Oracle® Communications
Network Charging and Control
SMS Email Interface 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.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About This Document</td>
<td>v</td>
</tr>
<tr>
<td>Document Conventions</td>
<td>vi</td>
</tr>
<tr>
<td><strong>Chapter 1</strong></td>
<td></td>
</tr>
<tr>
<td><strong>System Overview</strong></td>
<td>1</td>
</tr>
<tr>
<td>Overview</td>
<td>1</td>
</tr>
<tr>
<td>What is SEI?</td>
<td>1</td>
</tr>
<tr>
<td>SMS to Email</td>
<td>1</td>
</tr>
<tr>
<td>Email to SMS</td>
<td>2</td>
</tr>
<tr>
<td><strong>Chapter 2</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td></td>
</tr>
<tr>
<td>Overview</td>
<td>3</td>
</tr>
<tr>
<td>eserv.config Configuration</td>
<td>3</td>
</tr>
<tr>
<td>SEI Configuration</td>
<td>4</td>
</tr>
<tr>
<td><strong>Chapter 3</strong></td>
<td>25</td>
</tr>
<tr>
<td><strong>Background Processes</strong></td>
<td></td>
</tr>
<tr>
<td>Overview</td>
<td>25</td>
</tr>
<tr>
<td>sei</td>
<td>25</td>
</tr>
<tr>
<td><strong>Chapter 4</strong></td>
<td>27</td>
</tr>
<tr>
<td><strong>About Installation and Removal</strong></td>
<td></td>
</tr>
<tr>
<td>Overview</td>
<td>27</td>
</tr>
<tr>
<td>Installation and Removal Overview</td>
<td>27</td>
</tr>
<tr>
<td>Checking the Installation</td>
<td>27</td>
</tr>
<tr>
<td><strong>NCC Glossary of Terms</strong></td>
<td>29</td>
</tr>
<tr>
<td><strong>Index</strong></td>
<td>33</td>
</tr>
</tbody>
</table>
About This Document

Scope

The scope of this document includes all the information required to install, configure and administer the SMS Email Interface application.

Audience

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

Prerequisites

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

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

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

Related documents

The following documents are related to this document:

- MM Technical Guide
- SEI Alarms 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.</td>
</tr>
<tr>
<td></td>
<td>Names of database tables and fields.</td>
</tr>
<tr>
<td><strong>Italics</strong></td>
<td>Name of a document, chapter, topic or other publication.</td>
</tr>
<tr>
<td></td>
<td>Emphasis within text.</td>
</tr>
<tr>
<td><strong>Button</strong></td>
<td>The name of a button to click or a key to press.</td>
</tr>
<tr>
<td></td>
<td><em>Example:</em> To close the window, either click <em>Close</em>, or press <em>Esc</em>.</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.</td>
</tr>
<tr>
<td></td>
<td><em>Example:</em> Ctrl+P, or Alt+F4.</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.</td>
</tr>
<tr>
<td></td>
<td><em>Example:</em> Operator Functions &gt; Report Functions</td>
</tr>
<tr>
<td></td>
<td><em>Example:</em> /IN/html/SMS/HelpText/</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.
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 SEI?  1
SMS to Email  1
Email to SMS  2

What is SEI?

Introduction

The Short Message Service Internet Email Interface (SEI) facilitates sending and receiving Internet email on a mobile telephone handset using Short Message Service (SMS) technology.

SMS to Email

Format of SMS

Subscribers can send a specially formatted SMS to a Direct SMS-to-Email short code to send email from their SMS enabled cell phone.

- This number would typically be labeled “email” in their own phone’s address book.
- The first word of the SMS will be the destination email address. The message body follows, for example:
  sam@gmail.com How about lunch?”
- Optionally the subject may be provided by prefixing it with an ‘s’, the subject continues until the first double space or newline in the SMS, for example:
  sam@gmail.com s Lunch today  How about lunch?
  The Subject is “Lunch today” and the message body is “How about lunch?”.
- The From:Email address will be in the form msisdn@domain. This facilitates replies back to the phone – see Email to SMS (on page 2).
- Subscribers may be charged, in the SMSC, for each email sent through the gateway.
Email to SMS

Format of email

Emails sent to an address that is known to be a mobile subscriber, for example msisdn@domain, will be relayed to the MSISDN as an SMS will be relayed to the MSISDN as an SMS through the Email_to_SMS control plan.

