Oracle® Communications
Network Charging and Control
EDR Control Agent Technical Guide
Release 5.0.1

June 2013
## Contents

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

### Chapter 1

**System Overview** ............................................................................................................. 1
  - Overview ........................................................................................................................... 1
  - Introduction to EDR Control Agent .................................................................................. 1
  - Statistics and Reports ....................................................................................................... 4

### Chapter 2

**Configuration** .................................................................................................................. 7
  - Overview ........................................................................................................................... 7
  - Configuration Overview .................................................................................................. 7
  - eserv.config Configuration ............................................................................................. 7

### Chapter 3

**Background Processes** .................................................................................................... 11
  - Overview .......................................................................................................................... 11
  - edrControlAgent ............................................................................................................. 11

### Chapter 4

**Troubleshooting** ............................................................................................................. 23
  - Overview .......................................................................................................................... 23
  - Possible Problems ............................................................................................................. 23

### Chapter 5

**About Installation and Removal** .................................................................................... 25
  - Overview .......................................................................................................................... 25
  - Installation and Removal Overview ................................................................................ 25
  - Checking the Installation ............................................................................................... 25
  - Post-installation Configuration ....................................................................................... 26

**NCC Glossary of Terms** .................................................................................................. 27

**Index** ................................................................................................................................. 31
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 EDR Control Agent.

Audience

This guide was written primarily for system administrators and persons installing, configuring and administering the EDR Control Agent 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 pre-requisite 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 pre-requisite 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:

- Oracle Communications Network Charging and Control SLEE Technical Guide
- Oracle Communications Network Charging and Control ACS Technical Guide
- Oracle Communications Network Charging and Control SMS User's Guide
- Oracle Communications Network Charging and Control SMS 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>
</table>
| **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. |
| **menu option > menu option >** | Used to indicate the cascading menu option to be selected, or the location path of a file.  
                        | Example: **Operator Functions > Report Functions**  
                        | Example: `/IN/html/SMS/Helptext/` |
| **hypertext link**    | Used to indicate a hypertext link on an HTML page. |

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

Introduction to EDR Control Agent

Statistics and Reports

Introduction to EDR Control Agent

Purpose

The Oracle Communications Network Charging and Control EDR Control Agent (ECA) is a SLEE interface which takes EDRs and translates them into InitialDPs. These IDPs can be used to trigger control plans on an SLC. This enables a user to use additional functionality provided by call processing (such as control plans) to process information collected in EDRs from switches or other sources.

Example solution

One possible use of the ECA is to check if a subscriber is using his or her SIM card with its original handset. This can be achieved by sending the IMEI to an ACS control plan.

ECA will read the IMEI from the EDR and send it to slee_acs in an InitialDP in the tbcdExtension string field. The control plan then uses a Profile Field Comparison node to compare the relevant extension call context value with a profile field holding the subscriber’s original IMEI. If it matches, a discount can be applied, or a bonus can be applied to their account.
**Component diagram**

This diagram shows the ECA components.

![Component Diagram](image_url)

**Component descriptions**

This table describes the main components involved in the ECA application.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Further Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>edrControlAgent</td>
<td>Main ECA executable, responsible for all EDR processing. Opens and parses EDRs in the input directory and moves EDRs into success or failed directories depending on plug-in processing. Uses plug-in libraries to process specific EDR types.</td>
<td>edrControlAgent (on page 11).</td>
</tr>
<tr>
<td>edrSimplePlugin.so</td>
<td>Basic plug-in for edrControlAgent.</td>
<td>edrSimplePlugin.so (on page 15).</td>
</tr>
<tr>
<td>SLEE</td>
<td>The SLEE routes calls between applications and interfaces.</td>
<td>SLEE Technical Guide.</td>
</tr>
<tr>
<td>SLEE.cfg</td>
<td>Main configuration file for SLEE. Provides some startup configuration for ECA, including number of instances.</td>
<td>SLEE Technical Guide.</td>
</tr>
<tr>
<td>IN Applications</td>
<td>This could be any call processing component, but is usually slee_acs.</td>
<td>ACS Technical Guide.</td>
</tr>
<tr>
<td>Switch</td>
<td>One of many possible sources of EDRs.</td>
<td></td>
</tr>
<tr>
<td>eserv.config</td>
<td>Main configuration file for ECA.</td>
<td>eserv.config Configuration (on page 7).</td>
</tr>
</tbody>
</table>
EDR processing diagram

This diagram shows ECA processing.

