DECT Messenger Installation and Commissioning — Book 2
Avaya Communication Server 1000

Release 7.6
NN43120-301-B2
Issue 04.01 Standard
March 2013

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

“Documentation” means information published by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya generally makes available to users of its products. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of documentation unless such modifications, additions, or deletions were performed by Avaya. End User agrees to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked websites referenced within this site or documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on its hardware and Software (“Product(s)”). Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya’s standard warranty language, as well as information regarding support for this Product while under warranty is available to Avaya customers and other parties through the Avaya Support website: http://support.avaya.com. Please note that if you acquired the Product(s) from an authorized Avaya reseller outside of the United States and Canada, the warranty is provided to you by said Avaya reseller and not by Avaya. “Software” means computer programs in object code, provided by Avaya or an Avaya Channel Partner, whether as stand-alone products or pre-installed on hardware products, and any upgrades, updates, bug fixes, or modified versions.

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, HTTP://SUPPORT.AVAYA.COM/LICENSEINFO ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA INC., ANY AVAYA AFFILIATE, OR AN AUTHORIZED AVAYA RESELLER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AUTHORIZED AVAYA RESELLER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA AUTHORIZED RESELLER; AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA INC. OR THE APPLICABLE AVAYA AFFILIATE (“AVAYA”).

Heritage Nortel Software

“Heritage Nortel Software” means the software that was acquired by Avaya as part of its purchase of the Nortel Enterprise Solutions Business in December 2009. The Heritage Nortel Software currently available for license from Avaya is the software contained within the list of Heritage Nortel Products located at http://support.avaya.com/LicenseInfo under the link “Heritage Nortel Products”. For Heritage Nortel Software, Avaya grants Customer a license to use Heritage Nortel Software provided hereunder solely to the extent of the authorized activation or authorized usage level, solely for the purpose specified in the Documentation, and solely as embedded in, for execution on, or (in the event the applicable Documentation permits installation on non-Avaya equipment) for communication with Avaya equipment. Charges for Heritage Nortel Software may be based on extent of activation or use authorized as specified in an order or invoice.

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, or hardware provided by Avaya. All content on this site, the documentation and the Product provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software unless expressly authorized by Avaya. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil offense under the applicable law.

“Third Party Components” mean certain software programs or portions thereof included in the Software that may contain software (including open source software) distributed under third party agreements (“Third Party Components”), which contain terms regarding the rights to use certain portions of the Software (“Third Party Terms”). Information regarding distributed Linux OS source code (for those Products that have distributed Linux OS source code) and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply is available in the Documentation or on Avaya’s website at: http://support.avaya.com/Copyright. You agree to the Third Party Terms for any such Third Party Components.

Note to Service Provider

The Product may use Third Party Components that have Third Party Terms that do not allow hosting and may need to be independently licensed for such purpose.

Preventing Toll Fraud

“Toll Fraud” is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf). Be aware that there can be a risk of Toll Fraud associated with your system and that, if Toll Fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya Toll Fraud intervention

If you suspect that you are being victimized by Toll Fraud and you need technical assistance or support, call Technical Service Center Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see the Avaya Support website: http://support.avaya.com. Suspected security vulnerabilities with Avaya products should be reported to Avaya by sending mail to: securityalerts@avaya.com.

The trademarks, logos and service marks (“Marks”) displayed in this site, the Documentation and Product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the Documentation and Product(s) should be construed as granting, by implication, estoppel, or otherwise,

any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

Avaya is a registered trademark of Avaya Inc.

All non-Avaya trademarks are the property of their respective owners, and “Linux” is a registered trademark of Linus Torvalds.

Downloading Documentation

For the most current versions of Documentation, see the Avaya Support website: http://support.avaya.com.

Contact Avaya Support

See the Avaya Support website: http://support.avaya.com for product notices and articles, or to report a problem with your Avaya product. For a list of support telephone numbers and contact addresses, go to the Avaya Support website: http://support.avaya.com, scroll to the bottom of the page, and select Contact Avaya Support.

Contents

Chapter 1: Module - eSMTP

The eSMTP module is an output program that receives message requests from the eKERNEL module. The eSMTP connects to an SMTP server, and delivers mail requests to the mail server according to the RFC821 specifications. This involves a sockets connection between eSMTP and the SMTP server of choice. For such a connection, eSMTP is TCP client and the SMTP server is TCP server, listening on port 25.

Initialization

The eSMTP module is started by means of a shortcut. Figure 1: Example of required keywords on page 17 shows an example of the required keywords:

```
"C:\SOPHO Messenger@Net\Exe\eSMTP.exe"
/Site:3
/eKernel address:*LOCAL
/eKernel port:3111
/Log drive:C
```

Figure 1: Example of required keywords

The following keywords are used:

- **Site**

  The Site keyword denotes the site that is assigned to the eSMTP module.

- **eKERNEL address**

  The eKERNEL address keyword denotes the IP address that is assigned to the eKERNEL module. The eSMTP contacts this IP address to connect to the eKERNEL.

- **eKERNEL por**

  The eKERNEL port keyword denotes the port number that is assigned in the configuration for the eSMTP client instance.

On startup, the eSMTP application attempts to connect to the eKERNEL. This is performed based upon the address and port information obtained from the shortcut.

At connection, the eSMTP requests the eKERNEL to provide additional configuration settings. This is known as a configuration request. The eKERNEL in turn authenticates the client and responds with a configuration reply.

Figure 2: eSMTP configuration request on page 18 shows the configuration request.

```
28/10/2001 15:28:39 - S:INF:
Application eSMTP - SOPHO Messenger@Net - v2.0.7 started with parameters
/Site:3 /eKernel address:*LOCAL /eKernel port:3111 /Log drive:C

28/10/2001 15:28:40 - S:INF:
TCP local port 01065 connected with remote port 03111 (eKERNEL)

28/10/2001 15:28:40 - O:TCP:
<xml>
<cfgrqs>
<appl>eSMTP</appl>
<site>3</site>
</cfgrqs></xml>

28/10/2001 15:28:40 - I:TCP:
<xml>
<cfgrpy>
<smtp_address>127.0.0.1</smtp_address>
<smtp_port>25</smtp_port>
<smtp_domain>GNTN1SFMI.ibsbe.be</smtp_domain>
<email_from>Messenger@Net</email_from>
<format>32^0^0^0^0</format>
<log_path>C:\SOPHO Messenger@net</log_path>
<log_days>1</log_days>
</cfgrpy>
</xml>
```

Figure 2: eSMTP configuration request

When the configuration is received, a window similar to the one shown Figure 3: Configuration information on page 18 opens. The configuration can be viewed in the Connections tab.

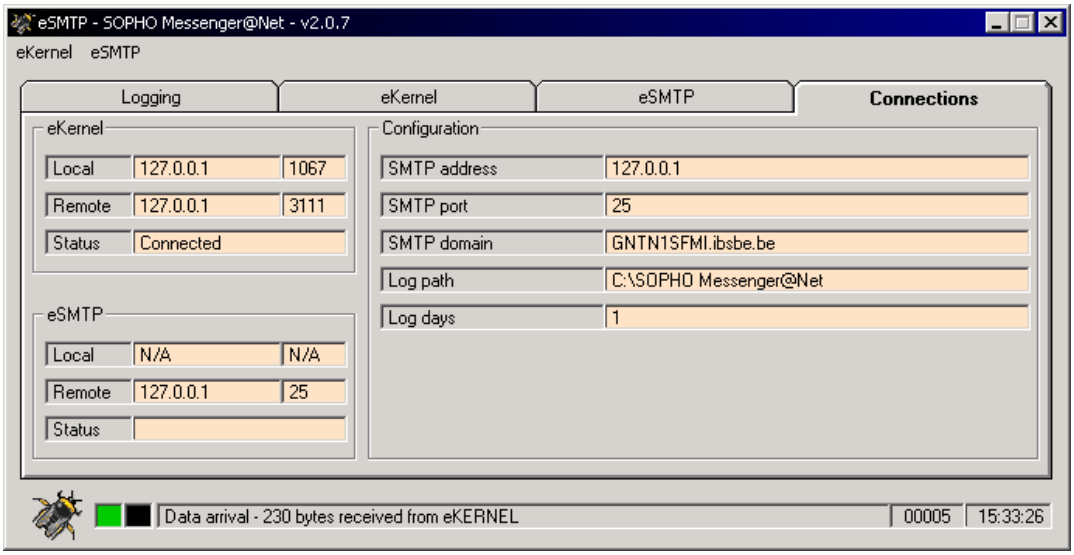


Figure 3: Configuration information

Output program activity

The eSMTP module is now ready to receive message requests from eKERNEL. These requests are handled on a first-in first-out basis.

The requests are received in the format shown in Figure 4: Message request format on page 19.

```
<xml>
<msgrqs>
<id>00251</id>
<to>befmi@1s.be</to>
<pag_01>Test to eSMTP</pag_01>
<pag_more>N</pag_more>
</msgrqs>
</xml>
```

Figure 4: Message request format

The message requests are executed one at a time, by means of a TCP sockets connection to the SMTP server of choice. The actual dialog box with the SMTP server can be monitored through the eSMTP tab, as shown in Figure 5: eSMTP tab on page 19.

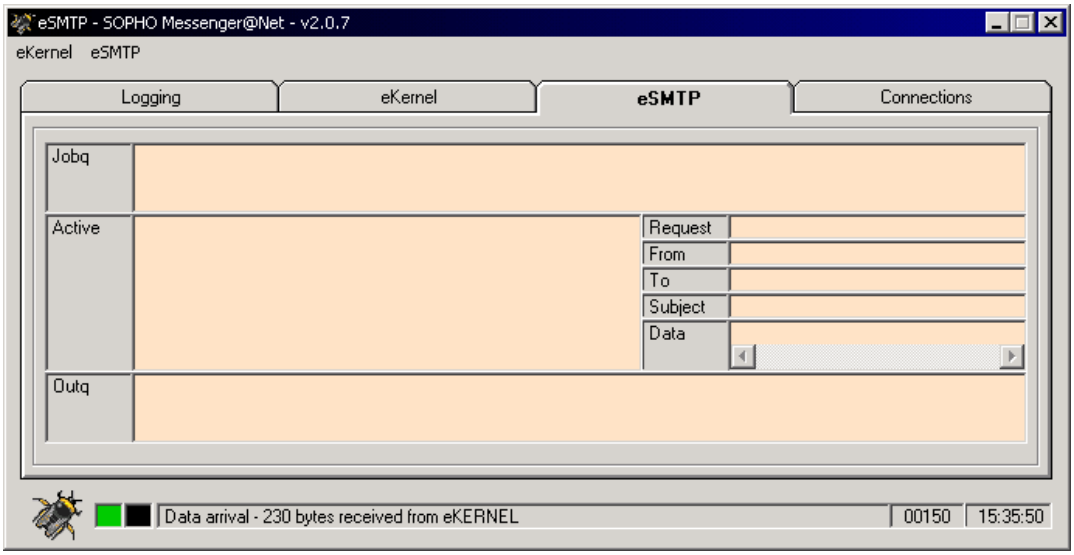


Figure 5: eSMTP tab

The eSMTP tab provides an overview of the requests that are waiting to be processed. This is visible in the top area (referred to as the job queue). Requests are handled as follows:

Request handling

1. The request is analyzed and the required keywords are extracted and shown to the right.

   The left-hand side of the window shows the actual dialog with the SMTP server. See Figure 6: Request queue with extracted keywords on page 20 for an example of an active message.

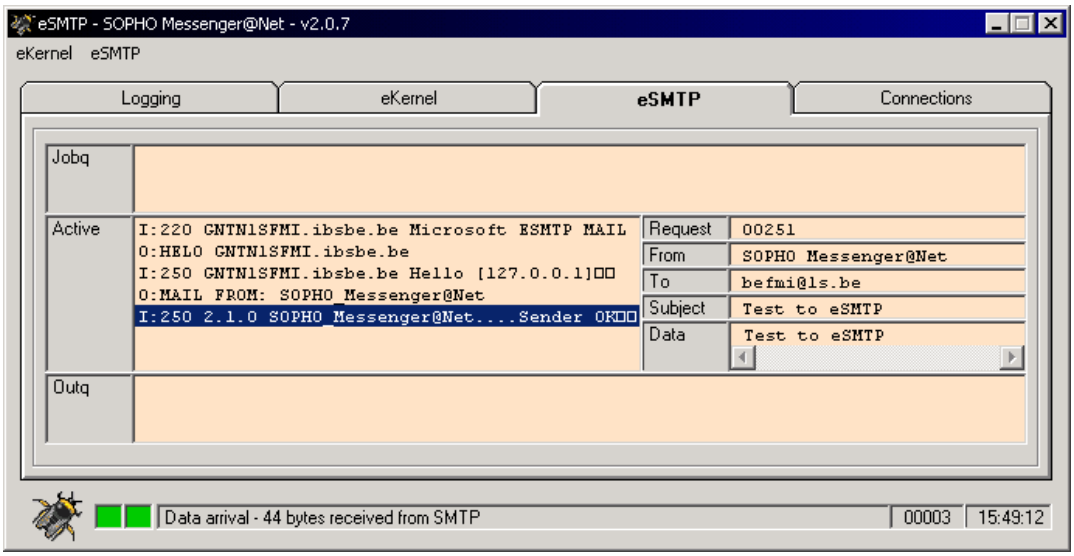


Figure 6: Request queue with extracted keywords

2. The eSMTP module sends the status of the request back to the eKERNEL. This status can either indicate a positive acknowledge or a negative acknowledge.

   The format of the message reply is shown in Figure 7: Message reply format on page 20.

```
<xml>
<msgrpy>
<id>00251</id>
<sts>ACK^</sts>
</msgrpy>
</xml>
```

Figure 7: Message reply format

3. The e-mail message is delivered to the mailbox of the destination user.

   Note that intermediate processing on the external SMTP server or servers is responsible for message delivery. This process is completely out of the control of the eSMTP application.

   Figure 8: Example of mail produced by eSMTP module on page 21 shows an example of the mail that is produced by the eSMTP module, when viewed using Microsoft Outlook Express.