The From:email address and subject will be tightly packed into the SMS as follows: for example:

From: Sam Smith <sam@gmail.com>
To: 027111222 <027111222@oracle.com>
Date: Aug 5, 2005 2:27 PM
Subject: RE: Lunch
How about Rahzoo?.

This email becomes the SMS:

sam@gmail.com  RE: Lunch:  How about Rahzoo?

The From:Email address is looked up in the mobile subscriber's profile. If it is:

- found, the From:address is set to the special short code plus a digit representing the placement in the listing.
  For example, sam@gmail.com is found in subscriber 027111222's Email Address 3. The From:address is set to 703, for example 70 is the special short code and 3 represents Email Address 3. The mobile subscriber would send a reply to 703 which will trigger the Enhanced_SMS_to_Email control plan which will replace the A party number with an email address in the form msisdn@domain - see Enhanced SMS to Email in MM User's Guide.
- not found, the From:address is set to a special short code for unprovisioned email addresses, for example 710. The mobile subscriber would send a reply to 710 which will trigger the SMS_to_Email control plan plan which will replace the A party number with an email address in the form msisdn@domain - see Direct SMS to Email in MM User's Guide.
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.
eserv.config Configuration 3
SEI Configuration 4

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:

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

or

```
{ name="route6"
  id = 3
  prefixes = [ "00000148"
               "0000473"
            ]
}
```
eserv.config files delivered

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

Editing the file

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

SEI Configuration

Introduction

The SEI section in the eserv.config file must be configured to enable the SEI to work. When SEI is installed, a soft link is created in /IN/service_packages/SEI/etc pointing eserv.config to eserv.config.pme. Therefore, any changes to the SEI configuration must be made in the .pme file.

An example eserv.config file showing all the available configuration options is installed by the seiScp package in:

/IN/services_packages/SEI/etc/eserv.config.example

The config file needs to be present on all SLCs.

Note: All mandatory configuration in the config file is done at installation time by the configuration script.

Example SEI configuration section

Here is an example SEI section of the eserv.config file. This example is the standard SMS to Email Interface setup for the MM PME package.

# The EMAIL_DOMAIN environment variable needs to be defined in order to run sei with this config
#

SEI = {
# Short message service internet Email Interface system
# some config values may have $values expanded
# $HOSTNAME the unix hostname
# $ENV the value of an environment variable, for example. $HOME
# $link.to.var another value in this config file, for example $SEI.sei
# exampledomain
# links may refer to other links

database = {
    user = "mmx_admin"
    password = "mmx_admin"
}

eemail = {
    domain = "$EMAIL_DOMAIN"
    errorMailbox = "error"
    numberRules = [
    ]
    postmasterAction = {
        action = "ignore"
    }
    errorNotificationAction = {
        action = "ignore"
    }
    deliveryStatusNotification = {
        directory = "/IN/service_packages/SEI/tmp/dsn"
        failDirectory = "/IN/service_packages/SEI/tmp/fail"
        retries = ["2 hours", "3 minutes"]
    }
}

client = {
    connectTimeout = 300
    initialMessageTimeout = 300
    helloTimeout = 300
    mailTimeout = 300
    recipientTimeout = 300
    dataTimeout = 120
    dataChunkTimeout = 300
    quitTimeout = 300
    origHostname = "$SEI.sei.email.domain"
    port = 25
}

server = {
    helloTimeout= 300
    mailTimeout = 300
}
recipientTimeout = 200
dataTimeout = 120
dataChunkTimeout = 600
receivedEmail = 300
quitTimeout = 300
port = 25
greetHostname = "$SEI.sei.email.domain"

} # server

} # email

SMS = {

replyMsisdn= {

    file = "/IN?Service/tmp/sei-reply.addrMap"
    prefix = "642188"
    maxSuffixDigits = 6
}

newLine = "CR"

numberRules = [
]

protocol = "SMPP"

