed: 0 - 255

Default:

Notes: 

Example: msisdn_ati_plan_override = 4

msisdn_sri_noa_override

Syntax: msisdn_sri_noa_override = noa

Description: Set the nature of address value to use for the MSISDN when forwarding a MAP SRI with an SCCP relay.

Type: Integer

Optionality: Optional

Allowed: 0 - 255

Default:

Notes: 

Example: msisdn_sri_noa_override = 4

msisdn_sri_plan_override

Syntax: msisdn_sri_plan_override = noa

Description: Set the numbering plan value to use for the MSISDN when forwarding a MAP SRI with an SCCP relay.

Type: Integer

Optionality: Optional

Allowed: 0 - 255

Default:

Notes: 

Example: msisdn_sri_plan_override = 4

msisdn_srism_plan_override

Syntax: msisdn_srism_plan_override = noa

Description: Set the numbering plan value to use for the MSISDN when forwarding a MAP SRI SM with an SCCP relay.

Type: Integer

Optionality: Optional

Allowed: 0 - 255

Default:

Notes: 

Example: msisdn_srism_plan_override = 4

msisdn_srism_noa_override

Syntax: msisdn_srism_noa_override = noa

Description: Set the nature of address value to use for the MSISDN when forwarding a MAP SRI SM with an SCCP relay.

Type: Integer

Optionality: Optional

Allowed: 0 - 255
normalisationRules

**Syntax:**

```
normalisationRules = [ { prefix = "number", noa = nature_of_address, minLength = number, maxLength = number, remove_chars = number, add_chars = "number" } ]
```

**Description:**
The normalization rules to apply to the MSISDN. The normalized MSISDN will then be sent in an IDP to the service key defined in the triggers parameter. If normalization rules are used then at least one character must be removed or added.

**Type:** Parameter array.

**Optionality:** Optional.

**Allowed:**

**Default:**

**Notes:**
- **prefix** - normalization will be applied to any any MSISDN DigitsDialled or MDN matching the specified prefix. The first prefix to match will be the rule that is applied. The order in which the rules are defined is therefore important.
- **noa** - if an NoA is defined, then the rule will only match if the NoA of the number matches the defined NoA. If no prefix and no NoA are defined then the rule will match any number.
- **minLength** – the rule will only match if the length of the number is equal to or greater than this value.
- **maxLength** – the rule will only match if the length of the number is less than this value.
- **remove_chars** - defines the number of characters to remove from the start of the DRA.
- **add_chars** - list of characters to add to the start of the DRA after remove_chars has been applied.
- either remove_chars or add_chars or both must be defined.

**Example:**

```
gttRules = [
    { prefix="88", noa = 4, minLength=2, maxLength=10, remove_chars = 1, add_chars = "44" }
]
```

**pc_format**

**Syntax:**

```
pc_format = "Point_Code_standard"
```

**Description:**
The point code standard used in the configuration

**Type:** String

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

**Allowed:** ITU-TS or ANSI

**Default:** ITU-TS

**Notes:**
- ITU-TS point codes will be encoded as a 14-bit structure consisting of a:
  - 3-bit zone identification, and 8-bit area/network identification
  - 3-bit signaling point identification.
- ANSI point codes will be encoded in a 24-bit structure consisting of an:
- 8-bit network identification
- 8-bit cluster identification, and
- 8-bit member identification.

Example: `pc_format = "ITU-TS"

**prefer_imsi**

**Syntax:** `prefer_imsi = true|false`

**Description:** If this parameter is set to true, a message received containing both an IMSI and a MIN, will have the IMSI stored in the IMSI field of the IDP and the IMSI field of the response. Else, the MIN will be present in the IMSI field of the IDP and the MIN field of the response.

**Type:** Boolean

**Optionality:** Optional

**Allowed:** true, false

**Default:** true

**Notes:**

Example: `msisdn_ati_plan_override = 4`

**sccp_loop_compare_digits_only**

**Syntax:** `sccp_loop_compare_digits_only = true|false`

**Description:** When performing an SCCP relay, the MTA checks that the outgoing message has enough differences from the incoming message to make a possible loop unlikely. If this item is set to true and both the incoming and outgoing destination routing indicators are set to global title, then the global title digits will be checked to make sure they are different. When set to false, the whole global title (including encoding scheme and NoA) will be checked.

**Type:** Boolean

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

**Allowed:** true, false

**Default:** true

**Notes:**

Example: `sccp_loop_compare_digits_only = true`

**sccp_loop_compare_pc_only**

**Syntax:** `sccp_loop_compare_pc_only = true|false`

**Description:** Used when performing an SCCP relay loop-check. If this item is set to false and the routing indicator on both the incoming and outgoing message is set to point code, then the point code and subsystem number will be checked to ensure that at least one of them has changed. If it is set to true then only the point code will be checked.

**Type:** Boolean

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

**Allowed:** true, false

**Default:** false

**Example:** `sccp_loop_compare_pc_only = true`
sri_sm_dra_location

Syntax:  

\texttt{sri\_sm\_dra\_location = "DRA\_field"}

Description:  

Defines the field in the v2/3 MAP\_SRI\_SM response in which to place the Destination Routing Address from the CONNECT message received.

Type:  

String

Optionality:  

Optional (default used if not set).