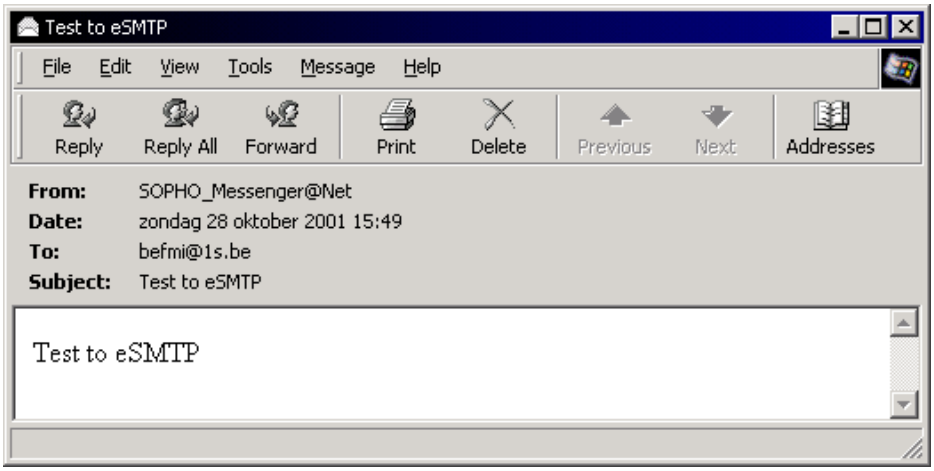


Figure 8: Example of mail produced by eSMTP module

Figure 8: Example of mail produced by eSMTP module on page 21 shows an example of the raw data of the mail that is produced by the eSMTP module.

```
Received: from GNTN1SFMI.ibsbe.be ([127.0.0.1]) by GNTN1
with Microsoft SMTPSVC(5.0.2195.2966);
 Sun, 28 Oct 2001 15:49:14 +0100
From: SOPHO_Messenger@Net
To: befmi@1s.be
Subject: Test to eSMTP
Return-Path: SOPHO_Messenger@Net
Message-ID: <GNTN1SFMIF60lTy3RuX00000002@GNTN1SFMI.ibsbe
X-OriginalArrivalTime: 28 Oct 2001 14:49:15.0119 (UTC) F
TIME=[B6ABCFF0:01C15FBF]
Date: 28 Oct 2001 15:49:15 +0100

Test to eSMTP
```

Figure 9: Raw data of mail produced by eSMTP module

Logging

The eSMTP application provides logging both on-screen and on disk.

Figure 10: eSMTP on-screen logging on page 22 shows the on-screen logging, displayed on the Logging tab.

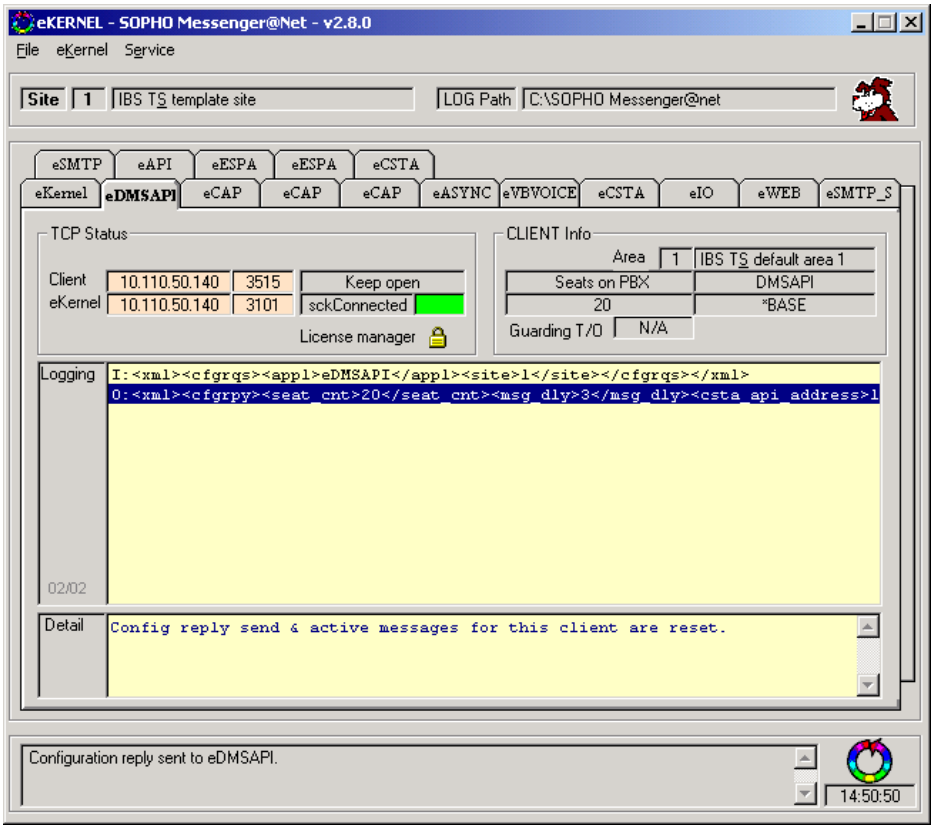


Figure 10: eSMTP on-screen logging

The following two figures show an example of a log file on disk, as viewed with a text editor.

```
28/10/2001 14:46:32 - I:TCP:
221 2.0.0 GNTN1SFMI.ibsbe.be Service closing transmission channel

28/10/2001 14:46:33 - O:TCP:
<xml><msgrpy><id>00001</id><sts>NACK - 550 5.7.1 Unable to relay for
francis.missiaen@1s.be^</sts></msgrpy></xml>

28/10/2001 14:51:10 - S:INF:
TCP local port 01063 connected with remote port 00025 (eDMSAPI)

28/10/2001 14:51:10 - I:TCP:
220 GNTN1SFMI.ibsbe.be Microsoft ESMTP MAIL Service, Version:
5.0.2195.2966 ready at  Sun, 28 Oct 2001 14:51:10 +0100

28/10/2001 14:51:11 - O:TCP:
HELO GNTN1SFMI.ibsbe.be

28/10/2001 14:51:11 - I:TCP:
250 GNTN1SFMI.ibsbe.be Hello [127.0.0.1]

28/10/2001 14:51:12 - O:TCP:
MAIL FROM: SOPHO.Messenger@Net
28/10/2001 14:51:12 - I:TCP:
250 2.1.0 SOPHO.Messenger@Net....Sender OK

28/10/2001 14:51:13 - O:TCP:
RCPT TO: francis.missiaen@1s.be

28/10/2001 14:51:13 - I:TCP:250 2.1.5 francis.missiaen@1s.be

28/10/2001 14:51:14 - O:TCP:
DATA

28/10/2001 14:51:14 - I:TCP:
354 Start mail input; end with <CRLF>.<CRLF>

28/10/2001 14:51:15 - O:TCP:
From: SOPHO.Messenger@Net

continued on next page...
```

Figure 11: Log files on hard disk — part 1

```
28/10/2001 14:51:15 - O:TCP:
To: francis.missiaen@1s.be

28/10/2001 14:51:15 - O:TCP:
Subject: REA K100

28/10/2001 14:51:15 - O:TCP:

28/10/2001 14:51:15 - O:TCP:
REA K100

28/10/2001 14:51:15 - O:TCP:
.

28/10/2001 14:51:15 - I:TCP:
250 2.6.0 <GNTN1SFMIrywNUG0Vy100000001@GNTN1SFMI.ibsbe.be> Queued mail
for delivery

28/10/2001 14:51:16 - O:TCP:
quit

28/10/2001 14:51:16 - I:TCP:
221 2.0.0 GNTN1SFMI.ibsbe.be Service closing transmission channel

28/10/2001 14:51:17 - O:TCP:
<xml><msgrpy><id>00001</id><sts>ACK^</sts></msgrpy></xml>
```

Figure 12: Log files on hard disk — part 2

Relaying and Routing

Important:

A common configuration error, related to relaying and routing settings, occurs when eSMTP tries to deliver a message to a mail destination user that is not residing in the same domain, as shown in Figure 13: Relaying and Routing error on-screen on page 25.

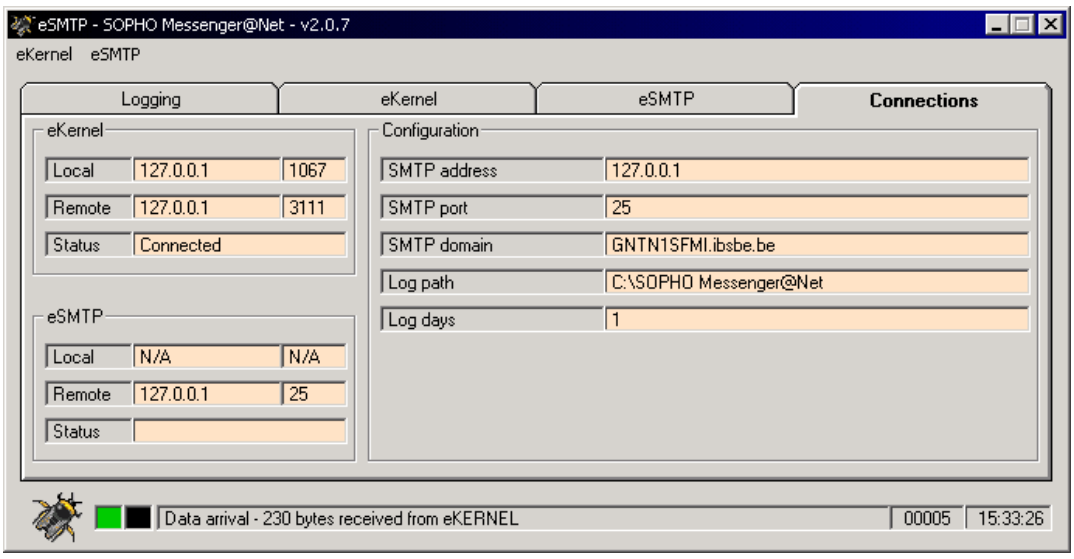


Figure 13: Relaying and Routing error on-screen

The error is usually recorded in the log files with a message similar to the one shown in Figure 14: Relaying error log (relay failed) on page 25.

```
:

28/10/2001 14:46:31 - O:TCP:
RCPT TO: francis.missiaen@1s.be

28/10/2001 14:46:31 - I:TCP:
550 5.7.1 Unable to relay for francis.missiaen@1s.be

:
```

Figure 14: Relaying error log (relay failed)

Other messages can be shown instead, for example, 550 - prohibited, 550 - Unable to relay, and so on.

To correct this issue, consult with the system administrator regarding the rights granted for routing and relaying in the module. Avaya recommends that the IP address of eSMTP be defined in the SMTP server of the mail platform, so that eSMTP is allowed to send mail to destinations that are not in the local domain.

The related configuration issues are beyond the scope of this document. In the following pages, configuration information is shown for illustration only. Look for a more detailed discussion of relaying and routing issues in the official documentation for your SMTP server (Windows 2000, Exchange, Domino, iSeries 400, and so on).

Windows SMTP server

In Windows SMTP Server (part of the Internet Information Server), you can for instance grant access by clicking **Start** on the Windows task-bar, and choosing **Settings > Control Panel > Administrative Tools > Properties > Internet Service Manager**.

Figure 15: Setting SMTP relay on page 26 illustrates the settings needed to grant the SMTP server access to relay from both 127.0.0.1 and 10.110.50.138. These addresses are the addresses where eSMTP modules reside.

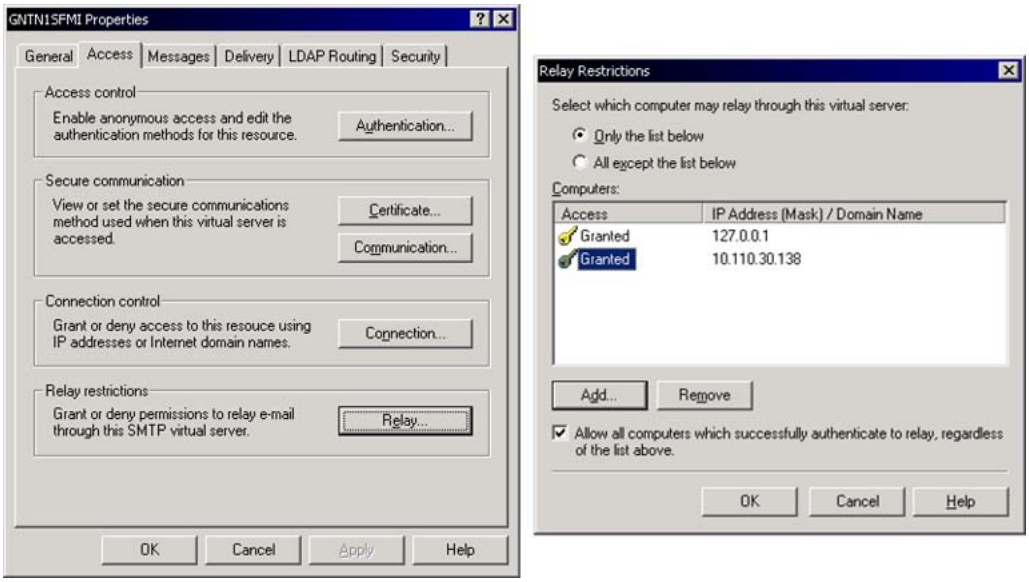


Figure 15: Setting SMTP relay

eSMTP can send mail to users that do not reside in the local domain. This is indicated in the log as shown in Figure 16: Relaying successful on page 26.

```
:
:
28/10/2001 14:51:13 - O:TCP:
RCPT TO: francis.missiaen@1s.be

28/10/2001 14:51:13 - I:TCP:
250 2.1.5 francis.missiaen@1s.be
:
:
```

Figure 16: Relaying successful

Domino (Lotus Notes)

The same techniques discussed for Windows SMTP server on page 26 can be implemented on other SMTP servers. For example, in Domino (Lotus Notes), you can allow inbound SMTP requests from other parties (eSMTP).

To configure inbound SMTP options, click **Router/SMTP > Restrictions and Controls > SMTP Inbound Controls > Allows messages only from** Figure 17: Enable messages from external hosts to be sent to external Internet domains on page 27 illustrates the settings needed to allow messages from external hosts to be sent to external Internet domains.