SMPP = {
    remote = {
        host = "$HOSTNAME"
        port = 3003
    }

    username = "1234"
    password = "PASSWORD"
    preOpen = true
    version = "5.0"
    maxConcurrentTransactions = 1024
    outgoingTimeout = 10
    idleTimeout = 0
    heartbeatInterval = 10

    adapter = {
        lib = "mmxiSMPP.so"
        SSN = 0
        adapterName = "SMPP1"

        config = {
            suppressPathInfoReport = true
            displayZeroPathReport = false
            PathReportingInterval = 60

            smppDefaults = {
            }
        }
    } # adapter
} # SMPP
}

# sei
SEI section

The SEI section contains the SEI.sei section of parameters.

sei section configuration

Here are the configuration of the SEI.sei section of the eserv.config file.

```plaintext
sei = {
    usleep = 10000

    database = {
        user = "mmx_admin"
        password = "mmx_admin"
    }

    email = {
        domain = "$EMAIL_DOMAIN"
        errorMailbox = "error"

        numberRules = {
        }

        postmasterAction = {
            action = "ignore"
        }

        errorNotificationAction = {
            action = "ignore"
        }

        deliveryStatusNotification = {
            directory = "/IN/service_packages/SEI/tmp/dsn"
            failDirectory = "/IN/service_packages/SEI/tmp/fail"
            retries = ["2 hours", "3 minutes"]
        }

        client = {
            connectTimeout = 300
            initialMessageTimeout = 300
            helloTimeout = 300
            mailTimeout = 300
            recipientTimeout = 300
            dataTimeout = 120
            dataChunkTimeout = 300
            quitTimeout = 300
            origHostname = "$SEI.sei.email.domain"
            port = 25
        }
    }
}
```

} # SEI
mailTimeout = 300
recipientTimeout = 200
dataTimeout = 120
dataChunkTimeout = 600
receivedEmail = 300
quitTimeout = 300
port = 25
greetHostname = "$SEI.sei.email.domain"

} # server

} # email

SMS = {

    replyMsisdn= {

        file = "/IN?Service/tmp/sei-reply.addrMap"
        prefix = "642188"
        maxSuffixDigits = 6
    }

    newline = "CR"

    numberRules = []

    protocol = "SMPP"

    SMPP = {

        remote = {

            host = "$HOSTNAME"
            port = 3003

        }

        username = "1234"
        password = "PASSWORD"
        preOpen = true
        version = "5.0"
        maxConcurrentTransactions = 1024
        outgoingTimeout = 10
        idleTimeout = 0
        heartbeatInterval = 10

        adapter = {

            lib = "mmxiSMPP.so"
            SSN = 0
            adapterName = "SMPP1"

            config = {

                suppressPathInfoReport = true
                displayZeroPathReport = false
                PathReportingInterval = 60

                smppDefaults = {
                }

            } # mmxiSMPP.so config

        } # adapter

    } # SMPP

} # SMS
sei parameters

Here are the parameters for SEI.sei section.

usleep

Syntax:      usleep = miSecs
Description: How many micro seconds to sleep if there is nothing to do.
Type:       Integer
Optionality: Mandatory
Default:    
Notes:      10,000 = 0.01 sec
Example:    usleep = 10000

database parameters

The database parameters provide access to the database.

Note: There is no tnsname, so sei will use $ORACLE_SID to find the local database.

password

Syntax:      password = "pass"
Description: The user's password.
Type:       String
Optionality: Mandatory
Default:    
Notes:      
Example:    password = "mmx_admin"

user

Syntax:      user = "username"
Description: The user name.
Type:       String
Optionality: Mandatory
Default:    
Notes:      
Example:    user = "mmx_admin"

e-mail section parameters

Here is a high level view of the parameters in the e-mail section.

    email = {
        domain = "$EMAIL_DOMAIN"
errorMailbox = "error"

numberRules = [
]

postmasterAction = {
    action = "ignore"
}

errorNotificationAction = {
    action = "ignore"
}

deliveryStatusNotification = {
    deliveryStatusNotification_parameters
}

client = {
    clientSection_parameters
}

server = {
    serverSection_parameters
}

}

domain

Syntax:      domain= "name"
Description: The domain to use for recipient email addresses.
Type:        String
Optionality: Mandatory
Allowed:     
Default:     
Notes:       This value matches the first and second levels of the domain, so if domain = "example.com", then:
              host.example.com
              example.com
              will be valid, but:
              badexample.com
              will not be.
              You may use $values.
Examples:    
              domain = "example.com"
              domain = "$EMAIL_DOMAIN"

errorMailbox

Syntax:      errorMailbox = "domain"
Description: The mailbox where email relay failures will be delivered to.
Type:        String
Optionality: Optional
Allowed:     
Default:     
Notes:       If no domain is given (no @) then the full email address is errorMailbox@domain
Example: \[ \text{errorMailbox} = "error" \]