EDR processing

This table describes the stages involved in processing EDRs.

Note: This process uses slee_acs, however other compatible call processing software could be used.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EDR files arrive in the input directory. Example: EDRs may be scp'd from a switch to the SLC.</td>
</tr>
<tr>
<td>2</td>
<td>edrControlAgent moves the EDRs from the input directory to the processing directory.</td>
</tr>
<tr>
<td>3</td>
<td>edrControlAgent passes the EDR file to its plug-in. The plug-in parses EDRs from the EDR file.</td>
</tr>
<tr>
<td>4</td>
<td>For each EDR in the EDR file, the edrControlAgent plug-in generates a TCAP BEGIN containing an InitialDP and call duration which the edrControlAgent sends to slee_acs across the SLEE. The plug-in populates the InitialDP using specific configuration (such as idpParameters (on page 16)). For more information about SLEE configuration, see NCC Service Logic Execution Environment Technical Guide. For more information about slee_acs, see NCC Advanced Control Services Technical Guide.</td>
</tr>
<tr>
<td>5</td>
<td>slee_acs runs the relevant control plan. This may involve further TCAP operations being passed between slee_acs and edrControlAgent.</td>
</tr>
<tr>
<td>6</td>
<td>When the call is over (due to an appropriate TCAP operation or a timeout), the edrControlAgent's plug-in closes the file. If the EDRs were successful, it moves the file to the success directory. Any EDRs which could not be parsed are written to a separate file in the failed directory.</td>
</tr>
</tbody>
</table>
Supported InitialDP operations

edrControlAgent uses CAP 3 INAP in messages to slee_acs. These INAP operations are supported:

- InitialDP
- ApplyChargingReport
- EventReportBCSM
- RequestReportBCSMEvent
- ApplyCharging
- Connect
- Continue
- ReleaseCall

Unsupported InitialDP parameters

All other INAP operations are not supported, including:

- ConnectToResource
- EstablishTemporaryConnection
- PlayAnnouncement
- PromptAndCollectUserInformation
- DisconnectLeg
- CallInformationRequest

Warning: You must set up the control plans and the `acs.conf` and `eserv.config` files so that these and other unsupported operations are not received by edrControlAgent in EDRs.

Statistics and Reports

Introduction

edrControlAgent will log statistics to the Service Management System statistics subsystem if it is configured to do so in `eserv.config`. edrControlAgent will use Application ID ECA.

For more information about the SMS statistics subsystem, see *NCC Service Management System Technical Guide* and *NCC Service Management System User's Guide*.

Statistics

This table describes the statistics which can be logged by edrControlAgent.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INITIALDPS_SENT</td>
<td>Number of InitialDP operations sent.</td>
</tr>
<tr>
<td>CALL_PROCESSING_FAILED</td>
<td>Number of InitialDPs for which processing failed.</td>
</tr>
<tr>
<td>APPLY_CHARGING_CONNECT_SUCCESSFUL</td>
<td>Number of calls where AC-Connect received.</td>
</tr>
<tr>
<td>APPLY_CHARGING_CONTINUE_SUCCESSFUL</td>
<td>Number of calls where AC-Continue received.</td>
</tr>
<tr>
<td>APPLY_CHARGING_CONNECT_INSUFFICIENT_FUNDS</td>
<td>Number of calls where (AC-Connect) duration is greater than granted time.</td>
</tr>
<tr>
<td>APPLY_CHARGING_CONTINUE_INSUFFICIENT_FUNDS</td>
<td>Number of calls where (AC-Continue) duration is greater than granted time.</td>
</tr>
</tbody>
</table>
Chapter 1

System Overview

RELEAS|E CALL
Number of calls where a (ReleaseCall) was received. Detail = cause.

CONNECT
Number of calls where Connect received.

CONTINUE
Number of calls where Continue received.

TIMED_OUT
Number of calls for which the tssf timer expired. For more information about the tssf timer, see `tssf` (on page 15).

Reports

ECA does not install any specific reports. However, you can report on any statistics which are recorded using the SMS Application report on the Service Management System Report Functions screen.

For more information about running SMS reports, see *NCC Service Management System User’s Guide*.