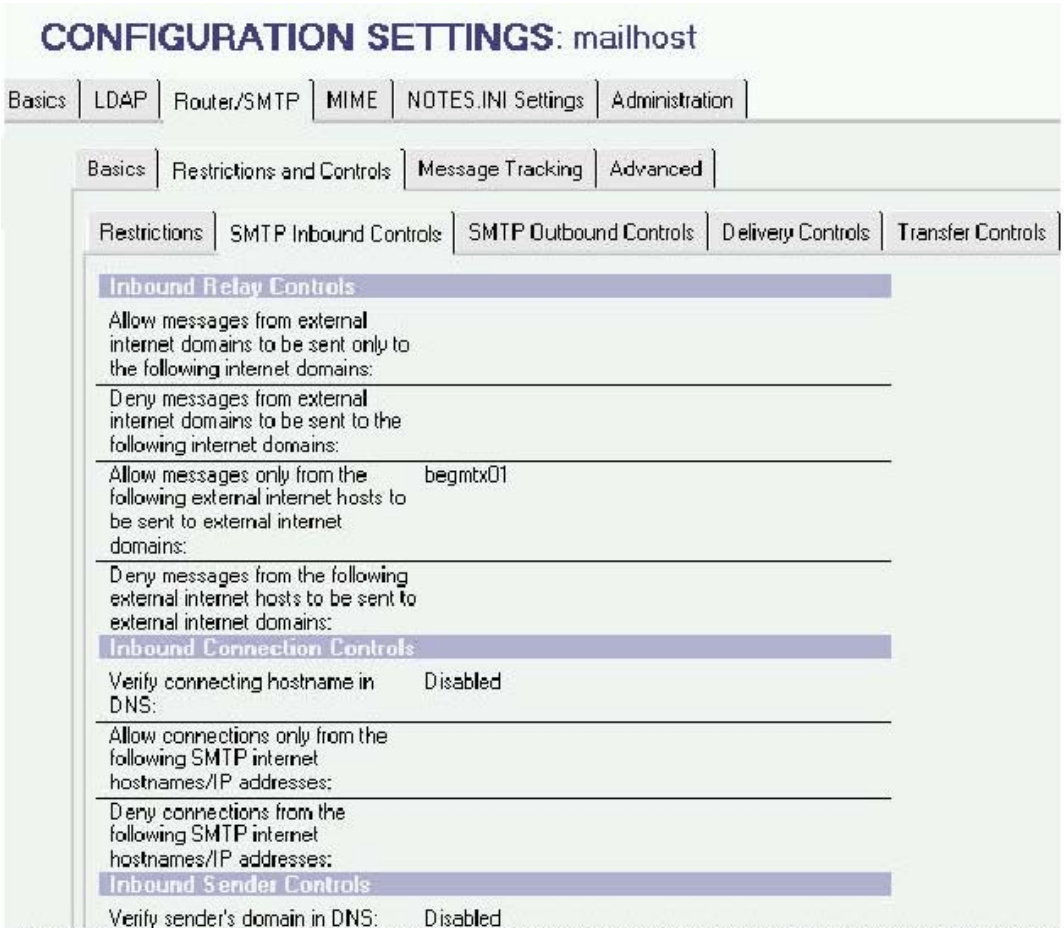


Figure 17: Enable messages from external hosts to be sent to external Internet domains

Consult with your network administrator for more information on configuration aspects and network design.

Chapter 2: Module - eSMTP_server

The eSMTP_server module is a member of the input program family. Therefore, the eSMTP_server is capable of generating alarms to eKERNEL.

The name eSMTP_server can be rather confusing. In fact, there is no SMTP server functionality implemented in the module. This means the application is not acting as an SMTP server, and is not listening on port 25 for inbound SMTP requests.

The module eSMTP_server must always be seen in conjunction with the SMTP Server component that is shipped with Windows, as part of the Internet Information Server software.

The actual role of SMTP server (handling inbound sockets connections on port 25) is played by the Microsoft component. This component stores inbound mails in a directory structure, as specified during configuration of the Microsoft component.

A typical configuration sends inbound mails to the directory c:\inetpub\mailroot\drop.

These e-mail files are in fact readable text-files that can be opened with a text editor, such as Notepad. Figure 18: Example of inbound e-mail on page 30 shown an example of an inbound e-mail:

```
x-sender: francis.missiaen@1s.be
x-receiver: kristien.daneels@1s.be
Received: from gntn1sfmi ([10.110.49.102]) by GNTN1SFMI.ibsbe.be with Mi-
crosoft SMTPSVC(5.0.2195.2966);
 Wed, 27 Jun 2001 14:50:25 +0200
From: beibsbru@ibsbe.be
To: kristien.daneels@1s.be
Subject: Reanimation
MIME-Version:1.0
Content-Type: multipart/mixed; boundary="--
_=_NextPart_000_01C07713.6DAD45D0"
Content-Disposition: inline
Return-Path: beibsbru@ibsbe.be
Message-ID: <GNTN1SFMIfznRukVykX00000004@GNTN1SFMI.ibsbe.be>
X-OriginalArrivalTime: 27 Jun 2001 12:50:25.0293 (UTC) FILE-
TIME=[BC2773D0:01C0FF07]
Date: 27 Jun 2001 14:50:25 +0200

----_=_NextPart_000_01C07713.6DAD45D0
Content-type: text/html
Content-transfer-encoding: binary

<html>
<body bgcolor='#FFFFFF' link='#336699' alink='#336699'>
:
:
:
</body></html>

----_=_NextPart_000_01C07713.6DAD45D0
Content-type: text/plain; charset=iso-8859-1
Content-Disposition: attachment; filename="Attach_0.txt"
Content-transfer-encoding: binary

:
:
:
----_=_NextPart_000_01C07713.6DAD45D0--
```

Figure 18: Example of inbound e-mail

Important:

There are many competing specifications for mail formatting. A basic implementation is specified in RFC821. Many other specifications were added, for example, RFC1251 described the MIME format. The current release of eSMTP_server is not designed to be fully compatible with all available functionality embedded in e-mail messages. Future releases of the eSMTP_server can be enhanced with, for instance, functionality that is capable of detaching media streams (for example, BASE64 encoded audio/wave contents).

Keyword processing

For the purpose of illustration, examples in this chapter ignore all mail contents, and process only the following keywords:

- **x-sender**. The value of the x-sender tag is stored.
- **x-receiver**. The value of the x-receiver tag is stored.
- **Subject**. The value of the Subject: tag is stored.

Because the x-sender and x-receiver tags are Microsoft proprietary, the module eSMTP_server also looks for **From**and **To**keywords, if the x-sender and x-receiver tags are missing. Although not officially supported, it is possible to use the eSMTP_server in environments that work with other SMTP Servers than the one officially supported (Microsoft Internet Information Server).

The information in Figure 19: Keyword processing of selected e-mail tags on page 31 is stored for further processing.

```
<from>francis.Missiaen@1s.be</from>
<to>kristien.daneels@1s.be</to>
<subject>Reanimation</subject>
```

Figure 19: Keyword processing of selected e-mail tags

Initialization

The eSMTP_server is started by means of a shortcut. This shortcut contains required parameters illustrated in Figure 20: Shortcut parameters on page 31.

```
"C:\SOPHO Messenger@Net\Exe\eSMTP_server.exe"
/Site:3
/eKernel address:*LOCAL
/eKernel port:3110
/Log drive:C
```

Figure 20: Shortcut parameters

The following keywords are used:

- **Site**

  The Site keyword denotes the site that is assigned to the eSMTP_server module.

- **eKERNEL**

The eKernel address keyword denotes the IP address that is assigned to the eKERNEL module. The eSMTP_server contacts this IP address to connect to the eKERNEL.

- **eKERNEL Port**

  The eKernel Port keyword denotes the port number that is assigned in the configuration for the eSMTP_server instance.

On startup, the eSMTP_server application attempts to connect to the eKERNEL, as shown in Figure 21: eKERNEL connection attempt on page 32. This is performed based upon the address and port information obtained from the Shortcut.

```
28/10/2001 16:08:07 - S:INF:
Application eSMTP_server - SOPHO Messenger@Net - v2.0.7 started with pa-
rameters /Site:3 /eKernel address:*LOCAL /eKernel port:3110 /Log drive:C

28/10/2001 16:08:08 - S:INF:
TCP local port 01127 connected with remote port 03110 (eKERNEL)
```

Figure 21: eKERNEL connection attempt

At connection, the eSMTP_server requests the eKERNEL to provide additional configuration settings, as shown in Figure 22: Configuration request on page 33. The eKERNEL authenticates the client and responds with a configuration reply, as shown in Figure 23: Configuration reply on page 33.

```
228/10/2001 16:08:08 - O:TCP:
<xml><cfgrqs><appl>eSMTP_server</appl><site>3</site></cfgrqs></xml>

28/10/2001 16:08:08 - I:TCP:<xml>
<cfgrpy><email_dir>c:\inetpub\mailroot\drop</email_dir><poll_intv>10</
poll_intv><email_dir_processed>c:\inetpub\mailroot\drop\processed</
email_dir_processed><keep_processed>5</
keep_processed><email_dir_error>c:\inetpub\mailroot\drop\error</
email_dir_error><keep_error>5</keep_error><log_path>C:\SOPHO Messen-
ger@net</log_path><log_days>1</log_days> </cfgrpy></xml>
```

Figure 22: Configuration request

```
<xml>
<cfgrpy>
<email_dir>c:\inetpub\mailroot\drop</email_dir>
<poll_intv>10</poll_intv>
<email_dir_processed>c:\inetpub\mailroot\drop\processed</
email_dir_processed>
<keep_processed>5</keep_processed>
<email_dir_error>c:\inetpub\mailroot\drop\error</email_dir_error>
<keep_error>5</keep_error>
<log_path>C:\SOPHO Messenger@net</log_path>
<log_days>1</log_days>
</cfgrpy>
</xml>
```

Figure 23: Configuration reply

When the configuration is received, the Connections tab of the eSMTP_server module is updated with information similar to what is shown the panel shown in Figure 24: Updated eSMTP Connection information on page 33.

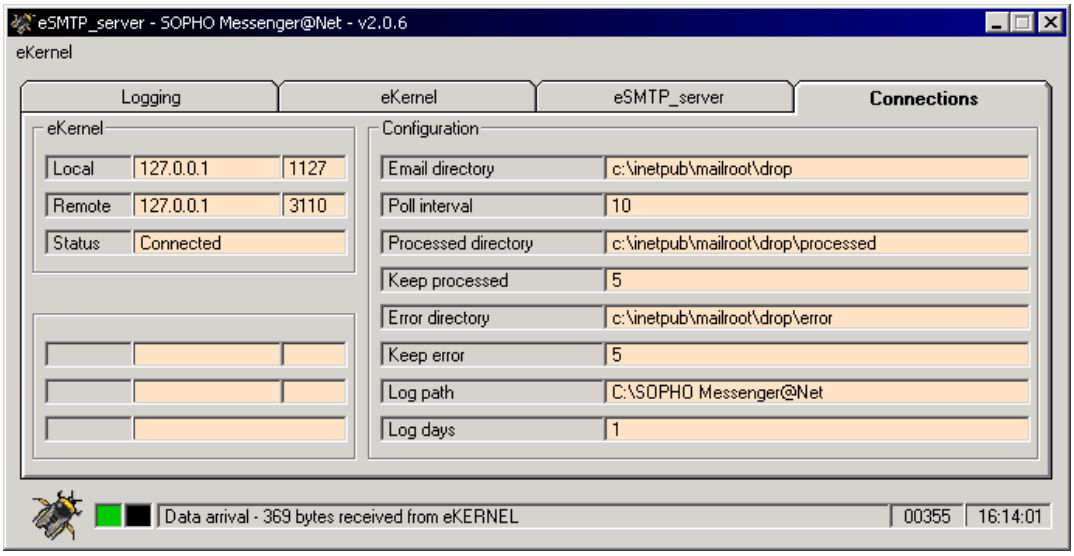


Figure 24: Updated eSMTP Connection information

Activity of eSMTP_server

The eSMTP_server module is now ready to send message requests to eKERNEL. These requests are sent on a first-in first-out basis.

Click the eSMTP_server tab to view request processing, as shown in Figure 25: Request processing shown on the eSMTP_server tab on page 34.

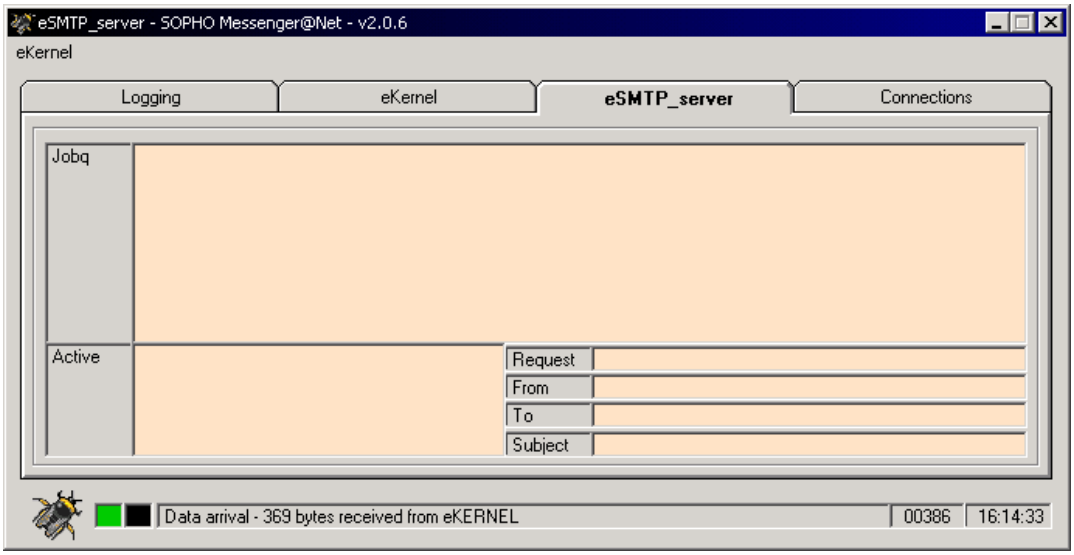


Figure 25: Request processing shown on the eSMTP_server tab

As specified in the configuration reply, the eSMTP_server polls the specified directory for new inbound mail messages at fixed intervals. This interval is usually 10 seconds. The default directory is C:\Inetpub\mailroot\Drop, as shown in Figure 26: Default inbound mail (drop) directory on page 35.

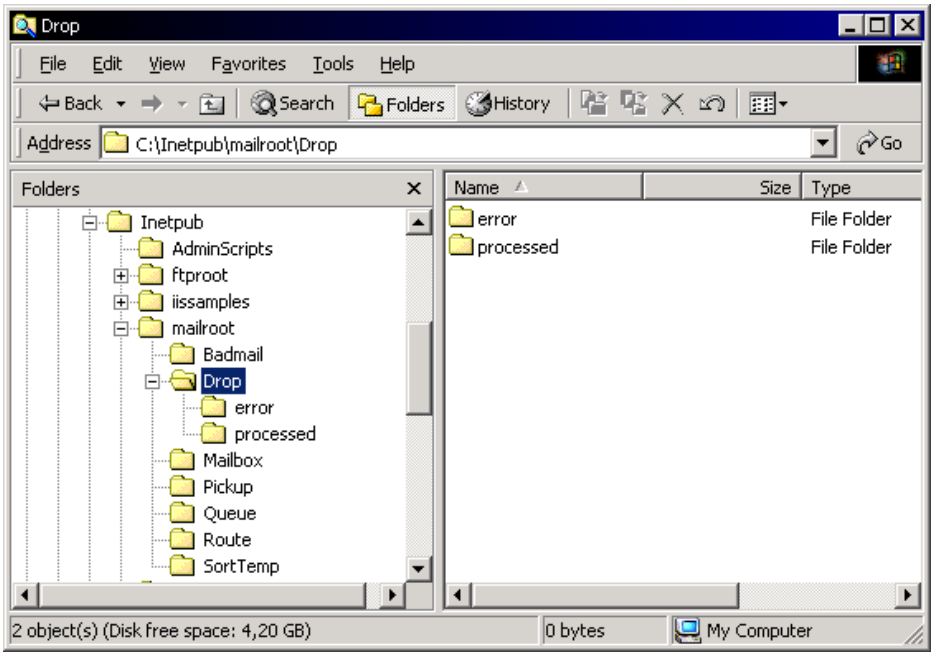


Figure 26: Default inbound mail (drop) directory

Inbound mail messages are processed one by one. During processing, a window opens similar to the one shown in Figure 27: Mail processing on page 35.