**errorNotificationAction**

**Syntax:** \[ \text{errorNotificationAction} = \{ \text{action} = "enaction" \} \]

**Description:** \(<\text{enaction}>\) is what to do with returned mail notification emails.

**Type:** String

**Optionality:**

**Allowed:**
- "ignore" - silently ignore the emails
- "relay" - relay on to another email address
- "save" - save to disk

**Default:**

**Notes:**

**Example:**

\[
\text{errorNotificationAction} = \{
\begin{array}{l}
\text{action} = "ignore"
\end{array}
\}
\]

**numberRules**

**Syntax:** \[ \text{NumberRules} = \[ \text{rule} \] \]

**Description:** The rules for how to transform the MSISDN in the database into the SMS world and back.

**Type:** Array

**Optionality:** Optional

**Allowed:**

**Default:**

**Notes:** Not used for PME

**Example:**

**postmasterAction**

**Syntax:** \[ \text{postmasterAction} = \{ \text{action} = "pmaction" \} \]

**Description:** What to do with emails directed to the postmaster

**Type:** String

**Optionality:**

**Allowed:**
- "ignore" - silently ignore the emails
- "relay" - relay on to another email address
- "save" - save to disk

**Default:**

**Notes:**

**Example:**

\[
\text{postmasterAction} = \{
\begin{array}{l}
\text{action} = "ignore"
\end{array}
\}
\]

**deliveryStatusNotification parameters**

Here are the parameters for this section, which handles sending of message delivery failures.
deliveryStatusNotification = {
    directory = "/IN/service_packages/SEI/tmp/dsn"
    failDirectory = "/IN/service_packages/SEI/tmp/fail"
    retries = ["2 hours", "3 minutes"]
}

directory
Syntax:   directory = "path"
Description:    The directory to save emails while sending
Type:    String
Optionality:    Mandatory
Allowed:    
Default:    
Notes:    
Example:   directory = "/IN/service_packages/SEI/tmp/dsn"

failDirectory
Syntax:   failDirectory = "path"
Description:    The directory for emails that failed to be sent.
Type:    String
Optionality:    Mandatory
Allowed:    
Default:    
Notes:    This is only used if action = "relay" and saveFailed is true.
   See errorNotificationAction (on page 11) for details.
Example:   failDirectory = "/IN/service_packages/SEI/tmp/fail"

retries
Syntax:   retries = [periods]
Description:    The list of when to retry sending the email after it fails.
Type:    Array of strings
Optionality:    Mandatory
Allowed:    Either periods, for example, "2 hours", "3 minutes"
   or "3 times period" to try 3 times every hour
   "after period" to try that long after the last attempt.
Default:    
Notes:    This is only used if action = "relay". See errorNotificationAction
   (on page 11) for details.
Example:   retries = ["2 hours", "3 minutes"]

client section parameters
Here are the parameters.

    client = {

        connectTimeout = 300
        initialMessageTimeout = 300
        helloTimeout = 300
    }
mailTimeout = 300  
recipientTimeout = 300  
dataTimeout = 120  
dataChunkTimeout = 300  
quitTimeout = 300  
origHostname = "$SEI.sei.email.domain"  
port = 25

} # client

connectTimeout

Syntax:        connectTimeout = seconds
Description:   How long to wait for the TCP connection to complete.
Type:          Integer
Optionality:   Mandatory
Allowed:       in seconds
Default:       
Notes:         
Example:       connectTimeout = 300

dataChunkTimeout

Syntax:        dataChunkTimeout = seconds
Description:   How long to wait for the data chunk response.
Type:          Integer
Optionality:   Mandatory
Allowed:       In seconds
Default:       
Notes:         
Example:       dataChunkTimeout = 300

dataTimeout

Syntax:        dataTimeout = seconds
Description:   How long to wait for the data command response.
Type:          Integer
Optionality:   Mandatory
Allowed:       In seconds
Default:       
Notes:         
Example:       dataTimeout = 120

helloTimeout

Syntax:        helloTimeout = seconds
Description:   How long to wait for the hello command response.
Type:          Integer
Optionality:   Mandatory
Allowed:       In seconds
Default:       
Notes:         
Example: \[ \text{helloTimeout} = 300 \]