Report example

This text shows an example of the SMS Application report run for ECA for the previous 24 hours showing Total counts for each statistic.

```
Application Statistics Listing
-----------------------------
Hours since: 24
Application: ECA
Report Type: Totals

08 January 2008, 02:27:35

Node Name  Statistics ID                        Totals
----------- ------------------------------------- -------
prodscp1    CONNECT                               0
prodscp1    CONTINUE                              106
prodscp1    TIMED_OUT                             1200
prodscp1    RELEASE_CALL                          1209
prodscp1    INITIALDFS_SENT                       6102
prodscp1    CALL_PROCESSING_FAILED                91
prodscp1    APPLY_CHARGING_CONNECT_SUCCESSFUL     3192
prodscp1    APPLY_CHARGING_CONTINUE_SUCCESSFUL    0
prodscp1    APPLY_CHARGING_CONNECT_INSUFFICIENT_FUNDS 79
prodscp1    APPLY_CHARGING_CONTINUE_INSUFFICIENT_FUNDS 0

10 rows selected.
```

Completed
Chapter 2

Configuration

Overview

Introduction

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

In this chapter

This chapter contains the following topics.

- Configuration Overview
- eserv.config Configuration

Configuration Overview

Introduction

This topic provides a high level overview of how Oracle Communications Network Charging and Control EDR Control Agent is configured.

Configuration components

ECA 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>eserv.config</td>
<td>SLCs</td>
<td>The most important configuration file. It configures most NCC applications,</td>
<td>eserv.config Configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>including ECA processes. ECA is configured by the ECA section of eserv.config.</td>
<td>(on page 7).</td>
</tr>
<tr>
<td>SLEE.cfg</td>
<td>SLCs</td>
<td>Sets details about how the ECA runs including number of instances.</td>
<td>SLEE Technical Guide</td>
</tr>
</tbody>
</table>

eserv.config Configuration

Introduction

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

The eserv.config file is located in the /IN/service_packages/ directory.
The `eserv.config` file format uses hierarchical groupings, and most applications make use of this to divide up the options into logical groupings.

### Configuration file format

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

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

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

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

or

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

or

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

### ECA eserv.config example section

ECA adds a cut-down ECA section to `eserv.config` file. It is not a full list of all parameters that are available.

The ECA section has initial values which may need to be amended to suit a specific installation. Once amended, ECA will run with no further changes to Oracle. Where additional implementation changes need to be made to Oracle, refer to the *Background Processes* (on page 11) chapters for full descriptions of all parameters for the processes.

### Editing the file

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

Always keep a backup of your file before making any changes to it. This will ensure you have a working copy to which you can return.
Loading eserv.config changes

If you change the configuration file, then you must restart the appropriate parts of the service to enable the new options to take effect.
Chapter 3

Background Processes

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.
edrControlAgent

edrControlAgent

Purpose

edrControlAgent processes InitialDPs and sends them to an application across the SLEE. The edrControlAgent uses separate plugins to process different types of EDRs. It can only run one plugin per instance.

For more information about the SLEE, see NCC Service Logic Execution Environment Technical Guide.

Startup

This task is started by the SLEE, by the following line in the SLEE.cfg configuration file:

```
INTERFACE=edrControlAgent eca.sh /IN/service_packages/ECA/bin 1 EVENT
```

Note: The above are defaults and may vary.