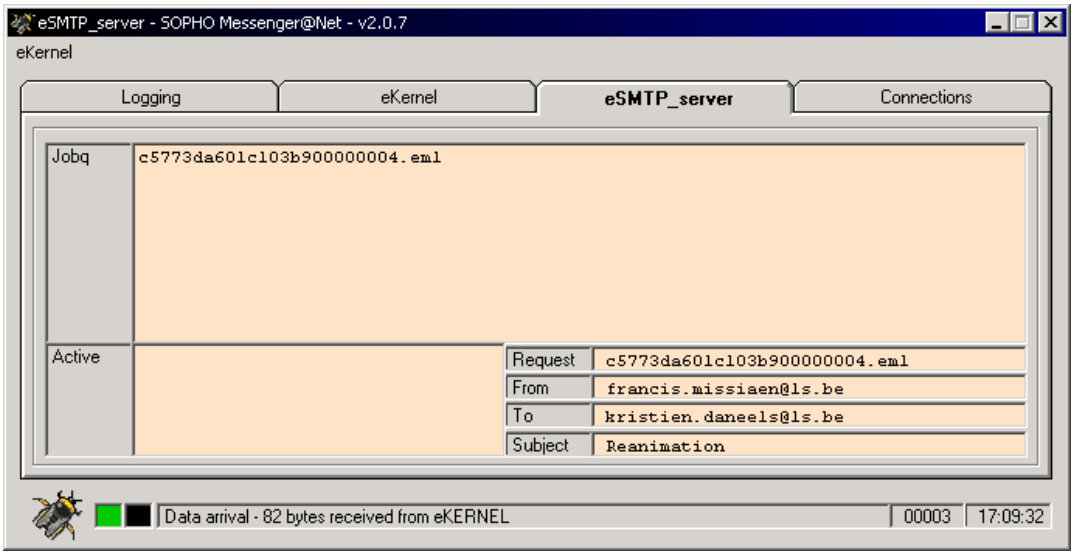


Figure 27: Mail processing

The Mail processing window shows:

- **Request identifier**

This is a long filename and refers to the filename of the e-mail message that is being processed. These names were generated by the Microsoft SMTP Server component.

- **From** field

  Isolated from the <x-receiver> tag.

- **To** field

  Isolated from the <x-sender> tag.

- **Subject** field

  Isolated from the Subject: tag.

With these values, the eSMTP_server produces a message request for eKERNEL, as shown in Figure 28: eSMTP message request for eKERNEL on page 36.

```
<xml>
<msgrqs>
<id>bc6c51d001c0ff0700000004.eml</id>
<from>francis.missiaen@1s.be</from>
<to>kristien.daneels@1s.be</to>
<subject>Reanimation</subject>
</msgrqs>
</xml>
```

Figure 28: eSMTP message request for eKERNEL

The eKERNEL then validates the message request, and either accepts or refuses the request. During the validation process, the eSMTP_server is considered as an input program, so all configuration settings must be defined correctly. One major criterion is whether for this input program the auto-create group is activated. Without auto creation of groups, both **From** and **To** must be known in the database.

- **Message Accepted**

  If the message is accepted, a reply is sent, as shown in Figure 29: Message reply: accepted on page 36.

```
<xml>
<msgrpy>
<id>bc6c51d001c0ff0700000004.eml</id>
<sts>ACK^</sts>
</msgrpy>
</xml>
```

Figure 29: Message reply: accepted

Upon receiving this acknowledgement, the eSMTP_server moves the original mail message to a processed location, unless the target directory is set to a value of *NONE. Figure 30: Specifying the location to file accepted messages on page 37 shows the target folder for accepted messages.

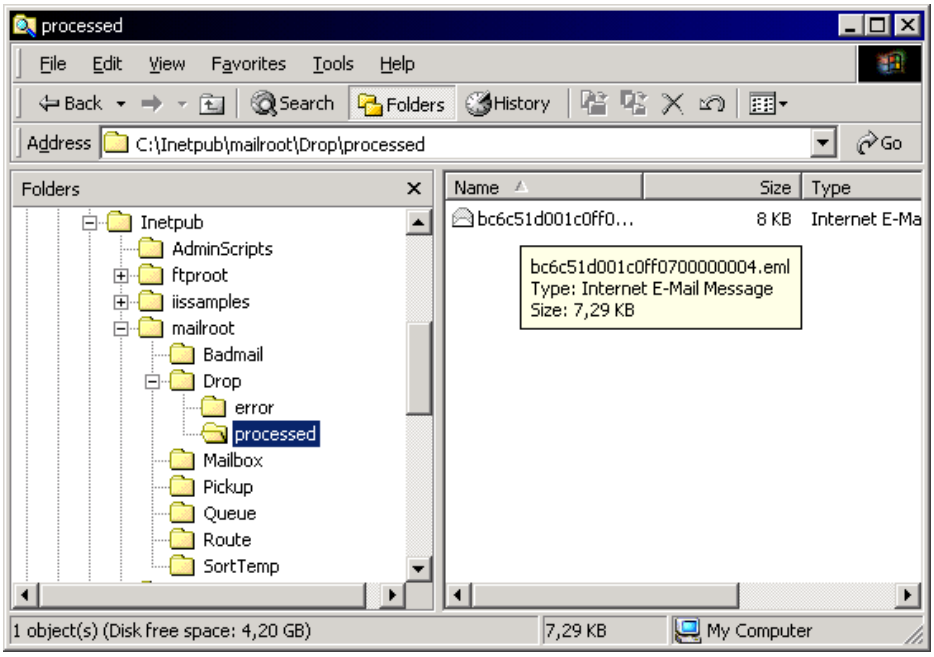


Figure 30: Specifying the location to file accepted messages

- **Message Rejected**

If the message is not accepted in eKERNEL, a negative reply is sent, as shown in Figure 31: Message reply: rejected on page 37.

```
<xml>
<msgrpy>
<id>bc6c51d001c0ff0700000004.eml</id>
<sts>NACK^</sts>
</msgrpy>
</xml>
```

Figure 31: Message reply: rejected

Refer to the log files of eKERNEL (see the **eKERNEL > Logging** tab) to find out why the message was not accepted. Following is an example of the informational message that is shown:

```
S: Alarm not processed. Unknown group in eKERNEL_GROUP table! Auto create
group for eSMTP_server is set to False.
```

Upon reception of this negative acknowledge (NACK), the eSMTP_server moves the original mail message to an error location, unless the target directory is set to a value of *NONE. Figure 32: Specifying the error target directory on page 38 shows the target folder for rejected messages.

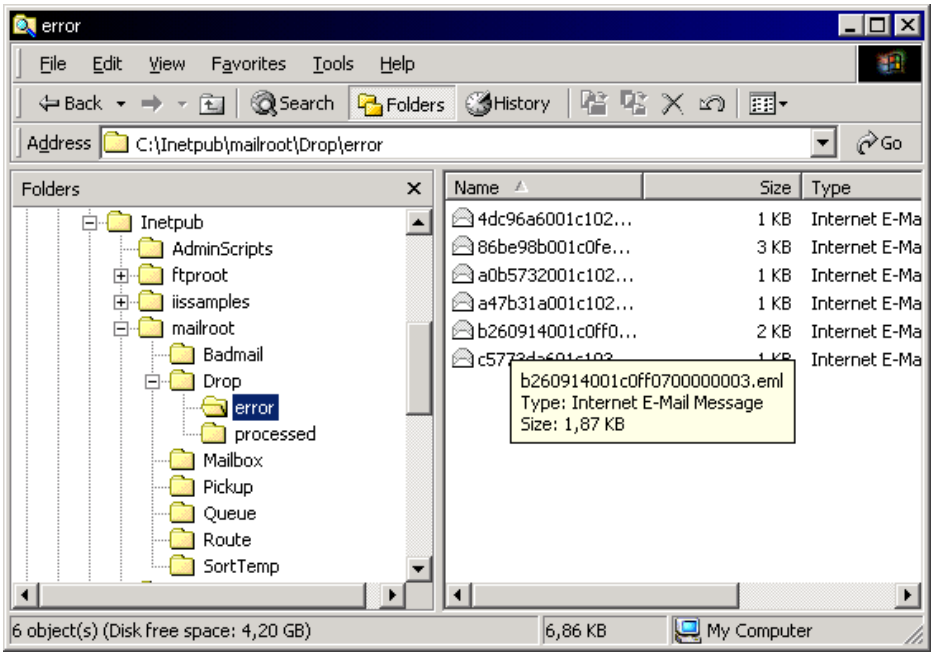


Figure 32: Specifying the error target directory

Note:

Because these rejected inbound mail messages are still available online, you can let the administrator determine the cause of the problem, and if necessary adjust the configuration settings. In many cases, the problems are related to wrong configuration, or processing of unexpected mail messages (spawn mail, hackers, and so on). After the configuration is fixed, the messages in error can be either deleted or moved back to the Dropped directory for reprocessing.

Logging

The eSMTP_server application provides logging both on-screen and on disk.

Figure 33: On-screen logging on page 39 shows the on-screen logging that can be found in the Logging tab.

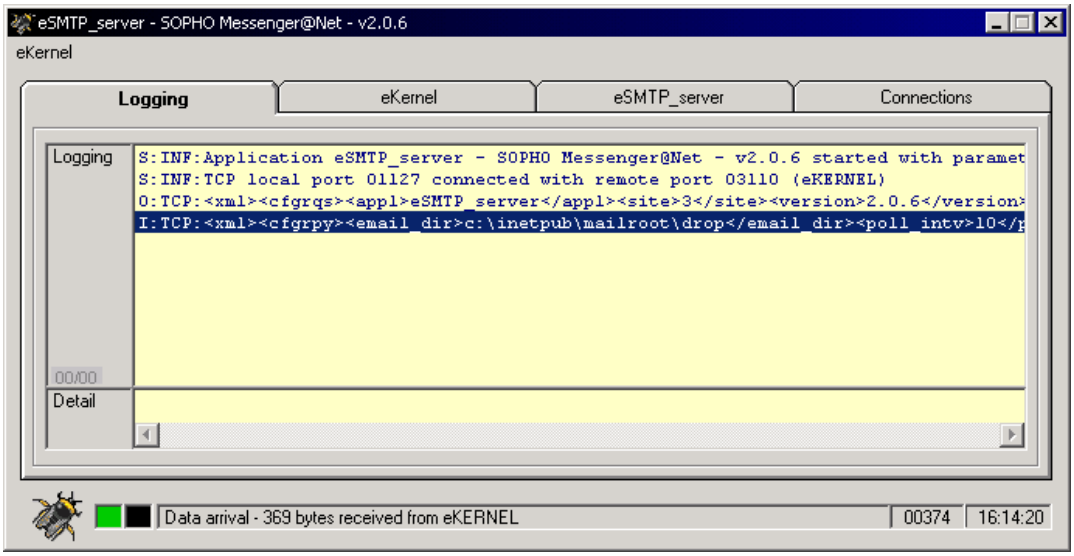


Figure 33: On-screen logging

Figure 36: Log file on disk — part 3 on page 42 shows the log file stored on disk.

```
28/10/2001 16:08:07 - S:INF:Application eSMTP_server - SOPHO Messen-
ger@Net - v2.0.6 started with parameters /Site:3 /eKernel address:*LOCAL
/eKernel port:3110 /Log drive:C
28/10/2001 16:08:08 - S:INF:TCP local port 01127 connected with remote
port 03110 (eKERNEL)
28/10/2001 16:08:08 - O:TCP:<xml><cfgrqs><appl>eSMTP_server</ap-
pl><site>3</site><version>2.0.6</version></cfgrqs></xml>
28/10/2001 16:08:08 - I:TCP:<xml><cfgrpy><email_dir>c:\inetpub\mail-
root\drop</email_dir><poll_intv>10</
poll_intv><email_dir_processed>c:\inetpub\mailroot\drop\processed</
email_dir_processed><keep_processed>5</
keep_processed><email_dir_error>c:\inetpub\mailroot\drop\error</
email_dir_error><keep_error>5</keep_error><log_path>C:\SOPHO Messen-
ger@net</log_path><log_days>1</log_days></cfgrpy></xml>

28/10/2001 16:20:02 - O:TCP:<xml><pgmsts><value>Shutdown</value></
pgmsts></xml>
28/10/2001 16:20:02 - S:INF:Application ended
28/10/2001 16:22:18 - S:INF:Application eSMTP_server - SOPHO Messen-
ger@Net - v2.0.7 started with parameters /Site:3 /eKernel address:*LOCAL
/eKernel port:3110 /Log drive:C
28/10/2001 16:22:19 - S:INF:TCP local port 01128 connected with remote
port 03110 (eKERNEL)
28/10/2001 16:22:19 - O:TCP:<xml><cfgrqs><appl>eSMTP_server</ap-
pl><site>3</site></cfgrqs></xml>
28/10/2001 16:22:20 - I:TCP:<xml><cfgrpy><email_dir>c:\inetpub\mail-
root\drop</email_dir><poll_intv>10</
poll_intv><email_dir_processed>c:\inetpub\mailroot\drop\processed</
email_dir_processed><keep_processed>5</
keep_processed><email_dir_error>c:\inetpub\mailroot\drop\error</
email_dir_error><keep_error>5</keep_error><log_path>C:\SOPHO Messen-
ger@net</log_path><log_days>1</log_days></cfgrpy></xml>

28/10/2001 16:53:21 - O:TCP:<xml><ms-
grqs><id>bc6c51d001c0ff0700000004.eml</id><from>beibsbru@ibsbe.be</
from><to>befmi@gntn1sfmi.ibsbe.be</to><subject>Subject goes here</sub-
ject></msgrqs></xml>

continued on the next page...
```