**initialMessageTimeout**

Syntax: \[ \text{initialMessageTimeout} = \text{seconds} \]

Description: How long to wait for the initial SMTP message.

Type: Integer

Optionality: Mandatory

Allowed: In seconds

Default:

Notes:

Example: \[ \text{initialMessageTimeout} = 300 \]

**mailTimeout**

Syntax: \[ \text{mailTimeout} = \text{seconds} \]

Description: How long to wait for the mail command response.

Type: Integer

Optionality: Mandatory

Allowed: In seconds

Default:

Notes:

Example: \[ \text{mailTimeout} = 300 \]

**origHostname**

Syntax: \[ \text{origHostname} = \text{name} \]

Description: The hostname we give to SMTP servers.

Type: String

Optionality: Optional

Allowed:

Default:

Notes: No host value means listen for incoming SMTP connections on any interface. You may use $values

Example: \[ \text{origHostname} = "$\text{SEI.sei.email.domain}$" \]

**port**

Syntax: \[ \text{port} = \text{num} \]

Description: The TCP port to connect to for SMTP.

Type: Integer

Optionality: Mandatory

Allowed:

Default:

Notes: Must be 25 in production - only change for testing

Example: \[ \text{port} = 25 \]

**quitTimeout**

Syntax: \[ \text{quitTimeout} = \text{seconds} \]

Description: How long to wait for the quit command response.
Type: Integer
Optionality: Mandatory
Allowed: In seconds
Default:
Notes:
Example: quitTimeout = 300

recipientTimeout

Syntax: recipientTimeout = seconds
Description: How long to wait for the recipient command response.
Type: Integer
Optionality: Mandatory
Allowed: In seconds
Default:
Notes:
Example: recipientTimeout = 300

server section parameters

Here are the parameters.

```plaintext
server = {
    helloTimeout= 300
    mailTimeout = 300
    recipientTimeout = 200
    dataTimeout = 120
    dataChunkTimeout = 600
    receivedEmail = 300
    quitTimeout = 300
    port = 25
    greetHostname = "$SEI.sei.email.domain"
}
```

dataChunkTimeout

Syntax: dataChunkTimeout = seconds
Description: How long to wait for the data chunks to be completed.
Type: Integer
Optionality: Mandatory
Allowed: In seconds
Default:
Notes:
Example: dataChunkTimeout = 600

dataTimeout

Syntax: dataTimeout = seconds
Description: How long to wait for the data command.
Type: Integer
Optionality: Mandatory
Allowed: In seconds
Default:
Notes:
Example: dataTimeout = 120

greetHostname
Syntax: greetHostname = name
Description: The SMTP initial greeting hostname.
Type: String
Optionality: Mandatory
Allowed:
Default:
Notes: You may use $values
Example: greetHostname = "$SEI.sei.email.domain"

helloTimeout
Syntax: helloTimeout = seconds
Description: How long to wait for the hello command.
Type: Integer
Optionality: Mandatory
Allowed: In seconds
Default:
Notes:
Example: helloTimeout = 300

mailTimeout
Syntax: mailTimeout = seconds
Description: How long to wait for the mail command.
Type: Integer
Optionality: Mandatory
Allowed: In seconds
Default:
Notes:
Example: mailTimeout = 300

port
Syntax: port = num
Description: The port to listen for SMTP.
Type: Integer
Optionality: Mandatory
Allowed:
Default:
Notes: Should be 25 for serving the Internet.
Example: port = 25
quitTimeout
Syntax: quitTimeout = seconds
Description: How long to wait for the quit command.
Type: Integer
Optionality: Mandatory
Allowed: In seconds
Default:
Notes:
Example: quitTimeout = 300

receivedEmail
Syntax: receivedEmail= seconds
Description: How long to wait for SEI to process the email.
Type: Integer
Optionality: Mandatory
Allowed: In seconds
Default:
Notes:
Example: receivedEmail = 300

recipientTimeout
Syntax: recipientTimeout = seconds
Description: How long to wait for the recipient command.
Type: Integer
Optionality: Mandatory
Allowed: In seconds
Default:
Notes:
Example: recipientTimeout = 300