Configuration

In order to load and operate, the edrControlAgent reads the ECA section of the eserv.config file. The high level structure of the ECA section is shown below.

```
ECA = [
  {
    sleeInterfaceName = "uniqueID"
    inputDirectory = "dir"
    processingDirectory = "dir"
    badFileDirectory = "dir"
    fileNamePattern = "pattern"
  }
]`
sleeServiceKey = key
maxIdpsPerSecond = seconds
statisticsEnabled = true|false
tssf = seconds
pluginLibrary = "lib"
PluginSpecificConfig = {
inapServiceKey = key
    commentChar = "str"
    separatorChar = "str"
    idpParameters = ["str"
        ...
    ]
    NumberRules = [
        { [remove=int][, replace=str] }
        ...
    ]
    successDirectory = "dir"
    failedDirectory = "dir"
}
{additional_ECA_instances}
...
}

**Parameters**

Parameters of the edrControlAgent are listed below.

**badFileDirectory**

**Syntax:**

badFileDirectory = "dir"

**Description:**
The edrControlAgent moves EDR files that it cannot open for processing to this directory.

**Type:**
String

**Optionality:**
Required

**Default:**
/IN/service_packages/ECA/failed

**Example:**
badFileDirectory = "/IN/service_packages/ECA/failed"

**fileNamePattern**

**Syntax:**

fileNamePattern = "pattern"

**Description:**
The pattern that the edrControlAgent uses to identify EDR files in the input directory.

**Type:**
String

**Optionality:**
Required

**Default:**
fileNamePattern = "*.edr"

**Notes:**
This parameter can be used to:

- Ensure only EDR files are processed (while other files such as README files are left alone)
- Enable a specific edrControlAgent instance to select specific EDR files from a common input directory

**Example:**
fileNamePattern = "*.edr"
inputDirectory

Syntax: \texttt{inputDirectory = "dir"}

Description: The full path of the directory which holds EDRs to be processed by edrControlAgent.

Type: String

Optionality: Mandatory

Default: /IN/service_packages/ECA/input

Example: inputDirectory = "/IN/service_packages/ECA/input"

maxIdpsPerSecond

Syntax: \texttt{maxIdpsPerSecond = num}

Description: The maximum number of IDPs edrControlAgent will send across the SLEE within one second.

Type: Integer

Optionality: Optional, default used if not set

Allowed: Positive integers.
If set to 0 (zero), no throttling will be performed.

Default: 0

Example: maxIdpsPerSecond = 250

pluginLibrary

Syntax: \texttt{pluginLibrary = "lib"}

Description: The ECA plug-in which processes EDRs of a specific type.

Type: String

Optionality: Optional, default used if not set

Default: ECASimplePlugin.so

Notes: You can only have one plug-in per instance. To run two plug-ins, you must have two edrControlAgents configured in the ECA section of eserv.config.

Example: pluginLibrary = "ECASimplePlugin.so"

PluginSpecificConfig

Syntax: \texttt{PluginSpecificConfig = \{ config \}}

Description: Configuration for the library specified in pluginLibrary (on page 13).

Type: Array

Optionality: Required

Notes: The detail of this array will depend on the plug-in.

Example: PluginSpecificConfig = {
  inapServiceKey = 900
  commentChar = "#"
  separatorChar = "|"
  idpParameters = [
    "calledPartyNumber"
    "skip"
    "callingPartyNumber"
    "locationNumber"
    "tbcdExtension 1234"
    "asn1Extension 1234"
    "callDurationDeciseconds"}
NumberRules = [
    { prefix = "6449", min = 5, remove = 2, prepend = "0",
      resultNoa = 2 }
]  
successDirectory = "/IN/service_packages/ECA/success"
failedDirectory = "/IN/service_packages/ECA/failed"

processingDirectory
Syntax: processingDirectory = "dir"
Description: Before processing the EDR files, the edrControlAgent moves the files to the specified directory.
Type: String
Optionality: Required
Default: /IN/service_packages/ECA/processing
Notes: You can use this parameter to enable different instances of edrControlAgent to run on the same machine, while using the same input directory.
Example: processingDirectory = "/IN/service_packages/ECA/processing"

sleeInterfaceName
Syntax: SleeInterfaceName = "name"
Description: The unique identifier of this instance of the edrControlAgent interface in SLEE.cfg.
Type: Integer
Optionality: Required
Default: edrControlAgent
Notes: Must match the unique identifier of edrControlAgent in the SLEE.cfg file.
For more information about SLEE.cfg, see NCC Service Logic Execution Environment Technical Guide.
Example: SleeInterfaceName = "edrControlAgent"

sleeServiceKey
Syntax: SleeServiceKey = skey
Description: The service key to use in outgoing InitialDPs.
Type: Integer
Optionality: Required
Default: 900
Notes: Must match the SERVICEKEY entry for the edrControlAgent (slee_acs on install) in SLEE.cfg. For more information about SLEE.cfg, see NCC Service Logic Execution Environment Technical Guide.
Set to 900 at installation.
Example: SleeServiceKey = 900

statisticsEnabled
Syntax: statisticsEnabled = true|false
Description: Defines whether edrControlAgent will log statistics about its processing to the SMS statistics sub-system.