Allowed:

\begin{itemize}
  \item "NETWORKNODE\_NUMBER"
  \item "MSC\_NUMBER"
  \item "BOTH"
\end{itemize}

Default:  

"NETWORKNODE\_NUMBER"

Notes:

Example:  

\texttt{sri\_sm\_dra\_location = "NETWORKNODE\_NUMBER"}

triggerRules

Syntax:  

\texttt{triggerRules = [ \{ msg="map\_message\_list", msisdn="msisdn", msisdnSuffixMinLength=minLength, msisdnSuffixMaxLength=maxLength, dgtsdial="dgtsdial", dgtsDialSuffixMinLength=minLength, dgtsDialSuffixMaxLength=maxLength, mdn="mdn", mdnSuffixMinLength=minLength, mdnSuffixMaxLength=maxLength, min="min", imsi="imsi", cdpn="list\_of\_SCCP\_cdpns", cdpnoa="list\_of\_SCCP\_cdpn\_noas", cgpn="list\_of\_SCCP\_cdpns", trigger="name" \} ]}

Description:  

List of trigger rules against which the MAP and SCCP parameters from the MAP message will be matched. The first rule to match a message will be fired, therefore the order in which the rules are defined is important.

Type:  

Parameter array

Optionality:  

Mandatory.

Notes:  

Within the parameter group:

\begin{itemize}
  \item msg  defines a list of map messages that will match this rule.
  \item trigger  defines the trigger that will fire if the rule is matched.
\end{itemize}

The following parameters within the group are optional:

\begin{itemize}
  \item msisdn
  \item msisdnSuffixMinLength
  \item msisdnSuffixMaxLength
  \item dgtsdial
  \item dgtsDialSuffixMinLength
  \item dgtsDialSuffixMaxLength
  \item mdn
  \item mdnSuffixMinLength
  \item mdnSuffixMaxLength
  \item min
  \item imsi
  \item cdpn
  \item cgpn
  \item cdpnoa
Chapter 3

Background Processes

- cgpono

Example:

```
triggerRules = [
    { msg="MAP_SRI", msisdn="99952044,99952,99904452,88",
      cdpn="99952,88", cgponoa=4, cgpn="88852,77", cgponoa=4,
      msisdnSuffixMinLength=2, msisdnSuffixMaxLength=10,
      trigger="VoiceNP" }
    { msg="MAP_SRI,MAP_SRI_SM", trigger="ACS" }
    { msg="MAP_ATI", imsi="333, 444", trigger="ACS" }
    { msg="LOCREQ", dgtsDial="18852044, 19052",
      dgtsDialSuffixMinLength=2, dgtsDialSuffixMaxLength=10,
      trigger="ACS" }
    { msg="SMSREQ", min="788044, 6752", trigger="ACS" }
    { msg="SMSREQ", imsi="55555", trigger="ACS" }
    { msg="SMSREQ", mdn="123456", mdnSuffixMinLength=2,
      mdnSuffixMaxLength=10, trigger="ACS" }
]
```

```
triggers
Syntax:     triggers = [{ name = "name", servicekey = sknumber}]
Description: List of triggers for MAP messages and the service keys to send the translated IDPs to.
Type:       Parameter array
Optionality: Mandatory.
Notes:      The value defined for the name variable maps to the trigger variable in the triggerRules parameter array.
Example:    triggers = [{ name = "ACS", servicekey = 111}]
```

**Failure**

If the mta fails, then alarms will be raised to the syslog.

**Output**

The mta writes error messages to the system messages file and writes additional output to /IN/service_packages/NP_SERVICE_PACK/tmp/mta.log.

**npMfileCarrierDaemon**

**Purpose**

This process ensures the carrier data stored in memory is synchronized with the Carrier database table. The daemon ensures that the optimal performance, when querying the database and storing records, is maintained.

**Startup**

This process is started by entry scb1 in the inittab, via the shell script:

```
/IN/service_packages/NP_SERVICE_PACK/bin/startCarrierDaemon.sh
```
You can check if the process is running by using the Unix ps command.

To check the process, type:

```
ps -ef | grep npMfileCarrierDaemon
```

**Result:** The listed process is the compliant process.

**Note:** Placing the npMfileCarrierDaemon startup script in the inittab file ensures that if npMfileCarrierDaemon should die, it will be automatically restarted by the operating system within a few seconds.

### Shutdown

To terminate this process, use the Unix ps command to identify the process number and then kill it manually.

### Location

This process is located on the SLC.

### Parameters

There are no command line parameters for the npMfileRuleDaemon process.

### Failure

If this process fails, alarms will be raised to the syslog and updates to carrier records will not be available in the production system.

### Output

The npMfileCarrierDaemon writes error messages to the system messages file and writes additional output to:

```
/IN/service_packages/NP_SERVICE_PACK/npMfileCarrierDaemon.log
```

### npMfilePQYZDaemon

#### Purpose

This process ensures the PQYZ data stored in memory is synchronized with PQYZ data in the database tables. The daemon ensures that the optimal performance, when querying the database and storing records, is maintained.

#### Startup

This process is started by entry scb2 in the inittab, via the shell script:

```
/IN/service_packages/NP_SERVICE_PACK/bin/startPQYZDaemon.sh
```

You can check if the process is running by using the Unix ps command.

To check the process, type:

```
ps -ef | grep npMfilePQYZDaemon
```

**Result:** The listed process is the compliant process.
Note: Placing the npMfilePQYZDaemon startup script in the inittab file ensures that if npMfilePQYZDaemon should die, it will be automatically restarted by the operating system within a few seconds.

Shutdown

To terminate this process, use the Unix ps command to identify the process number and then kill it manually.

Location

This process is located on the SLC.