SMS section configuration
The SMS section provides the configuration for the interface to SMS functionality.
Here is a high level view of this section.

```bash
SMS = {
    replyMsisdn= {
        replyMsisdn_section_parameters
    }
    newline = "CR"
    numberRules = []
    protocol = "SMPP"
}
```
SMPP = {
  SMPP_section_parameters

    Adapter = {
      Adapter_section_parameters
    }

}

**SMSSection Parameters**

Here are the parameters in this section.

**newLine**

**Syntax:** `newline = "nl"`

**Description:** Defines how newlines are represented in text messages

**Type:** String

**Optionality:** Mandatory

**Allowed:** Options are:

- "CR"
- "LF"
- "CRLF"

**Default:**

**Notes:**

**Example:** `newline = "CR"

**numberRules**

**Syntax:** `NumberRules= [rule]`

**Description:** The rules for how to transform the MSISDN in the database into the SMS world and back.

**Type:** Array

**Optionality:** Optional

**Allowed:**

**Default:**

**Notes:** Not used for PME

**Example:**

**protocol**

**Syntax:** `protocol = "name"

**Description:** The protocol to use.

**Type:** String

**Optionality:** Mandatory

**Allowed:**

**Default:**

**Notes:** This must have a corresponding configuration section.

**Example:** `protocol = "SMPP"`
replyMsisdn section parameters

The `replyMsisdn` configuration section is used to maintain the mappings from incoming emails to the origination address for SMSs and facilitates replies to these SMSs to be directed back to the original emailer.

Here is the configuration of this section.

```plaintext
replyMsisdn= {
    file = "/IN?Service/tmp/sei-reply.addrMap"
    prefix = "642188"
    maxSuffixDigits = 6
}
```

**file**

**Syntax:**

`file = "path"`

**Description:**

Specifies the file to store the mapping in.

**Type:**

String

**Optionality:**

Mandatory

**Allowed:**

Default:

Notes:

**Example:**

`file = "/tmp/sei-reply.addrMap"`

**maxSuffixDigits**

**Syntax:**

`maxSuffixDigits = num`

**Description:**

The maximum number of digits to append to prefix when all of these are used but old ones will be reused.

**Type:**

Integer

**Optionality:**

Mandatory

**Allowed:**

Default:

Notes:

The maximum number 6 = 1,111,111 numbers, 580MB

**Example:**

`maxSuffixDigits = 6`

**prefix**

**Syntax:**

`prefix = "prefix"`

**Description:**

The prefix to use when generating reply SMS addresses.

**Type:**

Number String

**Optionality:**

Mandatory

**Allowed:**

Default:

Notes:

**Example:**

`prefix = "642188"`

SMPP protocol parameters

The SEI uses the SMPP protocol, allowing an ASP to communicate with the SMSC, or an application, such as Messaging Manager, which has a configured SMPP adapter.
Here is high-level view of this section, showing the **SMPP** configuration required for SEI.

```yaml
SMPP = {
    remote = {
        host = "$HOSTNAME"
        port = 3003
    }
    username = "1234"
    password = "PASSWORD"
    preOpen = true
    version = "5.0"
    maxConcurrentTransactions = 1024
    outgoingTimeout = 10
    idleTimeout = 0
    heartbeatInterval = 10
    adapters = {
        adapter_section_parameters
    }
}
```