Figure 34: Log file on disk — part 1

```
28/10/2001 16:53:21 - I:TCP:<xml><ms-
grpy><id>bc6c51d001c0ff0700000004.eml</id><sts>NACK^</sts></msgrpy></
xml>

28/10/2001 17:00:10 - O:TCP:<xml><ms-
grqs><id>c5773da601c103b900000004.eml</id><from>francis.missi-
aen@1s.be</from><to>kristien.daneels@1s.be</to><subject>Reanimation</
subject></msgrqs></xml>
28/10/2001 17:00:11 - I:TCP:<xml><ms-
grpy><id>c5773da601c103b900000004.eml</id><sts>NACK^</sts></msgrpy></
xml>

28/10/2001 17:03:25 - O:TCP:<xml><ms-
grqs><id>c5773da601c103b900000004.eml</id><from>francis.missi-
aen@1s.be</from><to>kristien.daneels@1s.be</to><subject>Reanimation</
subject></msgrqs></xml>
28/10/2001 17:03:26 - I:TCP:<xml><ms-
grpy><id>c5773da601c103b900000004.eml</id><sts>NACK^</sts></msgrpy></
xml>

28/10/2001 17:07:00 - O:TCP:<xml><ms-
grqs><id>c5773da601c103b900000004.eml</id><from>francis.missi-
aen@1s.be</from><to>kristien.daneels@1s.be</to><subject>Reanimation</
subject></msgrqs></xml>
28/10/2001 17:07:00 - I:TCP:<xml><ms-
grpy><id>c5773da601c103b900000004.eml</id><sts>NACK^</sts></msgrpy></
xml>

28/10/2001 17:07:39 - O:TCP:<xml><pgmsts><value>Shutdown</value></
pgmsts></xml>
28/10/2001 17:07:39 - S:INF:Application ended
28/10/2001 17:09:06 - S:INF:Application eSMTP_server - SOPHO Messen-
ger@Net - v2.0.7 started with parameters /Site:3 /eKernel address:*LOCAL
/eKernel port:3110 /Log drive:C
28/10/2001 17:09:08 - S:INF:TCP local port 01129 connected with remote
port 03110 (eKERNEL)
28/10/2001 17:09:08 - O:TCP:<xml><cfgrqs><appl>eSMTP_server</ap-
pl><site>3</site></cfgrqs></xml>

continued on the next page...
```

Figure 35: Log file on disk — part 2

```
28/10/2001 17:09:08 - I:TCP:<xml><cfgrpy><email_dir>c:\inetpub\mail-
root\drop</email_dir><poll_intv>10</
poll_intv><email_dir_processed>c:\inetpub\mailroot\drop\processed</
email_dir_processed><keep_processed>5</
keep_processed><email_dir_error>c:\inetpub\mailroot\drop\error</
email_dir_error><keep_error>5</keep_error><log_path>C:\SOPHO Messen-
ger@net</log_path><log_days>1</log_days></cfgrpy></xml>

28/10/2001 17:09:29 - O:TCP:<xml><ms-
grqs><id>c5773da601c103b900000004.eml</id><from>francis.missi-
aen@1s.be</from><to>kristien.daneels@1s.be</to><subject>Reanimation</
subject></msgrqs></xml>
28/10/2001 17:09:29 - I:TCP:<xml><ms-
grpy><id>c5773da601c103b900000004.eml</id><sts>ACK^</sts></msgrpy></
xml>
```

Figure 36: Log file on disk — part 3

Chapter 3: Module - eSNMP

Important:

Due to the ongoing development of the DECT Messenger product suite, some modules that provide additional functionality may become available after the initial release of DECT Messenger 4.0.

The following modules are described in this document but are not available at initial General Availability.

- eFR
- eLICENSE
- eLOCATION
- eSMS
- eSNMP
- eVBVOICE

The eFR module is an add-on module and is licensed separately through the eLICENSE module. Some of the modules listed in this attention box are available only on a site-specific basis.

Architecture

The eSNMP module is able to receive SNMPv1 and SNMPv2 traps sent from an external SNMP trap sender to DECT Messenger. The eSNMP module uses an SNMP trap receiver. As a result of a received SNMP trap, an alarm is activated or deactivated on the Messenger platform.

At startup, eSNMP contacts eKERNEL to request the configuration of eKERNEL. The IP address and port of eSNMP is configured in the shortcut. When parameters are absent, the startup values are prompted, as shown in the following figure.

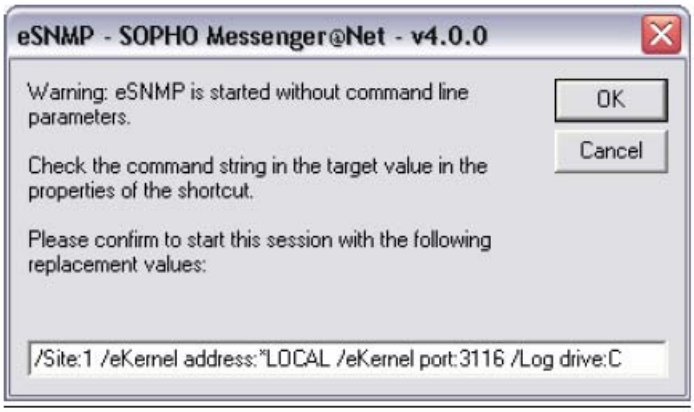


Figure 37: eSNMP parameters missing

The module eKERNEL responds to the <cfgrqs> with <cfgrpy>. This configuration is taken from the Messenger_CFG database table eSNMP. The Messenger_CFG database table contains, for example, the port number 162 that is used to receive SNMPv1 and SNMPv2 traps. See the following figure.

```
S:INF:TCP local port 01238 connected with remote port 03116 (eKERNEL)
O:TCP:<xml><cfgrqs><appl>eSNMP</appl><site>1</site></cfgrqs></xml>
I:TCP:<xml><cfgrpy><port>162</port><log_path>C:\SOPHO
Messenger@net</log_path><log_days>14</log_days></cfgrpy></xml>
```

Figure 38: Messenger_CFG database table

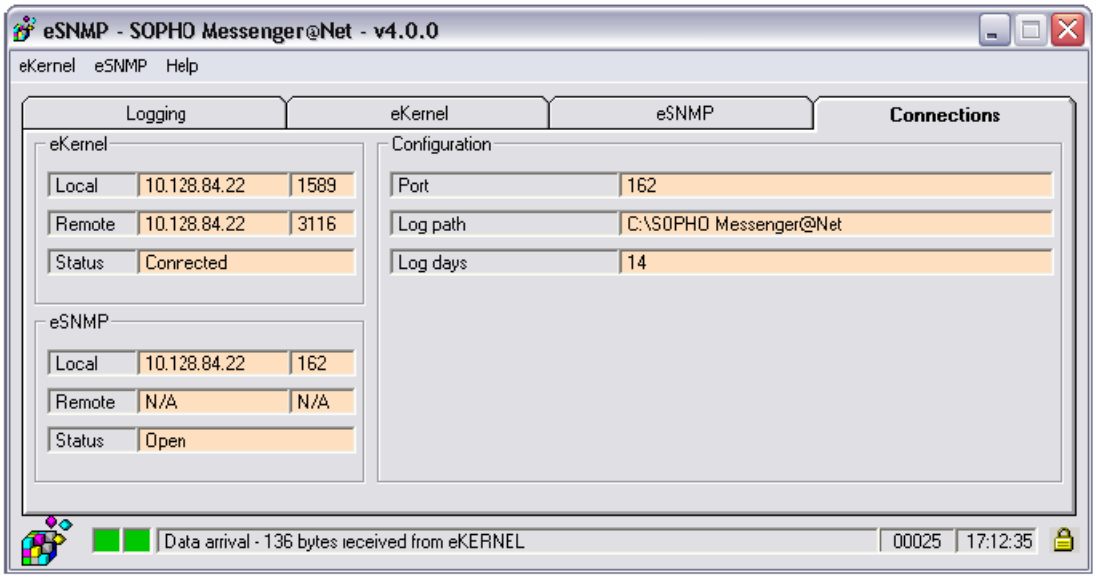


Figure 39: eSNMP connections

The SNMP traps are shown under the eSNMP tab. SNMP traps are ASN.1 BER-encoded. In the eSNMP, the received data is represented in an XML-style way to improve readability. See the following figure.

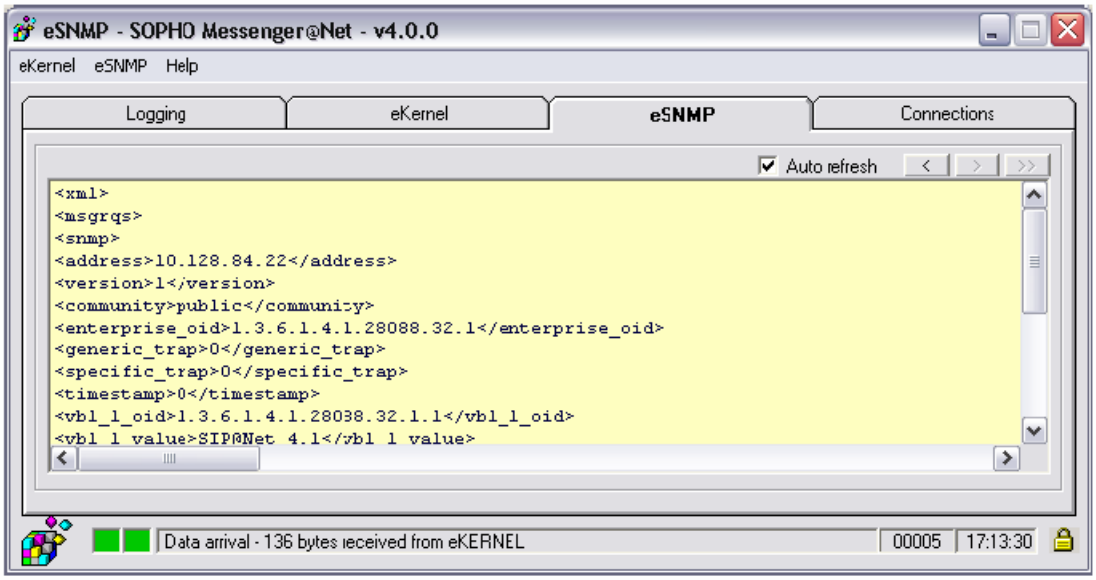


Figure 40: eSNMP traps

The eSNMP module can determine the originator of the SNMP trap. For example, in Figure 40: eSNMP traps on page 45, an SNMPv1 trap was received from 10.128.84.22. The SNMP trap also contains some “header” information, such as the following.

- community, for example public
- enterprise oid, for example 1.3.6.1.4.1.28088.32.1
- generic trap (value between 0 and 6)
- specific trap (0 or >0)
- a timestamp

Optionally, additional parameters can be received. The eSNMP module can handle up to nine additional parameters or varbind parameters. Each varbind parameter is up to 32 characters long.

When an SNMPv1 or SNMPv2 trap is received, a message is sent to eKERNEL. eKERNEL considers the eSNMP modules as an input program, and requires the typical parameters needed for generating a message. See the following figure.

```
O:TCP:<xml><msgrqs><snmp><address>127.0.0.1</address><version>1</version><community>public
</community><enterprise_oid>1.3.6.1.4.1.17338.32.1001</enterprise_oid><generic_trap>6</gen
eric_trap><specific_trap>1</specific_trap><timestamp>0</timestamp><vbl_1_oid>1.3.6.1.4.1.2
854.6.1.1.1</vbl_1_oid><vbl_1_value>2003</vbl_1_value><vbl_1_type>2</vbl_1_type><vbl_2_oid
>1.3.6.1.4.1.2854.6.1.1.2</vbl_2_oid><vbl_2_value>2</vbl_2_value><vbl_2_type>2</vbl_2_type
><vbl_3_oid>1.3.6.1.4.1.2854.6.1.1.3</vbl_3_oid><vbl_3_value>Perflib</vbl_3_value><vbl_3_t
ype>4</vbl_3_type><vbl_4_oid>1.3.6.1.4.1.2854.6.1.1.4</vbl_4_oid><vbl_4_value>GNTN1SPMI</v
bl_4_value><vbl_4_type>4</vbl_4_type><vbl_5_oid>1.3.6.1.4.1.2854.6.1.1.5</vbl_5_oid><vbl_5
_value>Not pecified</vbl_5_value><vbl_5_type>4</vbl_5_type><vbl_6_oid>1.3.6.1.4.1.2854.
6.1.1.6</vbl_6_oid><vbl_6_value>The configuration information of the performance library
"C:\WINNT\system32\w3ctrs.dll" for the   "W3SVC" service does not match the trusted
performance library information   stored in the registry. The functions in this library
will not be treated   as trusted.</vbl_6_value><vbl_6_type>4</vbl_6_type>
<vbl_7_oid>1.3.6.1.4.1.2854.6.1.1.7</vbl_7_oid><vbl_7_value>0</vbl_7_value><vbl_7_type>2</
vbl_7_type></snmp></msgrqs></xml>
```


Figure 41: Parameters used by eKERNEL to generate a message

The eKERNEL module has two configuration tables to support the eSNMP module.

The eSNMP table provides configuration items for an instance of eSNMP, and provides the information in <cfgrqs> and <crgryp> parameter exchange. The eSNMP table can also define automatic creation of trap definitions in the eSNMP_TRAPS table, and provides default values for those automatically created definitions in eSNMP_TRAPS. These parameters include action, alarm identifier, group, message text, and so on. See the following figure.

C:\SOPHO Messenger@Net\Mdb\Messenger_CFG.MDB
Table: eSNMP

Name	Type	Size
eSNMP_Site_id_n	Integer	2
eSNMP_Area_id_n	Integer	2
eSNMP_Port_str	Text	5
eSNMP_Autocreate_b	Yes/No	1
eSNMP_Default_Action_str	Text	6
eSNMP_Default_ALA_id_n	Long Integer	4
eSNMP_Default_GRP_Name_str	Text	128
eSNMP_Default_Msg_str	Text	255
eSNMP_Default_Activate_b	Yes/No	1
eSNMP_Comments_str	Text	255

Figure 42: eSNMP table

eSNMP_Site_id_n	1
eSNMP_Area_id_n	1
eSNMP_Port_str	162
eSNMP_Autocreate_b	-1
eSNMP_Default_Action_str	*SET
eSNMP_Default_ALA_id_n	1120102
eSNMP_Default_GRP_Name_str	SNMP
eSNMP_Default_Msg_str	Guarding@Net : Unknown trap [e
eSNMP_Default_Activate_b	-1
eSNMP_Comments_str	Default SNMP configuration

Figure 43: eSNMP

The other configuration table that the eKERNEL module uses to support the eSNMP module is eSNMP_TRAPS. This module defines the traps that are processed by the eSNMP module.

When the eSNMP table specifies auto-configuration, the eSNMP_TRAPS are automatically populated with definitions as new traps are received. This allows system administrators to gradually optimize configurations by updating the definitions, and associating alarm identifiers, groups and message. See the following figure.