Type: Integer
Optionality: Optional, default used if not set.
Allowed: true, false
Default: true
Notes:
Example: statisticsEnabled = false

tssf

Syntax: \texttt{tssf = \text{seconds}}

Description: The number of seconds edrControlAgent will wait for a response before abandoning the IDP before closing the dialog.

Type: Integer

Optionality: Optional, default used if not set.

Allowed: 

Default: 30

Notes: If the plug-in requires it, edrControlAgent will increment a statistic count when a message times out.

Example: tssf = 10

\texttt{edrSimplePlugin.so}

The \texttt{PluginSpecificConfig} section of the ECA \texttt{eserv.config} configuration supports these parameters for \texttt{edrSimplePlugin.so}.

\begin{verbatim}
PluginSpecificConfig = {
    inapServiceKey = key
    commentChar = "str"
    separatorChar = "str"
    idpParameters = [
        "str"
        ...
    ]
    NumberRules = [
        { [prefix="str", ][min=int, ][max=int, ] remove=int[, replace=str][, prepend="str"][, resultNoa=int] } 
        ...
    ]
    successDirectory = "dir"
    failedDirectory = "dir"
}
\end{verbatim}

The parameters are described in detail below.

\textbf{commentChar}

Syntax: \texttt{commentChar = "char"

Description: The character which will signify that the characters after this character are comments, and should not be processed.

Type: String

Optionality: Optional, default will be used if not set.

Allowed: ASCII characters

Default: "#"

Notes:

Example: commentChar = "#"
failedDirectory
Syntax: failedDirectory = "dir"
Description: The directory edrControlAgent writes EDRs to when it has encountered an error while parsing them.
Type: String
Optionality: Mandatory
Allowed: 
Default: None
Notes: Successful EDRs from the file are written to a separate file in the directory specified by successDirectory (on page 17).
Example: failedDirectory = "/var/edr/failed"

idpParameters
Syntax: idpParameters = ["parameters"]
Description: The EDR fields in the order they appear in the file.
Type: Array of strings
Optionality: Mandatory
Allowed: For information about the fields you can use, see idpParameter details (on page 16). If fields other than those specified are used, edrControlAgent will not startup.
Default: None
Notes: The separator in the EDR file is specified in separatorChar (on page 17).
Example: idpParameters = ["calledPartyNumber" "skip" "callingPartyNumber" "locationNumber" "tbcdExtension 1234" "callDurationSeconds"]
This configuration will correctly parse the following EDR:
123456789|skip|987654321|123456|490154203237518|123

idpParameter details
You can specify the following strings for idpParameters:

<table>
<thead>
<tr>
<th>String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>skip</td>
<td>The field in this position is irrelevant. Skip it.</td>
</tr>
<tr>
<td>callDurationSeconds</td>
<td>Call duration in seconds</td>
</tr>
<tr>
<td>callDurationDeciseconds</td>
<td>Call duration in deciseconds</td>
</tr>
<tr>
<td>tbcdExtension(&lt;int extension type&gt;)</td>
<td>An IDP extension with the given type, as an octet string. The octet string contains a coded up MAP TBCD-STRING.</td>
</tr>
<tr>
<td>calledPartyNumber</td>
<td>IDP.calledPartyNumber</td>
</tr>
<tr>
<td>callingPartyNumber</td>
<td>IDP.callingPartyNumber</td>
</tr>
<tr>
<td>callingPartysCategory</td>
<td>IDP.callingPartysCategory</td>
</tr>
<tr>
<td>locationNumber</td>
<td>IDP.locationNumber</td>
</tr>
<tr>
<td>additionalCallingPartyNumber</td>
<td>IDP.additionalCallingPartyNumber</td>
</tr>
<tr>
<td>redirectingPartyID</td>
<td>IDP.redirectingPartyID</td>
</tr>
<tr>
<td>iMSI</td>
<td>IDP.iMSI</td>
</tr>
</tbody>
</table>
### inapServiceKey

**Syntax:**

inapServiceKey = *skey*

**Description:**
The service key for all IDPs generated by this plug-in.

**Type:**
Integer

**Optionality:**
Mandatory

**Allowed:**
Valid IDPs as specified in the relevant standard.

**Default:**
None

**Notes:**
If you want to run some EDRs on a different service key, you will need to run more than one instance of edrControlAgent, and split the EDRs for each process.

**Example:**
inapServiceKey = 111

### separatorChar

**Syntax:**

separatorChar = "*char*

**Description:**
The character which separates the fields in the EDR.

**Type:**
String

**Optionality:**
Optional, default will be used if not set.

**Allowed:**
ASCII

**Default:**
|

**Notes:**

**Example:**
separatorChar = "|"

### successDirectory

**Syntax:**

successDirectory = "*dir*

**Description:**
The directory edrControlAgent should move successfully processed EDR files to.

**Type:**
String

**Optionality:**
Mandatory

**Allowed:**

**Default:**
None

**Notes:**

**Example:**
successDirectory = "/var/edr/success"

---

**NumberRules**

The **NumberRules** subsection of the ECA **eserv.config** configuration supports these parameters. If the **NumberRules** section is not present, all numbers will be assumed to be in international format. This section applies to denormalization only.