**heartbeatInterval**

**Syntax:** `heartbeatInterval = hbind`

**Description:** How often to send enquire_link messages to check that the connection is up.

**Type:** Integer

**Optionality:**

**Allowed:**

**Default:**

**Notes:**

**Example:** `heartbeatInterval = 10`

**maxConcurrentTransactions**

**Syntax:** `maxConcurrentTransactions = num`

**Description:** The maximum number of unanswered outstanding messages.

**Type:** Integer

**Optionality:**

**Allowed:**

**Default:**

**Notes:**

**Example:** `maxConcurrentTransactions = 1024`

**outgoingTimeout**

**Syntax:** `outgoingTimeout = seconds`

**Description:** The timeout period before shutting down if quiet for this long.

**Type:** Integer

**Optionality:** Mandatory

**Allowed:**

**Default:**

**Notes:**

**Example:** `outgoingTimeout = 10`
password
Syntax: password = "passw"
Description: The password for the user.
Type: String
Optionality: Mandatory
Allowed: 
Default: 
Notes: 
Example: password = "PASSWORD"

preOpen
Syntax: PreOpen = true|false
Description: Whether or not to open before there are any messages to send.
Type: Boolean
Optionality: Mandatory
Allowed: true, false
Default: 
Notes: 
Example: preOpen = true

username
Syntax: username = "name"
Description: The user name.
Type: String
Optionality: Mandatory
Allowed: 
Default: 
Notes: 
Example: username = "1234"

version
Syntax: version = "ver"
Description: The version of SMPP to use.
Type: String
Optionality: Mandatory
Allowed: Available versions are:
  - "3.4"
  - "5.0"
Default: 
Notes: 
Example: version = "5.0"

remote parameters
The remote section contains the parameters to identify the remote host.
Here is the configuration of the remote section.

```yaml
remote = {
    host = "$HOSTNAME"
    port = 3003
}
```

**host**

**Syntax:**

`host = "host"`

**Description:**

Identifies the host.

**Type:**

String

**Optionality:**

Mandatory

**Allowed:**

hostname, IP address, or $HOSTNAME

**Default:**

**Notes:**

**Example:**

`host = "$HOSTNAME"`

**port**

**Syntax:**

`port= num`

**Description:**

The TCP port to connect to.

**Type:**

Integer

**Optionality:**

Mandatory

**Allowed:**

**Default:**

**Notes:**

**Example:**

`port = 3003`

**adapter parameters**

Here is an example of the adapter section.

```yaml
adapter = {
    lib = "mmxiSMPP.so"
    SSN = 0
    adapterName = "SMPP1"

    config = {
        suppressPathInfoReport = true
        displayZeroPathReport = false
        PathReportingInterval = 60

        smppDefaults = {
        }

    } # mmxiSMPP.so config
} # adapter
```

**adapterName**

**Syntax:**

`adapterName = "adapter"`

**Description:**

The identifier for the adapter.

**Type:**

String

**Optionality:**

Mandatory
Allowed: Any text string, but should be meaningful, i.e. include the protocol used.
   For example "SMPP1" for SMPP
Default: No default.
Note: This name MUST also be in the configuration database before the application will run correctly.
Example: adapterName = "SMPP1"

lib
Syntax: lib = "name"
Description: The name of the file containing the adapter.
Type: String
Optionality: Mandatory
Allowed:
Default: No default
Notes:
Example: lib = "mmxiSMPP.so"

SSN
Syntax: SSN = num
Description: Destination subsystem number of messages to be handled by this adapter.
Allowed: Valid subsystem number
Notes: Non-zero to handle incoming TCAP.
Example: SSN = 18

config
The parameters in this sub-section below this give the configuration for all messages for this adapter.

suppressPathInfoReport
Syntax: suppressPathInfoReport = true|false
Description: Whether or not to suppress path connection reports
Type: boolean
Optionality: Mandatory
Allowed: true, false
Default: false
Notes: An ASP receives heartbeats from the SMSC when Messaging Manager is configured to operate as an SMSC, then it will respond to these heartbeats. These are logged in the xmsTrigger logfile. This can cause the logfile to fill up unnecessarily.
Example: suppressPathInfoReport = true

smppDefaults section
The smppDefaults section of the eserv.config specifies the SMPP values that will be used for connections. Refer to Messaging Manager Technical Guide for details.
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.

sei

25

sei

Purpose

The sei process converts between SMS and email and acts as a gateway.