C:\SOPHO Messenger@Net\Mdb\Messenger_CFG.MDB
Table: eSNMP_TRAPS

Name	Type	Size
eSNMPT_Site_id_n	Integer	2
eSNMPT_Area_id_n	Integer	2
eSNMPT_Address_str	Text	15
eSNMPT_Version_str	Text	1
eSNMPT_Community_str	Text	255
eSNMPT_Enterprise_OID_str	Text	255
eSNMPT_Generic_str	Text	1
eSNMPT_Specific_str	Text	255
eSNMPT_Activate_b	Yes/No	1
eSNMPT_Action_str	Text	6
eSNMPT_ALA_id_n	Long Integer	4
eSNMPT_GRP_Name_str	Text	128
eSNMPT_Message_str	Memo	-
eSNMPT_Comments_str	Text	255

Figure 44: eSNMP_TRAPS table

eSNMPT_Site_id_n	1
eSNMPT_Area_id_n	1
eSNMPT_Address_str	127.0.0.1
eSNMPT_Version_str	1
eSNMPT_Community_str	public
eSNMPT_Enterprise_OID_str	1.3.6.1.4.1.17338.32.1001
eSNMPT_Generic_str	6
eSNMPT_Specific_str	1
eSNMPT_Activate_b	-1
eSNMPT_Action_str	*SET
eSNMPT_ALA_id_n	1120102
eSNMPT_GRP_Name_str	SNMP
eSNMPT_Message_str	SNMP trap [enterprise_oid] from [address] [vbl_1_value] - [vbl_3_value]
eSNMPT_Comments_str	SOPHO 2000 IPS board failure

Figure 45: eSNMP_TRAPS

The eSNMP module is designed to handle the parameters available in SNMPv1 traps, such as community, enterprise OID, generic trap, and specific trap. These fields are considered key-fields, and allow the eSNMP module to associate.

- Action (*SET or *RESET)
- Alarm identifier (as specified in eKERNEL_ALARM table)
- Group (as specified in eKERNEL_GROUP and eKERNEL_GROUP_MEMBER)
- Message

The message can be a combination of fixed text and replacement values. In many cases, SNMP traps provide a list of variable binding parameters.

The eSNMP module supports up to nine varbind parameters. The value of those parameters can be embedded in the resulting message. The following figure shows a sample of such a definition.

```
SNMP trap [enterprise_oid] from [address] [vbl_1_value] - [vbl_3_value]
```

Figure 46: Message with varbind parameters

Supported replacement values are listed in the following figure.

```
[address]
[version]
[community]
[enterprise_oid]
[generic_trap]
[specific_trap]
[timestamp]
[vbl_1_oid] to [vbl_9_oid]
[vbl_1_value] to [vbl_9_value]
```

Figure 47: Supported replacement values

When the SNMPv2 syntax is used, the parameters enterprise OID, generic trap, specific trap, and timestamp are not available. These missing parameters are provided as value 0 so that the existing implementation can also handle SNMPv2 traps. You must specify 0 in the corresponding fields to preserve the unique key.

The definitions are related to site and area. This means every instance of eSNMP can have an appropriate definition. The IP address of the sender is included in the key.

When no GenericTrap field is available, the value 0 is used. The use of the value 0 allows a flexible filtering of alarms; for example, ignore one or more traps by not specifying the corresponding alarm and, for example, assigning priorities.

Note that generic traps are reserved values in SNMP protocol, as shown in the following figure.

```
0 cold start
1 warm start
2 link down
3 link up
4 authentication failure
5 egp neighbor loss
6 enterprise specific
```

Figure 48: Generic traps

Send SNMP Message for Win32

A sample program of Send SNMP Message is available on the DECT Messenger CD in the directory **09-Add-ons\Send SNMP Message**.

When you install the program, it resides in the directory **C:\Program Files\Send SNMP Message\Exe** and is called **Send SNMP Message**.

When the program is launched, a window appears and defaults to IP address 127.0.0.1 and port 162. If Send SNMP Message is installed on the same PC as the eSNMP module, the default value should not be changed. If Send SNMP Message is installed on a different PC, adjust the IP address and port to match the PC where eSNMP runs.

The window contains several input-capable fields that allow you to configure every parameter. When you click the **Send**button, an SNMPv1 trap is sent to eSNMP.

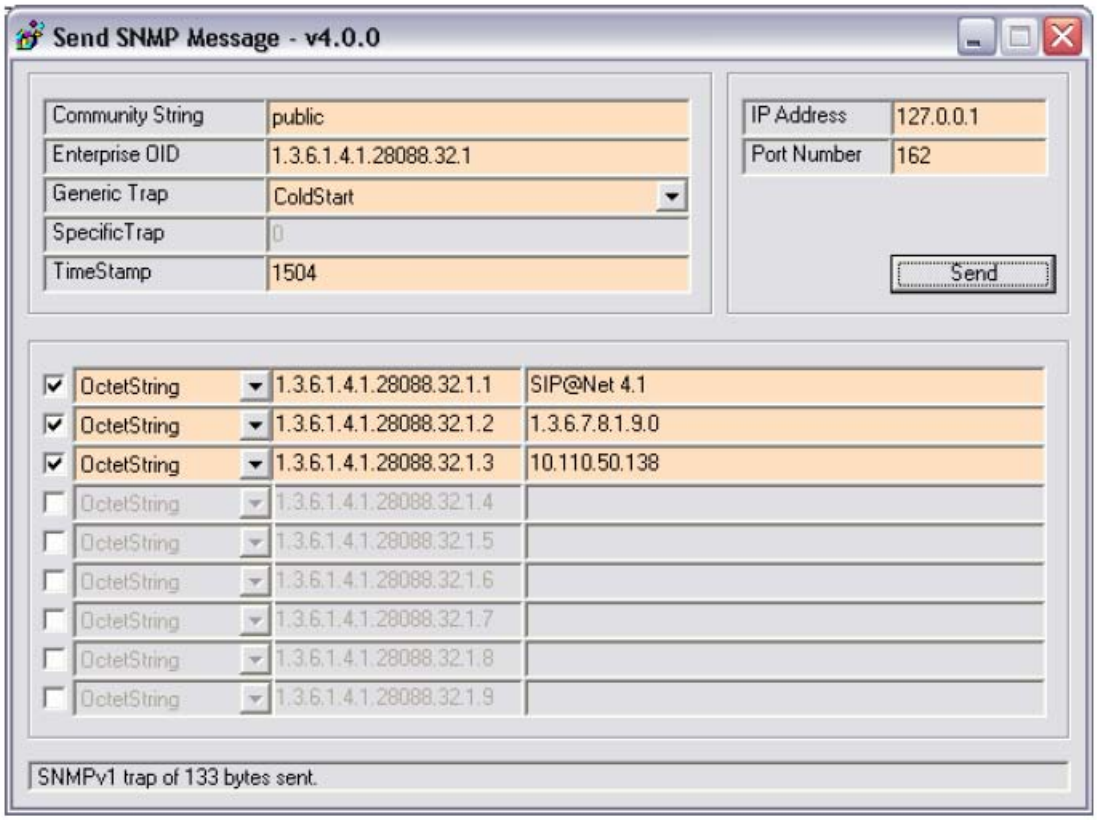


Figure 49: Send SNMP Message

The Send SNMP Message program is provided on an “as-is” basis.

Send SNMP Message for Web

The Web Administrator provides a web-based interface to send SNMPv1 traps from the Web Administrator to the module eSNMP.

The implementation is based upon PHP script and hosted in the Apache HTTP Server on the Messenger platform.

Since sending SNMPv1 traps is reserved for system administrators, the SNMP trap is reserved for Web Administrator users with a security level equal to administrator. Security level equal to administrator refers to users in the eWEB_AUTH table defined with USERA_Sec_level of 50 or above.

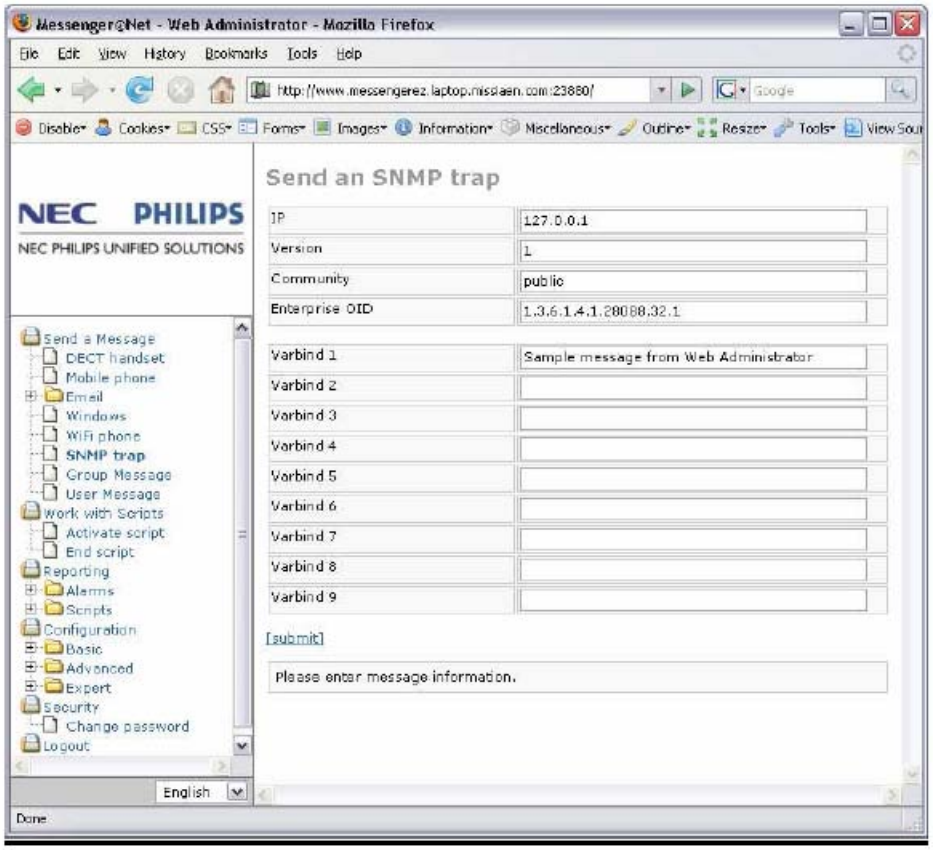


Figure 50: Send an SNMP trap

Send SNMP Message for iSeries

A similar program called Send SNMP Message is available for the IBM eServer iSeries platform. This program is also referred to as IBM AS/400 or IBM i5.

This Send SNMP Message for iSeries features a command line interface, and is easily embedded in existing legacy code written on CLP, RPG, RPG/LE an so on.

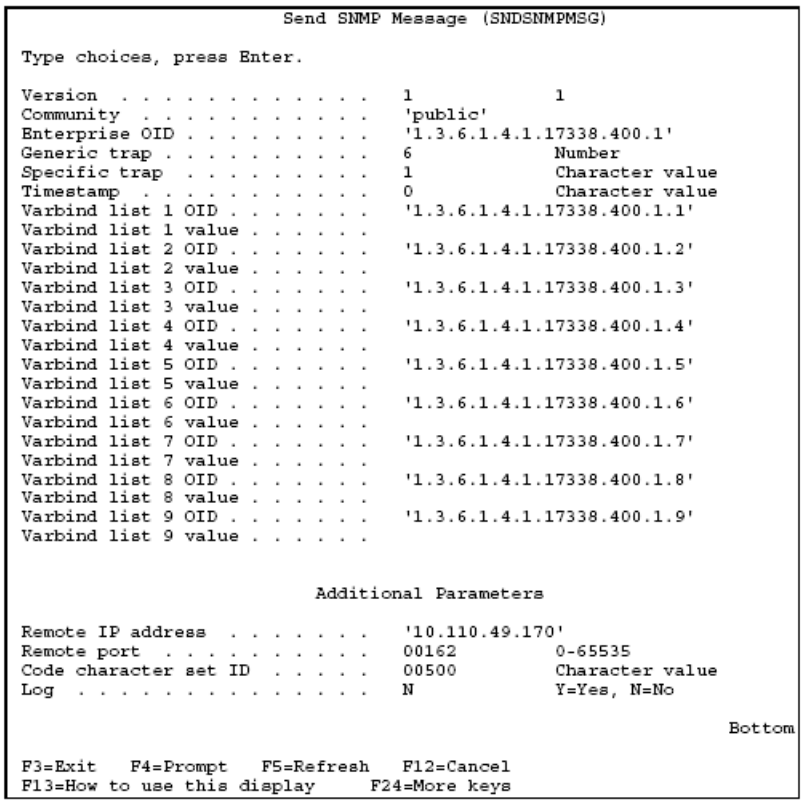


Figure 51: Send SNMP message (SNDSNMPMSG)

Contact Avaya product support to obtain more details.

Chapter 4: Module - eTM

The module eTM is an application that is represented as a small icon in the system tray on the bottom right-hand side of the desktop. This tray is usually populated with other applications, as shown in Figure 52: System Tray on page 53, where the eTM icon is shown to the immediate left of the clock.

Figure 52: System Tray

When the mouse is moved over the icon in the system tray, right-click to open the menu shown in Figure 53: Open Task Manager on page 53.

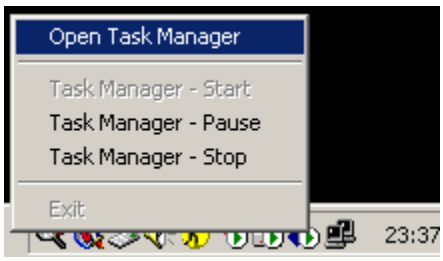


Figure 53: Open Task Manager

The menu option **Open Task Manager** restores the main menu, and can be opened to monitor the tasks in detail. This menu also provides options to **Start**, **Stop**, or **Pause** processing. Use the **Exit** menu option to terminate the eTM module and all associated tasks.

Select the **Open Task Manager** menu option in the pop-up menu, to open the Task Manager, as shown in Figure 54: eTM Task Manager on page 54.