Parameters

There are no command line parameters for the npMfileRuleDaemon process.

Failure

If this process fails, alarms will be raised to the syslog and updates to PQYZ records will not be available in the production system.

Output

The npMfilePQYZDaemon writes error messages to the system messages file and writes additional output to:

/IN/service_packages/NP_SERVICE_PACK/npMfilePQYZDaemon.log

npMfileRoutingDestinationDaemon

Purpose

This process ensures the routing destination data stored in memory is synchronized with routing destination data in the database tables. The daemon ensures that the optimal performance, when querying the database and storing records, is maintained.

Startup

Two instances of this process are started by the inittab. Entry scb3 (the routing destination index entry) and entry scb4 (the routing destinations entry) are started via the following shell scripts:

/IN/service_packages/NP_SERVICE_PACK/bin/startRoutingDestinationDaemonIndex.sh
/IN/service_packages/NP_SERVICE_PACK/bin/startRoutingDestinationDaemonDest.sh

You can check if the processes are running by using the Unix ps command.

To check the processes, type:

ps -ef | grep npMfileRoutingDestinationDaemon

Result: The listed processes are the complier processes.

Note: Placing the npMfileRoutingDestinationDaemon startup scripts in the inittab file ensures that if the npMfileRoutingDestinationDaemon processes should die, they will be automatically restarted by the operating system within a few seconds.
Chapter 3

Shutdown

To terminate this process, use the Unix ps command to identify the process number and then kill it manually.

Location

This process is located on the SLC.

Parameters

npMfileRoutingDestinationDaemon accepts the following command line parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-oracleuser</td>
<td>-</td>
<td>Specifies the oracle user and password on the SLC.</td>
</tr>
<tr>
<td>oracle_user/password</td>
<td></td>
<td>Example: -oracleuser scp/scp</td>
</tr>
<tr>
<td>-gpnamfile_type</td>
<td>-</td>
<td>Specifies the type of routing destination mfile. Valid values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• destination</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• index</td>
</tr>
</tbody>
</table>

Failure

If this process fails, alarms will be raised to the syslog and updates to routing destination records will not be available in the production system.

Output

npMfileRoutingDestinationDaemon writes error messages to the system messages file and writes additional output to:

/IN/service_packages/NP_SERVICE_PACK/npMfileRoutingDestinationDaemon.log

npMfileRuleDaemon

Purpose

This process ensures the rule data held in memory is synchronized with rule data stored in the database tables. The daemon ensures that the optimal performance, when querying the database and storing records, is maintained.

Startup

This process is started by entry scb5 in the inittab via the shell script:

/IN/service_packages/NP_SERVICE_PACK/bin/startRuleDaemon.sh

You can check if the process is running using the Unix ps command.

To check the process, type:

ps -ef | grep npMfileRuleDaemon

Result: The listed process is the complier process.

Note: Placing the npMfileRuleDaemon startup script in the inittab file ensures that if the npMfileRuleDaemon processes should die, it will be automatically restarted by the operating system within a few seconds.
**Shutdown**
To terminate this process, use the Unix ps command to identify the process number and then kill it manually.

**Location**
This process is located on the SLC.

**Parameters**
There are no command line parameters for the npMfileRuleDaemon process.

**Failure**
If this process fails, alarms will be raised to the syslog and updates to rule records will not be available in the production system.

**Output**
npMfileRuleDaemon writes error messages to the system messages file and writes additional output to:

/IN/service_packages/NP_SERVICE_PACK/npMfileRuleDaemon.log
Overview

Introduction

This chapter explains the tools and utilities that are available.

In this chapter

This chapter contains the following topics.

NP EDRs 33
prunePortedNumbers.sh 38
Statistics 39

NP EDRs

Introduction

The NP Service Pack produces ACS and LCR EDRs, on the SLC, for use in post processing as required.

EDR collection

Each call processed can produce a single EDR, or multiple EDRs, depending on the type and outcome of the call. As a minimum, each call invokes either an ACS or a CCS service, producing one ACS/CCS EDR for every termination attempt.

Where Least Cost Routing (LCR) is invoked, an LCR EDR is produced for every carrier selected for termination as part of the LCR service logic, in addition to the ACS/CCS EDR produced for every termination attempt. This means that the number of LCR EDRs and the number of ACS/CCS EDRs produced for the call is the same.

NP EDR files

The EDRs are saved to file in a location specified in the cdrIF.cfg configuration file. For details, see Configuring EDR Collection (on page 9).

EDR files have the following names, depending on the EDR type.

<table>
<thead>
<tr>
<th>EDR Type</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACS</td>
<td>ACS_yyyymmddhhss_PID.cdr</td>
</tr>
<tr>
<td>LCR</td>
<td>LCR_yyyymmddhhss.cdr</td>
</tr>
</tbody>
</table>

Where:
- \( yyyymmddhhss \) = the date and time when the file was opened
- \( PID \) = the Unix process ID of the service instance that created the EDR file
Chapter 4

**Note:** For the LCR EDRs, the file name is configured in the `cdrIF.cfg` and may be different to the format described in this topic.

Later the files are moved by the cmnPushFiles process from each SLC to a configurable location on the SMS. This location is specified in the `cmnPushFilesStartup.sh` script located in the `NP_SERVICE_PACK/bin` directory of each SLC.

The cdr files moved from the SLCs are prefixed with the name of their corresponding SLC, that is, `LCR_YYYYMMDDHHMMSS.cdr` from 'SLC1' will be renamed to `SLC1_LCR_YYYYMMDDHHMMSS.cdr` in the SMS.