```java
NumberRules = [  
    { [prefix="str", ][min=int, ][max=int, ] remove=int[, replace=str][, prepend="str"][, resultNoa=int] }  
    ...  
]
```

The parameters are described in detail below.
max
Syntax: \texttt{\texttt{max = maxNoLength}}
Description: The maximum number of digits a number may contain. If the number contains digits less than or equal to this value, the max part of the number rule is met.
Type: Integer
Optionality: Optional, (if not set, default it used).
Allowed: 
Default: 999
Notes: This parameter is an element of the \texttt{NumberRules} parameter array.
Example: max = 9

min
Syntax: \texttt{\texttt{min = minNoLength}}
Description: The minimum number of digits a number may contain. If the number contains digits is greater than or equal to this value, the min part of the number rule is met.
Type: Integer
Optionality: Optional, if not set default is used.
Allowed: \texttt{0 \leq \texttt{minNoLength}}
Default: 0
Notes: 
\begin{itemize}
  \item The remove parameter affects the min parameter.
    If remove is equal to \texttt{noOfRemovedDigits} and \texttt{noOfRemovedDigits} is greater than 0, then minNoLength must be set so minNoLength is greater than or equal to noOfRemovedDigits.
  \item This parameter is an element of the \texttt{NumberRules} parameter array.
\end{itemize}
Example: min = 5

prefix
Syntax: \texttt{\texttt{prefix = \"prefix\"}}
Description: Contains a digit or digits. Rule attempts to match the first digit or digits of a number with this value. If the digit or digits match, the prefix part of the number rule is met.
Type: String
Optionality: Optional
Allowed: One or more decimal digits.
Default: 
Notes: This parameter is an element of the \texttt{NumberRules} parameter array.
Example: prefix = "25"

prepend
Syntax: \texttt{\texttt{prepend = \"firstDigits\"}}
Description: Digits added to the beginning of a number.
Type: String
Optionality: Optional
Allowed: Can be:
\begin{itemize}
  \item Any combination of decimal digits
  \item A null string (""")
\end{itemize}
Default:

Notes:

- If the `remove` and `prepend` parameters are both used in the same number rule, `firstDigits` is added to the beginning of the number after the number has been modified by the `remove` parameter.
- This parameter is an element of the `NumberRules` parameter array.

Example: prepend = "0"

`remove`

Syntax: remove = `noOfRemovedDigits`

Description: The number of digits stripped from the beginning of a number.

Type: Integer

Optionality: Mandatory

Allowed: 

Default: 

Notes:

- The `remove` parameter affects the `min` parameter.
  - If `min` is equal to `minNoLength` and if `noOfRemovedDigits` is greater than 0, then `minNoLength` must be set to `minNoLength` is greater than or equal to `noOfRemovedDigits`.
  - See `min` (on page 18).
- The `remove` parameter is an element of the `NumberRules` parameter array.

Example: remove = 2

`replace`

Syntax: replace = "str"

Description: Characters that the number rule substitutes for a number.

Type: String

Optionality: Optional

Allowed: 

Default: 

Notes: This parameter is an element of the `NumberRules` parameter array.

Example: replace = "111"

`resultNoa`

Syntax: resultNoa = `NOA`

Description: The nature of address (NOA) sent to the network after denormalization.

Type: Integer

Optionality: Optional

Allowed: 

Default: 

Notes:

- A value for `NOA` is typically specified in denormalization rules.
- This parameter is an element of the `NumberRules` parameter array.

Example: resultNoa = 4
Example number denormalisation

Example 1