Startup

The sei process can be run in the following two ways:

- As a SLEE interface capable of triggering IN applications such as ACS. In this case, the sei is started automatically by the SLEE. For more information see SLEE.cfg Configuration.
- As a stand-alone binary which cannot trigger IN applications. In this case, the sei can be started from the command line or from inittab.
  - To start the sei process from the command line, enter:
    `/IN/services_packages/SEI/bin/sei`
  - When SEI is installed an inittab entry is created and SEI started from inittab:
    ```
    sei1:34:respawn:su - sei_oper -c "exec
    /IN/service_packages/SEI/bin/seiStartup.sh >>
    /IN/service_packages/SEI/tmp/sei.log 2>&1
    1" > /dev/null 2>&1 0<&1
    ```

Command line parameters

There are no command line parameters for the sei process.
Chapter 3

Configuration

The configuration parameters for the sei process are automatically added to the SEI section of eserv.config at installation. For details, see SEI Configuration (on page 4).

Failure

If the sei fails, alarms will be raised to the syslog and any incoming inbound emails and SMSs from xmsTrigger not be processed.
Chapter 4

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:

Installation and Removal Overview
Checking the Installation

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

SEI packages

An installation of SMS Email Interface includes the following package, on the SLC:
- seiScp

Checking the Installation

Introduction

Refer to these check lists to ensure the package has been installed correctly.

SEI directories and files

The SEI installation creates the following directories:

- /IN/service_packages/SEI/bin
- /IN/service_packages/SEI/etc
- /IN/service_packages/SEI/lib
- /IN/service_packages/SEI/tmp

The SEI installation installs the following binaries and interfaces:

- /IN/services_packages/SEI/bin/sei
The SEI installation installs the following example configuration files:

- `/IN/service_packages/SEI/etc/eserv.config.example`
- `/IN/service_packages/SEI/etc/eserv.config.pme`

**Error mailbox**

Before the SEI application can be used, a valid address for error messages must be configured. Please update the `errorMailbox` (on page 10) parameter, in the `eserv.config.pme` file in `/IN/service_packages/SEI/etc`, with a valid mailbox then restart SEI through inittab.

**Profile scp file**

The `.profile-scp` file is created in `/IN/service_packages/SEI` when SEI is installed. Here is an example.

```
ORACLE_SID=SCP
export ORACLE_SID
ORACLE_HOME=/u01/app/oracle/product/10/2/0/db_1
export ORACLE_HOME
ORACLE_BASE=/u01/app/oracle
export ORACLE_BASE
ORACLE_TERM=vt100
export ORACLE_TERM
LD_LIBRARY_PATH=${LD_LIBRARY_PATH:+$LD_LIBRARY_PATH:}/u01/app/oracle/product/10/2/0/db_1/lib32:/u01/app/oracle/product/10/2/0/db_1/lib:/usr/lib/secure:$ORACLE_HOME/lib32:$ORACLE_HOME/lib:/IN/service_packages/SEI/lib
export LD_LIBRARY_PATH
PATH=$PATH:$ORACLE_HOME/bin:/IN/service_packages/SEI/bin
export PATH
EMAIL_DOMAIN=mmx3tstscp11-zone03.oracle.com
ESERV_CONFIG_FILE=/IN/service_packages/SEI/etc/eserv.config
export EMAIL_DOMAIN ESERV_CONFIG_FILE
```
NCC Glossary of Terms

ACS
Advanced Control Services configuration platform.

ASP
- Application Service Provider, or

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

cron
Unix utility for scheduling tasks.

crontab
File used by cron.

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

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

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

IN
Intelligent Network

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

IP address
Internet Protocol Address - network address of a card on a computer
ISDN
Integrated Services Digital Network - set of protocols for connecting ISDN stations.

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

MM
Messaging Manager. Formerly MMX, see also XMS (on page 31) and Messaging Manager (on page 30).

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).

Oracle
Oracle Corporation

Peer
Remote machine, which for our purposes is capable of acting as a Diameter agent.

SGML

SLC
Service Logic Controller (formerly UAS).

SLEE
Service Logic Execution Environment

SMPP
Short Message Peer-to-Peer protocol

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

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

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

SSP
Service Switching Point

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

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

VWS
Oracle Voucher and Wallet Server (formerly UBE).

XMS
Three letter code used to designate some components and path locations used by the Oracle Communications Network Charging and Control Messaging Manager (on page 30) service and the Short Message Service. The published code is MM (on page 30) (formerly MMX).