**Note:** If the location directories for the EDR files are changed manually to something different from the package defaults, the new location directories will have to be manually created in the system and the process that create and move the EDR files (cmnPushFiles and cdrIF) will have to be manually restarted.

**EDR fields**

EDRs are saved to file in tag/value pairs, separated by "|", each record separated by a Unix newline character, in the following form:

```plaintext
APP|tag1=value1|tag2=value2|
```

**Note:** The first field in the EDR is not a tag/value pair. It contains the name of the service (either ACS or CCS) that created the EDR. For more information about the format of SLC generated EDRs, see SLC Generated EDRs.

For LDR EDRs, the row trailer (newline) and column separator can be configured in the `cdrIF.cfg` and may be different to the default described in this topic.

**ACS tags**

The following ACS tags are generated in the EDR.

- AIDL
- CA
- CAET
- CBAT
- CCET
- CCTS
- CGNA
- CGNN
- CID
- CLI
- CPC
- CPN
- CPNI
- CPNN
- CPeI
- CS
- CUST
- FATS
- HTS
- LAC
- LGID
Chapter 4

- LPN (not applicable for NP)
- NOAT
- OA
- OCPI
- OTI
- PCNA (not applicable for NP)
- PTNA (not applicable for NP)
- RELC
- SK
- SN
- TCE
- TCS
- TFN
- TGNA (not applicable for NP)
- TN
- TPNI (not applicable for NP)

**Note:** These are standard tags, as described in the ACS EDR tags topic in *NCC Event Detail Record Reference Guide*.

**Example ACS EDRs**

**Example 1**
This example shows the output produced for a successful termination attempt EDR.

```
ACS|CID=61080|OA=0|OTI=0|CUST=1|SN=2125551212|TN=2125551212|
CGN=93933301|CLI=3135551212|SK=111|TCS=20051026133312|
TCE=20051026133317|LPN=|LAC=|CS=4|CPC=10|CC=|CPNI=0|PCNA=|TFN=0|
PTNA=|CGNA=|TGN=|TFN=ST-1,DDS-5,ATTP-6,ATTP-8,END-3|LGID=0|
CPN=atp|CAET=5|CCET=0.1|CA=2125551212|RELC=16|OCPI=|CPNN=1|
CGNN=4|CPFI=1|NOAT=2|CBAT=1|FATS=0|CCTS=20051026133312|
HTS=20051026133312|AIDL=
```

**Example 2**
This example shows the output produced for a failed termination attempt EDR.

```
ACS|CID=61080|OA=0|OTI=0|CUST=1|SN=2125551212|TN=2125551212|
CGN=93933301|CLI=3135551212|SK=111|TCS=20051026133312|
TCE=20051026133312|LPN=|LAC=|CS=0|CPC=10|CC=|CPNI=0|PCNA=|
TPNI=0|PTNA=|CGNA=|TGN=|TFN=ST-1,DDS-5,ATTP-6|LGID=0|CPN=atp|
CAET=0|CCET=0.0|CA=2125551212|RELC=25|OCPI=|CPNN=1|CGNN=4|
CPFI=1|NOAT=1|CBAT=0|FATS=0|CCTS=20051026133312|
HTS=20051026133312|AIDL=
```

**LCR EDR tags**

The following standard ACS tags are generated in the LCR EDR.

- CID
- CLI
- CPN
- CUST
- SK
- SN
Note: These are standard tags, as described in the ACS EDR tags topic in NCC Event Detail Record Reference Guide.

LCR EDRs do not contain any information on whether the termination attempt was successful or not – this data is stored in the ACS/CCS EDRs.

The LCR EDRs also contain the same CID field that can be used for correlation purposes with ACS/CCS EDRs and with other LCR EDRs.

The following LCR tags are unique to NP.

CALLINGNUM (lcr set calling number)

Description: The calling number set by the LCR service. This is the number that the service uses as the calling number when an attempt is made to connect the call.

Format: Integer. May be up to 32 digits long.

Version: NP 2.4.1.1

Notes: Example: CALLINGNUM=331111111111

CALLINGNOA (noa of callingnum)

Description: The nature of address of the CALLINGNUM.

Format: A single digit

Version: NP 2.4.1.1

Notes: Example: CALLINGNOA=4

CARRIERNAME (carrier name)

Description: The name of the selected carrier.

Format: String. May be up to 30 characters long.

Version: NP 2.4.1.1

Notes: Example: CARRIERNAME=Test

CARRIERPOS (position of carrier name in hunt list)

Description: The position of the selected carrier in the hunt list.

Format: Integer from 1 to 8.

Version: NP 2.4.1.1

Notes: Example: CARRIERPOS=2

ORIGTRUNK (idp location number content)

Description: Contains contents of location number field from the IDP.