```
{ prefix="027", min=9, remove=1, resultNoa=3 }
```

This denormalization rule:

- Matches numbers that:
  - Start with the digits 027
  - Have a minimum of 9 digits
- Removes the first digit.
- Sets NOA = 3.

For example, in the outgoing message, 027nnnnnn would change to 27nnnnnn.

Example 2

```
{ prefix="00", min=5, remove=2, prepend="", resultNoa=4 }
```

This denormalization rule:

- Matches numbers that:
  - Start with the digits 00
  - Contain a minimum of 5 digits
- Removes the first two digits.
- Sets NOA = 4.

For example, the outgoing message 00nnnnnnnnnnn would change to nnnnnnnnnnn.

Example configuration

An example of the ECA section of a eserv.config file is listed below. Comments have been removed.

```plaintext
ECA = {
  sleeInterfaceName = "EDRControlAgent"
  inputDirectory = "/var/edr/input"
  processingDirectory = "/var/edr/processing"
  badFileDirectory = "/var/edr/unprocessable"
  fileNamePattern = "*.cdr"
  sleeServiceKey = 300
  maxIdpsPerSecond = 250
  statisticsEnabled = true
  tssf = 10
  pluginLibrary = "ECASimplePlugin.so"
  PluginSpecificConfig = {
    inapServiceKey = 111
    commentChar = "#"
    separatorChar = "\\""
    idpParameters = [
      "calledPartyNumber"
      "skip"
      "callingPartyNumber"
      "locationNumber"
      "tbcdExtension 1234"
      "callDurationSeconds"
    ]
    NumberRules = [
      { prefix = "6449", min = 5, remove = 2, prepend = "0", resultNoa = 2 }
    ]
    successDirectory = "/var/edr/success"
    failedDirectory = "/var/edr/failed"
  }
}
```
Failure

If edrControlAgent stops while processing an EDR file, it will write the following information to ecaProgress.txt:

- The names of all the files being processed
- The name of the EDR file currently being processed
- The number of IDPs sent for the current file
- (if the plug-in requires it) the file names and EDR number for all the 'calls' in progress

When it restarts, it will use ecaProgress.txt to identify where to start again.

If edrControlAgent stops without writing the ecaProgress.txt file, edrControlAgent will use the plug-in to move any files out of the progress directory to success or failed depending on whether they were able to be parsed or not.

Input

The edrControlAgent takes EDRs from the input directory configured by the inputDirectory (on page 13) parameter in eserv.config.

Output

edrControlAgent moves processed EDRs to the success or failed directory depending on the result of the EDR processing. These directory locations are configurable in eserv.config.

edrControlAgent also writes error messages to the system messages file, and also writes additional output to /IN/service_packages/ECA/tmp/edrControlAgent.log

Note: The above are defaults and may vary.
Overview

Introduction

This chapter explains the important processes on each of the server components in the NCC, and a number of example troubleshooting methods which will help aid the troubleshooting process before raising a support ticket.

In this chapter

This chapter contains the following topics.
Possible Problems 23

Possible Problems

Introduction

This topic lists common problems and actions which can be taken to investigate or solve them. This list enables you to check for alarms based on the overall behavior you are experiencing.

Flooding

dCrControlAgent may produce more IDPs than the processing applications can cope with. In this circumstance, dCrControlAgent will wait until the next second to re-send the IDP.

You can throttle the number of IDPs dCrControlAgent will send by specifying one of the following:

- A lower maxIdpsPerSecond (on page 13)
- A lower MAX_DIALOOGS for the dCrControlAgent INTERFACE entry in SLEE.cfg.
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.

Chapter 5, About Installation and Removal 25

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

ECA packages

An installation of Oracle Communications Network Charging and Control EDR Control Agent includes the following packages, on the:

- SMS:
  - ecaSms
- SLC:
  - ecaScp

Checking the Installation

Introduction

Refer to these checklists to ensure that ECA has installed correctly.

Checklist - SMS

Follow these steps in this checklist to ensure ECA has been installed on an SMS machine correctly.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Log in to SMS machine as root.</td>
</tr>
</tbody>
</table>
Step   Action
2   Check the following directory structure exists with subdirectories:   
    /IN/service_packages/ECA
3   Check that directories contain subdirectories and that all are owned by:   
    smf_oper user (group esg)

Checklist - SLC
Follow these steps in this checklist to ensure ACS has been installed on an SLC machine correctly.

Step   Action
1   Log in to SLC machine as root.
2   Check the following directory structure exists with subdirectories:   