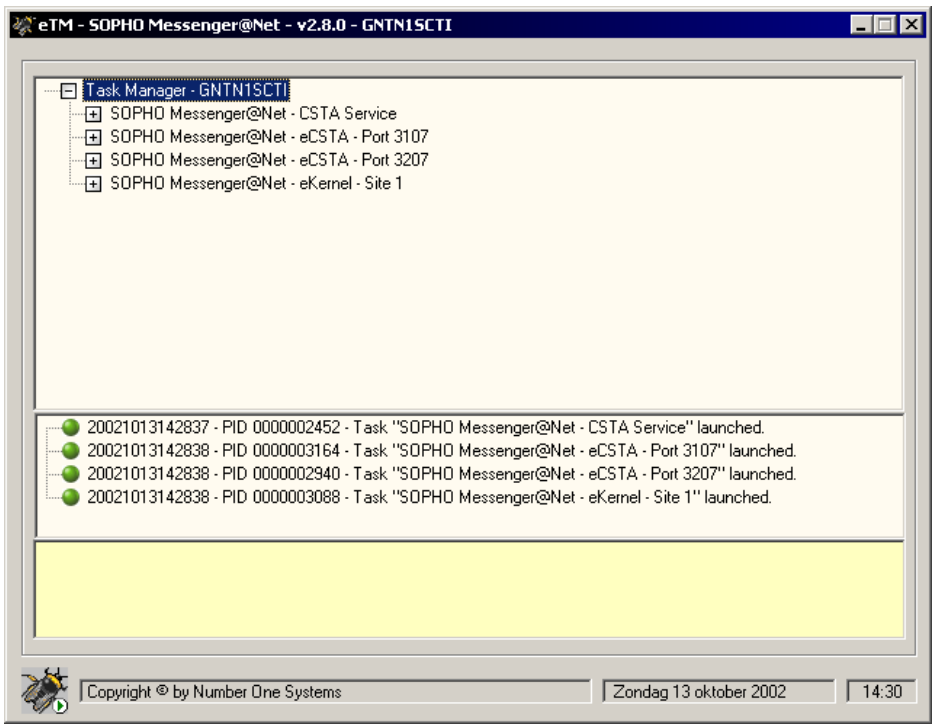


Figure 54: eTM Task Manager

Note:

The window contents vary according to your configuration settings.

The window is composed of the following sections:

- The upper section presents a tree-view of the environment, and contains a hierarchical overview of all configured tasks. Every task has the following keywords and values:
  - The keyword PID denotes the process identifier of the task. This identifier is formatted as a 10-digit numeric value. The PID is the value that is also shown when the system supplied Task Manager of Microsoft is used to represent the processes. A special value 0000000000 is shown when the task is not running.
  - The keyword Window style denotes the style of the window of the task. Supported values are described in Table 1: Supported window styles on page 54.

Table 1: Supported window styles

Value	Description
0	Window is hidden and focus is passed to the hidden window.
1	Window has focus and is restored to its original size and position.
2	Window is displayed as an icon with focus.
3	Window is maximized with focus.

Value	Description
4	Window is restored to its most recent size and position. The currently active window remains active.
6	Window is displayed as an icon. The currently active window remains active.

- The keyword Shortcut denotes the command line parameter that is used to launch the process.

- The second section shows a log of the changes in the state of the tasks.
- The third section shows some additional logging information, and is updated when, for instance, a task is terminated from within the eTM application.
- The bottom section shows on the left a small icon that denotes the current state of the eTM. This application can be started, paused or stopped.

The eTM is launched by means of the following command:

Table 2: Launch eTM Command

```
C:\SOPHO Messenger@Net\Exe\eTM.exe
```

In most cases there is only one environment configured, and the eTM uses this default configuration. When there is more than one environment configured, a selection window opens that allows you to specify the environment that must be started, as shown in Figure 55: Specify the eTM environment (when more than one is configured) on page 55.

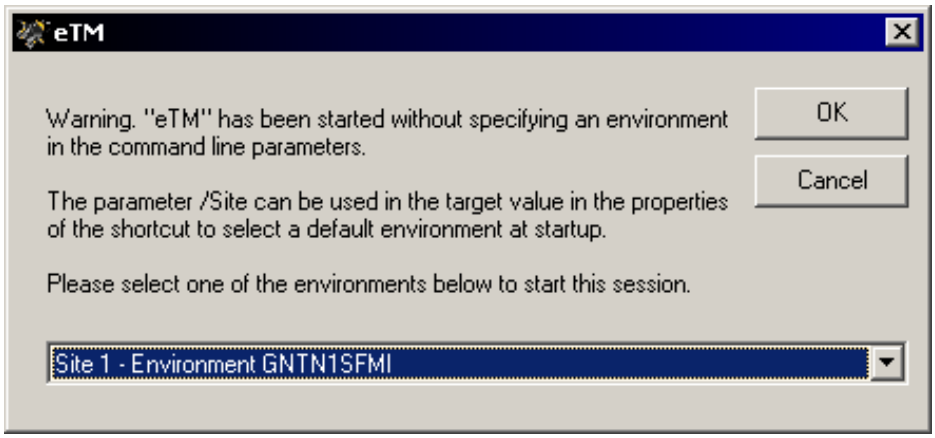


Figure 55: Specify the eTM environment (when more than one is configured)

If there is more than one environment configured, you can choose to automatically select a startup environment. This can be accomplished by extending the launch command with the keyword /Site:xxxxx, where xxxxx is to be replaced by the configured environment name. For example, the following command automatically launches the eTM for environment GNTN1SFMI.

Table 3: Launch ETM command

```
C:\SOPHO Messenger@Net\Exe\eTM.exe /Site:GNTN1SFMI
```

The Windows Registry Editor (regedit or regedt32) can be used to maintain the configuration of the eTM.

Figure 56: Sample eTM configuration registry entry on page 56 shows a sample configuration, as represented in the system registry as a result of the configuration process.

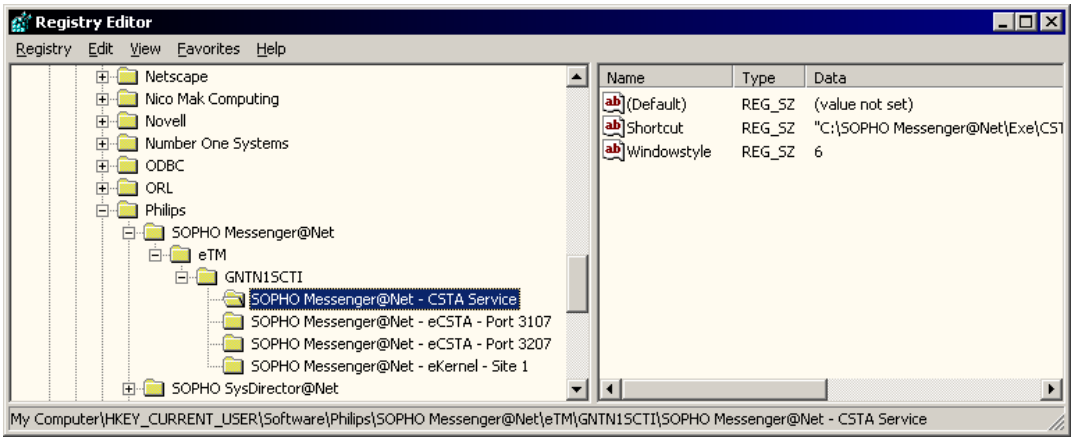


Figure 56: Sample eTM configuration registry entry

Figure 57: Sample eTM configuration on page 56 shows a sample configuration for the eTM module that defines the following:

- One instance of CSTA_Service.exe
- One instance of eKERNEL.exe

The text file represented in Figure 57: Sample eTM configuration on page 56 has a filename with extension .reg and can be created with a text editor (for example, Notepad).

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\GNTN1SCTI]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messen-
ger@Net\eTM\GNTN1SCTI\SOPHO Messenger@Net - CSTA Service]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\CSTA_service.exe\""
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messen-
ger@Net\eTM\GNTN1SCTI\SOPHO Messenger@Net - eKernel - Site 1]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eKERNEL.exe\" /Site:1"
"Windowstyle"="6"
```

Figure 57: Sample eTM configuration

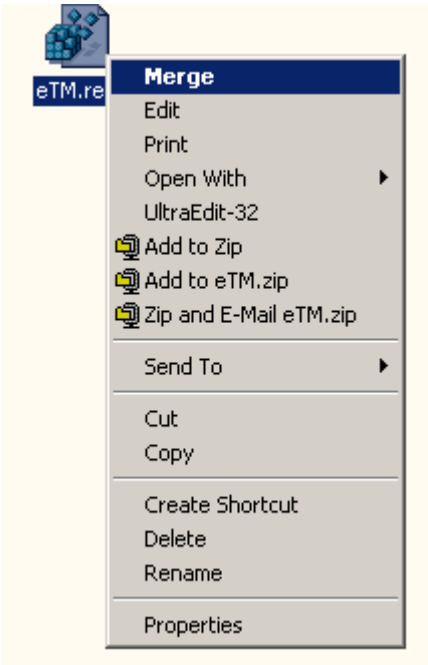
Files with .reg extension can be merged to the registry.

Merging .reg files

1. Select the **Merge**command.

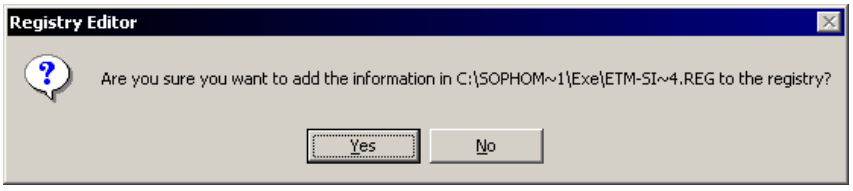
   In Windows Explorer:

   - Locate the file you wish to merge.
   - Right-click the file.
   - Choose **Merge** from the pop-up menu, as follows:



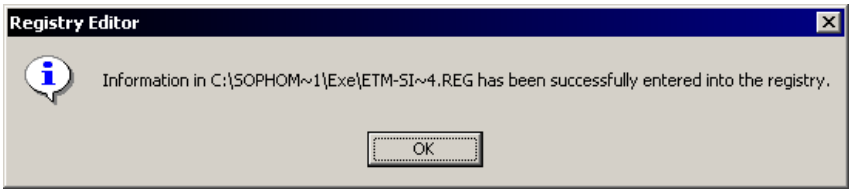
2. Confirm that you wish to merge the registry.

   Click **Yes** to continue.



3. Confirm completion of the registry merge.

   Click **OK**.



The command RegEdit or RegEdt32 can be used to verify the configuration, or to apply changes to an existing configuration.

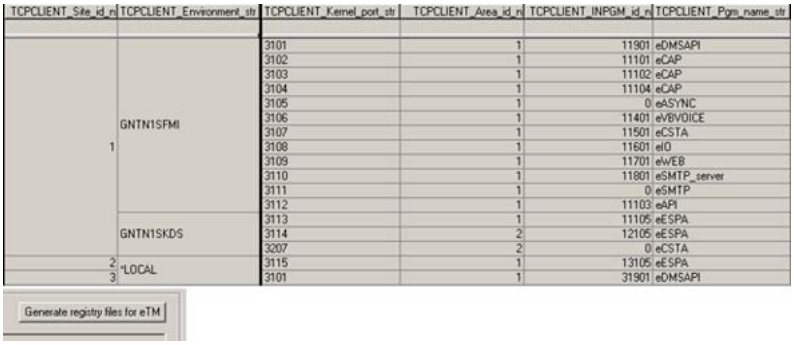
A future release of DECT Messenger will provide automatic procedures for configuring the Task Manager from the Configurator module.

In Release 4, the eGRID module features a command button **Generate registry files for eTM**. Click this button to read the eKERNEL_TCPCLIENT table and automatically generate the required shortcuts for each site and environment, as shown in Generate shortcuts on page 58.

Generate shortcuts

1. Use eGRID to generate registry files for eTM.

   Launch eGRID and click **Generate registry files for eTM**.



TCPCLIENT_Site_id_n	TCPCLIENT_Environment_str	TCPCLIENT_Kernel_port_str	TCPCLIENT_Area_id_n	TCPCLIENT_INPGM_id_n	TCPCLIENT_Pgm_name_str
1	GNTN1SFMI	3101	1	11901	eDMSAPI
		3102	1	11101	eCAP
		3103	1	11102	eCAP
		3104	1	11104	eCAP
		3105	1	0	eASYNC
		3106	1	11401	eVBVOICE
		3107	1	11501	eCSTA
		3108	1	11601	eIO
		3109	1	11701	eWEB
		3110	1	11801	eSMTP_server
		3111	1	0	eSMTP
		3112	1	11103	eAPI
	GNTN1SKDS	3113	1	11105	eESPA
		3114	2	12105	eESPA
		3207	2	0	eCSTA
2	*LOCAL	3115	1	13105	eESPA
3		3101	1	31901	eDMSAPI

2. Review the information provided, and acknowledge completion of the process.

   Click **OK** to continue.

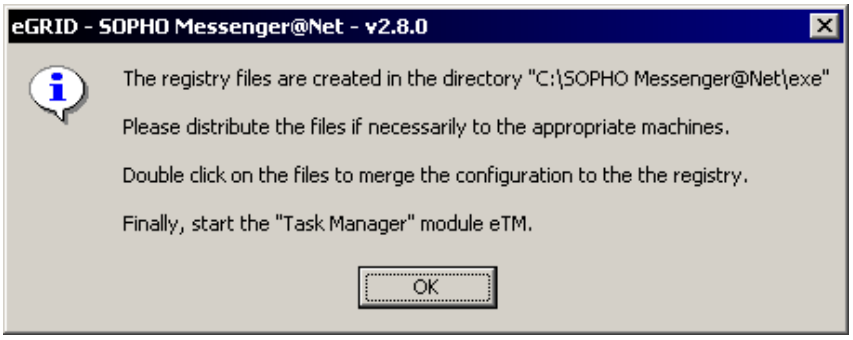


Figure 58: Example of configuration of four environments on page 59 shows a configuration of four environments.

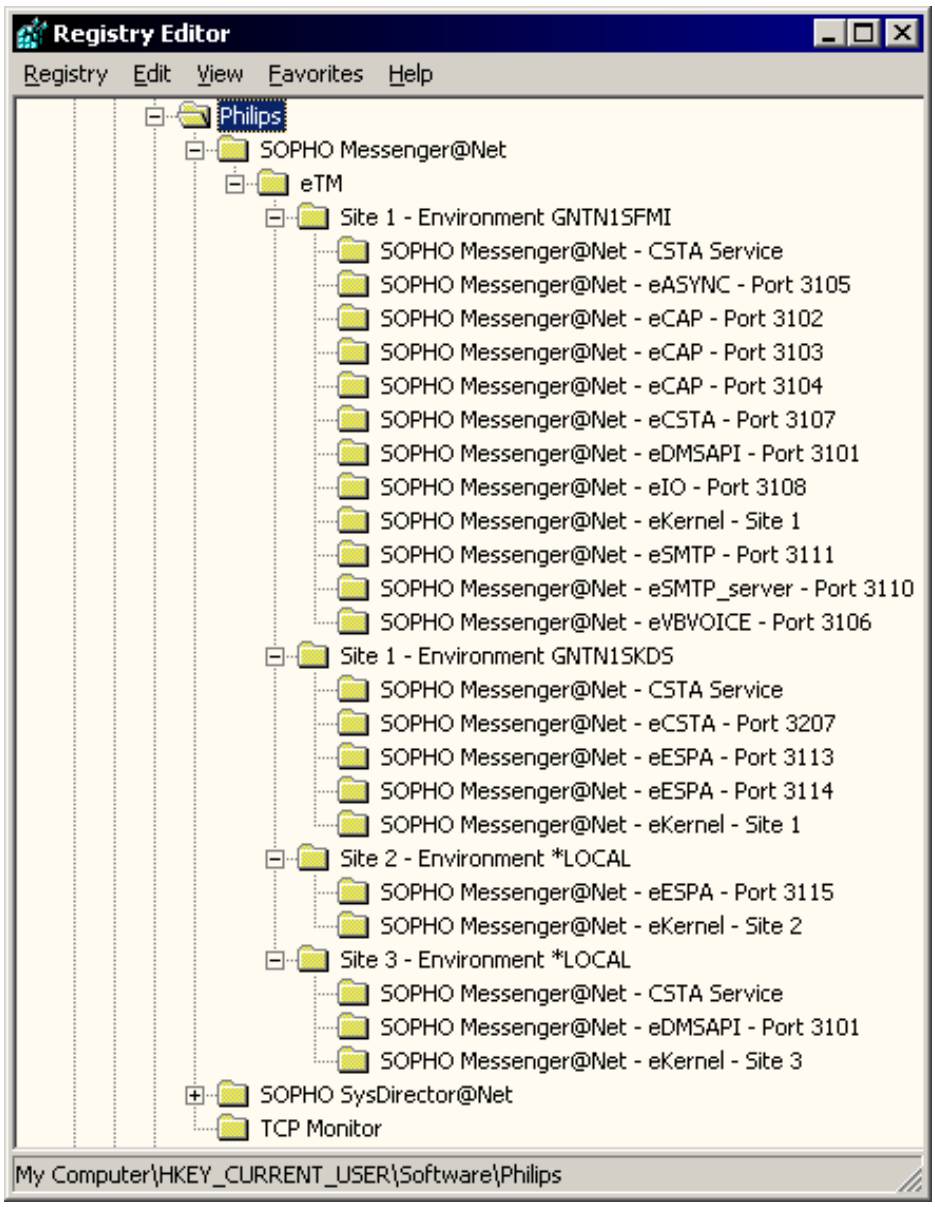


Figure 58: Example of configuration of four environments

- Site 1
  - Environment GNTN1SFMI
  - Environment GNTN1SKDS
- Site 2
  - Environment *LOCAL
- Site 3
  - Environment *LOCAL

The first two environments reside on site 1, the other environments reside on other sites. In this example, the modules of site 1 are distributed across two environments (two separate PC platforms). The PC with

environment GNTN1SFMI contains a full-featured installation with one or more instances of each module; the second environment GNTN1SKDS contains a subset of the modules only. Figure 59: eTM - Site 1 - Environment GNTN1SFMI.reg on page 60 shows the registry file corresponding to the foregoing example.

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
Environment GNTN1SFMI]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
Environment GNTN1SFMI\SOPHO Messenger@Net - eKernel - Site 1]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eKERNEL.exe\" /Site:1"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
Environment GNTN1SFMI\SOPHO Messenger@Net - eSMTP_server - Port 3110]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eSMTP_server.exe\" /Site:1
/eKernel address:GNTN1SFMI /eKernel port:3110 /Log drive:C"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
Environment GNTN1SFMI\SOPHO Messenger@Net - eCAP - Port 3102]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eCAP.exe\" /Site:1 /eKernel
address:GNTN1SFMI /eKernel port:3102 /Log drive:C"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
Environment GNTN1SFMI\SOPHO Messenger@Net - eCAP - Port 3103]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eCAP.exe\" /Site:1 /eKernel
address:GNTN1SFMI /eKernel port:3103 /Log drive:C"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
Environment GNTN1SFMI\SOPHO Messenger@Net - eCAP - Port 3104]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eCAP.exe\" /Site:1 /eKernel
address:GNTN1SFMI /eKernel port:3104 /Log drive:C"
"Windowstyle"="6"
```