Format: integer

Version: NP 2.4.1.1

Notes: Example: ORIGTRUNK=441473
PID (unix process ID)
Description: The Unix process ID of the service instance.
Format: Integer
Version: NP 2.4.1.1
Notes:
Example: PID=4355

PTI (product type ID)
Description: The product type ID for the CCS account type of the calling subscriber.
Format: Integer
Version: NP 2.4.1.1
Notes:
Example: PTI=2

ROUTEDEST (routing destination for call)
Description: The routing destination for the call.
Format: String. May be up to 64 characters long.
Version: NP 2.4.1.1
Notes:
Example: ROUTEDEST=Destination_3

TIME (creation timestamp of lcr edr)
Description: The timestamp for when the LCR EDR was created.
Format: Date
Version: NP 2.4.1.1
Notes:
Example: TIME=20051020154857

TNNUM (lcr terminating number)
Description: The terminating number set by the LCR service (the number that the service attempts to connect to).
Format: Number. May be up to 32 digits long.
Version: NP 2.4.1.1
Notes: The first digit of the number provides an indication of the NOA.
Example: TNNUM=4ABCD12AB987654321

TNNOA (noa of terminating number)
Description: The nature of address of the terminating number.
Format: Integer. A single digit.
Notes: This should correspond to the first digit of the TNNUM field.
Example: TNNOA=4
Chapter 4

Example LCR EDRs

Example 1
This example shows the output produced for a successful termination attempt EDR.

```
PID=4355|CID=142163|CUST=1|SN=987654321|TNNUM=4ABCD12AB987654321|TNNOA=4|CLI=33111111
1111111|SK=111|CPN=Rob_HR_LCR_plan|PTI=2|TIME=20051020154857|CALLINGNUM=33111111
11|CALLINGNOA=4|ROUTEDEST=Destination_3|CARRIERNAME=Test Carrier
2|CARRIERPOS=2|ORIGTRUNK=441473
```

Example 2
This example shows the output produced for a failed termination attempt EDR.

```
PID=4355|CID=142163|CUST=1|SN=987654321|TNNUM=4123412AB987654321|TNNOA=4|CLI=33111111
1111111|SK=111|CPN=Rob_HR_LCR_plan|PTI=2|TIME=20051020154857|CALLINGNUM=33111111
11|CALLINGNOA=4|ROUTEDEST=Destination_3|CARRIERNAME=Test Carrier
1|CARRIERPOS=1|ORIGTRUNK=441473
```

prunePortedNumbers.sh

**Purpose**

This script executes an SQL procedure to delete obsolete records from the DN range table. It looks for groups of records that have matching Dn start and Dn end dates and purges them according to these rules:

1. All records within the group that have an Activation Date in the past will be deleted except the currently active record (this is the one that is most recently in the past).
2. If the routing number for the currently active record is set to DEAD, then this record will also be deleted.

**Note:** For more information on the DEAD routing number, refer to the *NP Service Pack User's Guide*.

**Location**

prunePortedNumbers.sh is located in the `/IN/service_packages/NP_SERVICE_PACK/bin` directory.

**Startup**

prunePortedNumbers.sh can be started from the command line by using the command:

```
/IN/service_packages/NP_SERVICE_PACK/bin/prunePortedNumbers.sh
```

It can also be started by the cron daemon via an entry in a selected user's (such as smf_oper) crontab.

**Tip:** For details on setting up a crontab entry, see *Adding a prunePortedNumbers.sh crontab entry* (on page 38).

**Adding a prunePortedNumbers.sh crontab entry**

Follow these steps to add prunePortedNumbers.sh to a crontab entry for the smf_oper user, to automate the purging process.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Log on to the SMS as smf_oper.</td>
</tr>
<tr>
<td>2</td>
<td>Type in the command: <code>crontab -e</code></td>
</tr>
</tbody>
</table>
| 3    | Add these lines to the crontab:  
|      | `0 3 * * * (. /IN/service_packages/SMS/.profile ; .  
|      | /IN/service_packages/SMS/.profile-sms ;` |
Chapter 4

Step | Action
--- | ---

```
/IN/service_packages/NP_SERVICE_PACK/bin/prunePortedNumbers.sh )
>>
/IN/service_packages/NP_SERVICE_PACK/tmp/prunePortedNumbers.log
2>&1
```

4

*Save the changes.*

**Result:** The purge process will run every day at 3am.

**Note:** The actual time used can be changed to suit operational requirements by adjusting the values used for the first two fields of the crontab line.

**Failure**

If this script fails to run then the DN range table will not be purged and obsolete records will not be deleted from the database.

**Output**

`prunePortedNumbers.sh` writes error messages to the system messages file.

**Statistics**

**Introduction**

NP collects statistics using the standard SMS statistic mechanism and stores them to the SMF database. Please refer to *SMS Technical Guide* for details of how the statistics are collected.

**Statistics collected**

This table describes the statistics that are collected.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>This statistic is incremented each time destination selection processing is invoked.</td>
</tr>
<tr>
<td>DS2</td>
<td>This statistic is incremented for each internal destination (ported in) request.</td>
</tr>
<tr>
<td>DS3</td>
<td>This statistic is incremented each time the requested destination is not found.</td>
</tr>
<tr>
<td>DS4</td>
<td>This statistic is incremented for each invalid number.</td>
</tr>
<tr>
<td>DS5</td>
<td>This statistic is incremented for each external destination (ported out) request.</td>
</tr>
<tr>
<td>DS6</td>
<td>This statistic is incremented for each internal destination (native operator) request.</td>
</tr>
<tr>
<td>DS7</td>
<td>This statistic is incremented for each external destination (native operator) request.</td>
</tr>
<tr>
<td>LCR1</td>
<td>This statistic is incremented each time a carrier is made available for call routing.</td>
</tr>
<tr>
<td>LCR2</td>
<td>This statistic is incremented each time the LCR macro node is invoked.</td>
</tr>
</tbody>
</table>
Chapter 5

About Installation and Removal

Overview

Introduction

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

In this chapter

This chapter contains the following topics.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation and Removal Overview</td>
<td>41</td>
</tr>
<tr>
<td>NP Table Replication</td>
<td>41</td>
</tr>
<tr>
<td>Checking the Installation</td>
<td>43</td>
</tr>
<tr>
<td>Oracle Configuration</td>
<td>44</td>
</tr>
</tbody>
</table>