    /IN/service_packages/ECA
3   Check the directory contains subdirectories and that all are owned by:   
    smf_oper user (group esg)
4   Check that the processes listed in the process lists are running. For a list of the processes which should be running, see Process list - SLC (on page 26).

Process list - SLC
If the application is running correctly, the following process should be running on each SLC:
- Started during SLEE startup:
  - edrControlAgent

Post-installation Configuration

Configuration process overview
This table describes the steps involved in configuring ECA for the first time.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Uncomment (and if necessary update) the default SLEE configuration for the ECA the SLEE.cfg file. For more information, see NCC Service Logic Execution Environment Technical Guide.</td>
</tr>
<tr>
<td>2</td>
<td>The eserv.config file must be configured for ECA. The installation script will have added an ECA section to eserv.config. Any required configuration should be updated. For more information, see eserv.config Configuration (on page 7).</td>
</tr>
</tbody>
</table>
NCC Glossary of Terms

AC
Application Context. A parameter in a TCAP message which indicates what protocol is conveyed. May indicate, for example, MAP, CAMEL, or INAP. Also usually specifies the particular version of the conveyed protocol, for example, which CAMEL Phase.

ACS
Advanced Control Services configuration platform.

CAMEL
Customized Applications for Mobile network Enhanced Logic
This is a 3GPP (Third Generation Partnership Project) initiative to extend traditional IN services found in fixed networks into mobile networks. The architecture is similar to that of traditional IN, in that the control functions and switching functions are remote. Unlike the fixed IN environment, in mobile networks the subscriber may roam into another PLMN (Public Land Mobile Network), consequently the controlling function must interact with a switching function in a foreign network. CAMEL specifies the agreed information flows that may be passed between these networks.

CAP
CAMEL Application Part

cron
Unix utility for scheduling tasks.

crontab
File used by cron.

DP
Detection Point

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.

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

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.

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.

MS
Mobile Station

MT
Mobile Terminated

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.

Oracle
Oracle Corporation
**PLMN**
Public Land Mobile Network

**SGML**

**SGSN**
Serving GPRS Support Node

**SIM**
Usually referred to as a SIM card, the Subscriber Identity Module is the user subscription to the mobile network. The SIM contains relevant information that enables access onto the subscripted operator's network.

**SLC**
Service Logic Controller (formerly UAS).

**SLEE**
Service Logic Execution Environment

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

**SN**
Service Number

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

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

**VWS**
Oracle Voucher and Wallet Server (formerly UBE).
Index

A
About Installation and Removal • 25
About This Document • v
AC • 27
ACS • 27
Audience • v

B
Background Processes • 8, 11
badFileDirectory • 12

C
CAMEL • 27
CAP • 27
Checking the Installation • 25
Checklist - SLC • 26
Checklist - SMS • 25
commentChar • 15
Component descriptions • 2
Component diagram • 2
Configuration • 7, 11
Configuration components • 7
Configuration file format • 8
Configuration Overview • 7
Configuration process overview • 26
Copyright • ii
cron • 27
crontab • 27

D
Document Conventions • vi
DP • 27

E
ECA eserv.config example section • 8
ECA packages • 25
Editing the file • 8
EDR • 27
EDR processing • 3
EDR processing diagram • 3
edrControlAgent • 2, 11
edrSimplePlugin.so • 2, 15
eserv.config Configuration • 3, 7, 26
Example configuration • 20
Example number denormalisation • 20
Example solution • 1

F
failedDirectory • 16
Failure • 21
fileNamePattern • 12
Flooding • 23

G
GPRS • 27

H
HLR • 28
HPLMN • 28
HTML • 28

I
IDP • 28
idpParameter details • 16
idpParameters • 3, 16
IN • 28
INAP • 28
inapServiceKey • 17
Initial DP • 28
Input • 21
inputDirectory • 13, 21
Installation and Removal Overview • 25
Introduction • 4, 7, 23, 25
Introduction to EDR Control Agent • 1

L
Loading eserv.config changes • 9

M
MAP • 28
max • 18
maxIdpsPerSecond • 13, 23
min • 18, 19
MS • 28
MT • 28

N
NOA • 28
NumberRules • 18

O
Oracle • 28
Output • 21
Overview • 1, 7, 11, 23, 25

P
Parameters • 12
PLMN • 29
pluginLibrary • 13
PluginSpecificConfig • 13
Possible Problems • 23
Post-installation Configuration • 26
prefix • 18
prepend • 19
Prerequisites • v
Process list - SLC • 26
processingDirectory • 14
Purpose • 1, 11