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

NP packages

An installation of NP Service Pack includes the following packages, on the:

- SMS:
  - npSms
  - npciSms
  - npPISms
- SLC:
  - npScp

NP Table Replication

Introduction

You must replicate the following NP tables to the SLC to complete the installation of the npScp package:

- NP_CARRIER
- NP_DN_RANGE
- NP_HOME_ROUTING
- NP_PQYZ
Replicating the NP tables

Follow these steps to replicate the NP tables to the SLC.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In the SMS, select <strong>Node Management</strong> from the <strong>Operator Functions</strong> menu.</td>
</tr>
<tr>
<td>2</td>
<td>Select the <strong>Table Replication</strong> tab.&lt;br&gt;<strong>Result:</strong> You see the available and allocated replication groups.</td>
</tr>
<tr>
<td>3</td>
<td>Expand the NP group in the <strong>Available Replication Groups</strong> list.</td>
</tr>
<tr>
<td>4</td>
<td>Click on the table you want to replicate, and drag it to the appropriate SLC node under Service Control Points in the <strong>Allocated Replication Groups</strong> list.&lt;br&gt;<strong>Note:</strong> You can check which nodes are SLC nodes using <strong>Find</strong> and <strong>Search</strong> on the <strong>Replication Nodes</strong> tab.</td>
</tr>
<tr>
<td>5</td>
<td>Repeat step 4 for all the tables you want to replicate.</td>
</tr>
<tr>
<td>6</td>
<td>Click <strong>Save</strong>.</td>
</tr>
<tr>
<td>7</td>
<td>Click <strong>Create Config File</strong>.</td>
</tr>
<tr>
<td>8</td>
<td>Check the database on the SLC to ensure the data has been replicated.</td>
</tr>
</tbody>
</table>
Checking the Installation

Introduction

Refer to these check lists to ensure the NP Service Pack has been installed correctly.

NP database tables - SMS

The following tables should exist on the NP SMF database on the SMS:

- NP_CARRIER
- NP_DN_RANGE
- NP_HOME_ROUTING
- NP_PQYZ
- NP_ROUTING_DESTINATION
- NP_ROUTING_NUMBER
- NP_RULE
- NP_RULE_SET
- SMF_APPLICATION

NP database tables - SCP

The following tables should exist on the SCP database on the SLC:

- NP_CARRIER
- NP_DN_RANGE
- NP_HOME_ROUTING
- NP_PQYZ
- NP_ROUTING_DESTINATION
- NP_ROUTING_NUMBER
- NP_RULE
- NP_RULE_SET
- SMF_APPLICATION

NP directories and files

The NP installation creates the following directories:

/IN/service_packages/NP_SERVICE_PACK/bin
/IN/service_packages/NP_SERVICE_PACK/db/common
/IN/service_packages/NP_SERVICE_PACK/db/MacroNodes
/IN/service_packages/NP_SERVICE_PACK/db/LCR
/IN/service_packages/NP_SERVICE_PACK/db/HR
/IN/service_packages/NP_SERVICE_PACK/db/DS
/IN/service_packages/NP_SERVICE_PACK/etc
/IN/service_packages/NP_SERVICE_PACK/lib
/IN/service_packages/NP_SERVICE_PACK/tmp

The NP installation installs the following binaries and interfaces:

/IN/services_packages/NP_SERVICE_PACK/bin/cdrIF
/IN/services_packages/NP_SERVICE_PACK/bin/npMfileCarrierDaemon
/IN/services_packages/NP_SERVICE_PACK/bin/npMfilePQYZDaemon
/IN/services_packages/NP_SERVICE_PACK/bin/npMfileRoutingDestinationDaemon
/IN/services_packages/NP_SERVICE_PACK/bin/npMfileRoutingDestinationDaemonDest.sh
/IN/services_packages/NP_SERVICE_PACK/bin/npMfileRoutingDestinationDaemonIndex.sh
/IN/services_packages/NP_SERVICE_PACK/bin/npMfileRuleDaemon
/IN/services_packages/NP_SERVICE_PACK/bin/npMfileRuleDaemon.sh
/IN/services_packages/NP_SERVICE_PACK/bin/prunePortedNumbers.sh

The NP installation installs the following example configuration files:
/IN/services_packages/NP_SERVICE_PACK/etc/SLEE.cfg
/IN/services_packages/NP_SERVICE_PACK/etc/acs.cfg
/IN/services_packages/NP_SERVICE_PACK/etc/cdrIF.cfg
/IN/services_packages/NP_SERVICE_PACK/etc/slee_acs_NP.cfg
/IN/services_packages/NP_SERVICE_PACK/etc/mta.cfg

The NP installation installs the following shared libraries:
/IN/services_packages/NP_SERVICE_PACK/lib/libNpCpuChassisActions.so
/IN/services_packages/NP_SERVICE_PACK/lib/libNpSpecificMacroNodeLoader.so

**Oracle Configuration**

**Procedure initSCP.ora**

Follow these steps to configure the initSCP.ora file.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | Type on the command line:  
|      | cd $ORACLE_HOME/dbs |
| 2    | Type vi initSCP.ora to start an editing session. |
| 3    | Add or edit the following lines:  
|      | CURSOR_SHARING = EXACT  
|      | QUERY_REWRITE_ENABLED = TRUE |
| 4    | Save initSCP.ora and exit. |
| 5    | Restart the Oracle database for these settings to take effect. |
Overview

In this appendix

This appendix contains the following topics.

Index

55
NCC Glossary of Terms

ACS
Advanced Control Services configuration platform.

ANI
Automatic Number Identification - Term used in the USA by long-distance carriers for CLI.

ATI
Any Time Interrogation - this process is used on a GSM network to interrogate the HLR for location and or subscriber information.

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

CCS
1) Charging Control Services (or Prepaid Charging) component.
2) Common Channel Signalling. A signalling system used in telephone networks that separates signalling information from user data.

CDR
Call Data Record

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

CID
Call Instance Data

CLI
Calling Line Identification - the telephone number of the caller. Also referred to as ANI.

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

CPE
Control Plan Editor (previously Call Plan Editor) - software used to define the logic and data associated with a call -for example, "if the subscriber calls 0800 nnnnnn from a phone at location xxx then put the call through to bb bbb bbbb".

cron
Unix utility for scheduling tasks.
crontab
File used by cron.

DP
Detection Point

DRA
Destination Routing Address. The parameter in the INAP Connect operation, sent from ACS to the SSP. This is the number the SSP is instructed to connect to.

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.

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

GPRS
General Packet Radio Service - employed to connect mobile cellular users to PDN (Public Data Network - for example the Internet).

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.

GT
Global Title.
The GT may be defined in any of the following formats:
- Type 1: String in the form "1,<noa>,<BCD address digits>"
- Type 2: String in the form "2,<trans type><BCD address digits>"
- Type 3: String in the form "3,<trans type>,<num plan>,<BCD address digits>"
- Type 4: String in the form "4,<trans type>,<num plan>,<noa>,<BCD address digits>"
The contents of the Global Title are defined in the Q713 specification, please refer to section 3.4.2.3 for further details on defining Global Title.

GUI
Graphical User Interface
**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.

**HPLMN**

Home PLMN

**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.

**IDP**

INAP message: Initial DP (Initial Detection Point)

**IMSI**

International Mobile Subscriber Identifier. A unique identifier allocated to each mobile subscriber in a GSM and UMTS network. It consists of a MCC (Mobile Country Code), a MNC (Mobile Network Code) and a MSIN (Mobile Station Identification Number).

The IMSI is returned by the HLR query (SRI-SM) when doing FDA. This tells the MSC exactly who the subscriber is that the message is to be sent to.

**IN**

Intelligent Network

**INAP**

Intelligent Network Application Part - a protocol offering real time communication between IN elements.

**Initial DP**

Initial Detection Point - INAP Operation. This is the operation that is sent when the switch reaches a trigger detection point.

**IP**

1) Internet Protocol

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

**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

**LAC**

Location Area Code. This is an integer value specified as the third level of detail in the location area information. One LAC contains multiple Cell IDs or SAIs.

**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.

**MCC**

Mobile Country Code. In the location information context, this is padded to three digits with leading zeros. Refer to ITU E.212 ("Land Mobile Numbering Plan") documentation for a list of codes.

**MDN**

Mobile Directory Number

**MIN**

Mobile Identification Number, also known as a MSID.

**MNC**

Mobile Network Code. The part of an international address following the mobile country code (MCC), or at the start of a national format address. This specifies the mobile network code, that is, the operator owning the address. In the location information context, this is padded to two digits with a leading zero. Refer to ITU E.212 ("Land Mobile Numbering Plan") documentation for a list of codes.

**MNP**

Mobile Number Portability

**MS**

Mobile Station

**MSC**

Mobile Switching Centre. Also known as a switch.
**MSID**
Mobile Subscriber Identification, also known as an MIN.

**MSIN**
Mobile Station Identification Number.

**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).

**NOA**
Nature Of Address - a classification to determine in what realm (Local, National or International) a given phone number resides, for the purposes of routing and billing.

**NP**
Number Portability

**Oracle**
Oracle Corporation

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

**PI**
Provisioning Interface - used for bulk database updates/configuration instead of GUI based configuration.

**PLMN**
Public Land Mobile Network

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

**SCP**
Service Control Point. Also known as SLC.
SGML

SGSN
Serving GPRS Support Node

SK
Service Key

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

SN
Service Number

SQL
Structured Query Language - a database query language.

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).

SRI
Send Routing Information - This process is used on a GSM network to interrogate the HLR for subscriber routing information.
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.

SSP

Service Switching Point

SUA

Signalling Connection Control Part User Adaptation Layer

Switching Point

Anything that can send and receive C7 messages.

TCAP

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

URI

Uniform Resource Identifier.

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.
# Index

## A
- abort_code_mapping • 18
- About Installation and Removal • 41
- About This Document • v
- ACS • 47
- ACS tags • 34
- acs.conf Configuration • 3, 4
- acsChassis configuration • 4
- Adding a prunePortedNumbers.sh crontab entry • 38
- AddStopDigit • 5
- ANSI • 47
- Appendix • 45
- appID • 6
- ATI • 47
- Audience • v

## B
- Background Processes • 15

## C
- CALLINGNOA (noa of callingnum) • 36
- CALLINGNUM (lcr set calling number) • 36
- CARRIERNAME (carrier name) • 36
- CARRIERPOS (position of carrier name in hunt list) • 36
- CC • 47
- CCS • 47
- CCSDeployment • 6
- CDR • 47
- CdrFileMaxAge • 12
- CdrFileMaxSize • 12
- cdrIF • 15
- Checking the Installation • 43
- CID • 47
- CLI • 47
- ColumnDef • 12
- ColumnSeparator • 12
- Configuration • 3, 16
- Configuration components • 3
- Configuration file format • 4
- Configuration Overview • 3
- Configuring EDR Collection • 3, 10, 16, 33
- Connection • 47
- Copyright • ii
- CPE • 47
- cron • 47
- crontab • 48

## D
- DefaultDestination • 6
- Defaults section • 10
- digitsdialled_locreq_noa_override • 18
- DNMaximal • 6
- DNMinimal • 6
- DNNoaPrefix • 7
- Document Conventions • vi
- DoPOQYZAfterRanges • 7
- DP • 48
- DRA • 48
- DTMF • 48

## E
- Editing configuration files • 4
- EDR • 48
- EDR collection • 10, 33
- EDR fields • 34
- Example ACS EDRs • 35
- Example configuration file • 5
- Example LCR EDRs • 38

## F
- Failure • 16, 28, 29, 30, 31, 32, 39
- FDA • 48
- Features • 1
- FileDesc • 12
- FileDirectory • 10, 13
- FileFooter • 13
- FileHeader • 10, 13
- FileName • 13
- FileSize • 11, 13
- Format • 10
- FormatCLI • 7

## G
- GenLCRCDR • 7
- GPRS • 48
- GSM • 48
- GT • 48
gttRules • 18
- GUI • 48

## H
- HLR • 49
- hop_count • 19
- HPLMN • 49
- HTML • 49

## I
- Icons • vi
- IDP • 49
- IMSI • 49
- IN • 49
- in_timeout • 19
- INAP • 49
- Initial DP • 49
- Installation and Removal Overview • 41
- InternalDestination • 8
- Introduction • 1, 3, 4, 5, 9, 10, 33, 39, 41, 43
IP • 49
ISDN • 49
ISUP • 50
ISUPTrunkInCDR • 8
ITU • 50

L
LAC • 50
LCR EDR tags • 35
Loading configuration changes • 4
local_gt_digits • 20
local_gt_noa • 20
local_ni • 20
local_numplan • 20
local_pc • 20
local_ri • 21
local_ssn • 21
local_transtype • 21
Location • 15, 16, 29, 30, 31, 32, 38

M
MAP • 50
map_response_imsi • 21
map_version • 22
map1_sri_sm_dra_location • 22
MC • 50
MCC • 50
MDN • 50
mdn_smsreq_noa_override • 22
MIN • 50
MNC • 50
MNP • 50
MS • 50
MSC • 50
MSID • 51
MSIN • 51
MSISDN • 51
msisdn_ati_noa_override • 23
msisdn_ati_plan_override • 23
msisdn_sri_noa_override • 23
msisdn_sri_plan_override • 23
msisdn_srism_noa_override • 24
msisdn_srism_plan_override • 24
MT • 51
mta • 3, 16
mta.cfg configuration file • 16
MTP • 51

N
NOA • 51
normalisationRules • 24
NP • 51
NP database tables - SCP • 43
NP database tables - SMS • 43
NP directories and files • 43
NP EDR files • 33
NP EDRs • 33
NP macro nodes • 1
NP packages • 41
NP PI commands • 2
NP SLEE configuration • 9
NP Table Replication • 41
np_components.cfg configuration • 5
np_components.cfg Configuration • 3, 5
npMfileCarrierDaemon • 28
npMfilePQYZDaemon • 29
npMfileRoutingDestinationDaemon • 30
npMfileRuleDaemon • 31

O
OPInRanges • 8
Oracle • 51
Oracle Configuration • 44
ORIGTRUNK (idp location number content) • 36
Output • 28, 29, 30, 31, 32, 39
Overview • 1, 3, 15, 33, 41

P
Parameters • 5, 10, 18, 29, 30, 31, 32
PC • 51
pc_format • 25
PI • 51
PID (unix process ID) • 37
PLMN • 51
PQYZNumType • 8
prefer_imsi • 25
Prerequisites • v
Procedure initSCP.ora • 44
prunePortedNumbers.sh • 38
PTI (product type ID) • 37
Purpose • 15, 16, 28, 29, 30, 31, 38

R
RecordDef • 14
RecordDef section • 11
Related documents • v
RemoveNullColumns • 14
Replicating the NP tables • 42
ROUTEDEST (routing destination for call) • 37
RowHeader • 14
RowTrailer • 14

S
SCCP • 51
sccp_loop_compare_digits_only • 26
sccp_loop_compare_pc_only • 26
Scope • v
SCP • 51
SendCarrierCode • 9
SGML • 52
SGSN • 52
Shutdown • 29, 30, 32
SK • 52
SLC • 52
SLEE • 52
SLEE.cfg Configuration • 3, 9, 15, 16
SME • 52
SMS • 52
SN • 52
SQL • 52
SRF • 52
SRI • 52
sri_sm_dra_location • 26
SS7 • 53
SSP • 53
Startup • 15, 16, 28, 29, 30, 31, 38
Statistics • 39
Statistics collected • 39
SUA • 53
Switching Point • 53
System Overview • 1

T

TCAP • 53
TempDirectory • 11
TIME (creation timestamp of lcr edr) • 37
TNNOA (noa of terminating number) • 37
TNNUM (lcr terminating number) • 37
Tools and Utilities • 33
triggerRules • 27
triggers • 28
Typographical Conventions • vi

U

URI • 53
UseCutAndPaste • 9

V

VLR • 53

W

What is the NP Service Pack • 1