continued on next page...

```
[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
Environment GNTN1SFMI\SOPHO Messenger@Net - eASYNC - Port 3105]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eASYNC.exe\" /Site:1 /eKer-
nel address:GNTN1SFMI /eKernel port:3105 /Log drive:C"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
Environment GNTN1SFMI\SOPHO Messenger@Net - CSTA Service]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\CSTA_Service.exe\""
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
Environment GNTN1SFMI\SOPHO Messenger@Net - eDMSAPI - Port 3101]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eDMSAPI.exe\" /Site:1 /eK-
ernel address:GNTN1SFMI /eKernel port:3101 /Log drive:C"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
Environment GNTN1SFMI\SOPHO Messenger@Net - eSMTP - Port 3111]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eSMTP.exe\" /Site:1 /eKer-
nel address:GNTN1SFMI /eKernel port:3111 /Log drive:C"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
Environment GNTN1SFMI\SOPHO Messenger@Net - eIO - Port 3108]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eIO.exe\" /Site:1 /eKernel
address:GNTN1SFMI /eKernel port:3108 /Log drive:C"
"Windowstyle"="6"
```

Figure 59: eTM - Site 1 - Environment GNTN1SFMI.reg

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 - Environment
GNTN1SKDS]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 - Environment
GNTN1SKDS\SOPHO Messenger@Net - eKernel - Site 1]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eKERNEL.exe\" /Site:1"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 - Environment
GNTN1SKDS\SOPHO Messenger@Net - eESPA - Port 3113]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eESPA.exe\" /Site:1 /eKernel address:GNTN1SKDS /
eKernel port:3113 /Log drive:C"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 - Environment
GNTN1SKDS\SOPHO Messenger@Net - eESPA - Port 3114]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eESPA.exe\" /Site:1 /eKernel address:GNTN1SKDS /
eKernel port:3114 /Log drive:C"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 - Environment
GNTN1SKDS\SOPHO Messenger@Net - CSTA Service]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\CSTA_Service.exe\""
"Windowstyle"="6"
```

Figure 60: eTM - Site 1 - Environment GNTN1SKDS.reg

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 2 - Environment *LO-
CAL]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 2 - Environment *LO-
CAL\SOPHO Messenger@Net - eKernel - Site 2]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eKERNEL.exe\" /Site:2"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 2 - Environment *LO-
CAL\SOPHO Messenger@Net - eESPA - Port 3115]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eESPA.exe\" /Site:2 /eKernel address:*LOCAL /eKer-
nel port:3115 /Log drive:C"
"Windowstyle"="6"
```

Figure 61: eTM - Site 2 - Environment LOCAL.reg

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 3 - Environment *LO-
CAL]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 3 - Environment *LO-
CAL\SOPHO Messenger@Net - eKernel - Site 3]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eKERNEL.exe\" /Site:3"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 3 - Environment *LO-
CAL\SOPHO Messenger@Net - eDMSAPI - Port 3101]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eDMSAPI.exe\" /Site:3 /eKernel address:*LOCAL /eK-
ernel port:3101 /Log drive:C"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 3 - Environment *LO-
CAL\SOPHO Messenger@Net - CSTA Service]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\CSTA_Service.exe\""
"Windowstyle"="6"
```

Figure 62: eTM - Site 3 - Environment LOCAL.reg

At startup, the eTM retrieves the configuration, and launches all tasks that are defined in the environment according to the configuration. As shown in the example in Figure 62: eTM - Site 3 - Environment LOCAL.reg on page 63, the environment GNTN1SCTI launches the DECT Messenger modules CSTA Server, and the module eKERNEL.

When a task is successfully launched, the logging section features a green icon indicating a normal condition, as shown in Figure 63: Green icon indicates Normal Condition on page 63.

Figure 63: Green icon indicates Normal Condition

When the tasks is ended — for example, by means of the Alt-F4 keystroke combination — the eTM detects this and relaunches the missing task. This is indicated in the log as shown in Figure 64: Red icon indicates a task that is no longer running on page 63.

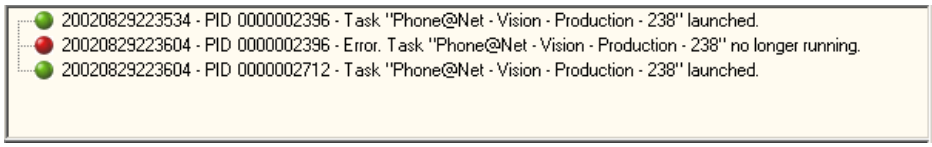


Figure 64: Red icon indicates a task that is no longer running

eTM checks every five seconds to ensure that each task is still running. When the eTM is paused or stopped, the routine that verifies and restarts the process is temporarily interrupted.

This interruption usually occurs during maintenance of one of more of the programs that are guarded by the eTM. Such a temporary condition is shown in the log as illustrated in Figure 65: Yellow icon indicates a task that is paused on page 64.

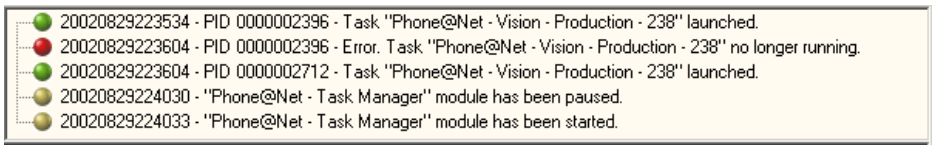


Figure 65: Yellow icon indicates a task that is paused

A system administrator can also terminate a task from within the eTM_HA environment using a **Terminate process** API-call.

Note:

Using the **Terminate process** API-call can cause data loss, as this does not provide any graceful cleanup or shutdown of the associated program.

To terminate a process, use the menu **Kill task** option, as shown in Figure 66: Kill Task on page 64. The **Kill task** option is available only when the tree-view is expanded and the mouse is right-clicked on the PID:xxxxxxxxxx line.

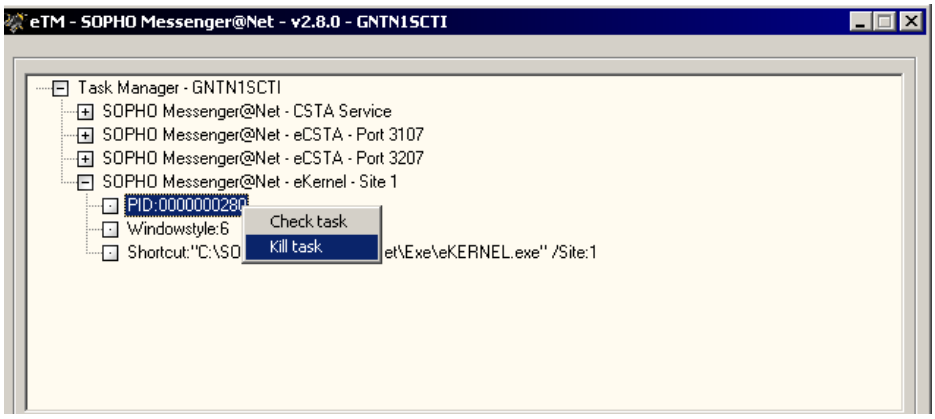


Figure 66: Kill Task

When **Kill task** is clicked, the running task is terminated, as shown in Figure 67: A task is terminated on page 65.

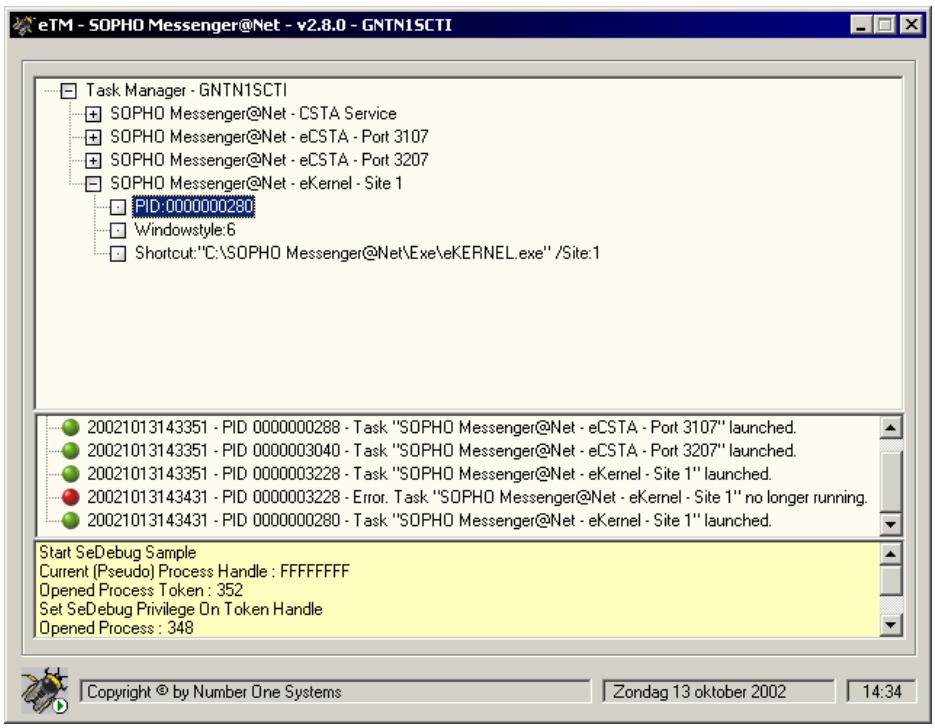


Figure 67: A task is terminated

Note:

When the eTM is running, the system relaunches the terminated tasks within 10 seconds.

When the eTM form is closed through the control box on the right top of the form, the application does not shut down, but is instead minimized to an icon in the system tray. This function is designed to prevent the user from accidentally closing the eTM and associated tasks. This approach is similar to monitoring applications of other vendors, such as the Apache Monitor or the SQL Server Service Manager.

Shutting down eTM_HA

The eTM can be shut down by opening the pop-up menu shown in Figure 53: Open Task Manager on page 53, and choosing the **Exit** menu option.

Important:

Avaya recommends that you close applications using shut down, exit or close options in the applications themselves, to ensure a clean shutdown. This helps to protect volatile data, properly close down serial and sockets communications, free resources, clean up garbage, and so on. To stop processes gracefully, follow the steps described in Shutting down eTM on page 66.

Shutting down eTM

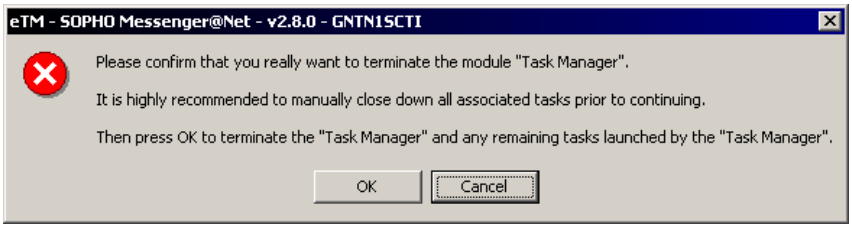
1. Open the eTM_HA pop-up menu.

   Right-click the eTM_HA icon in the system tray.

2. Stop the eTM_HA.

   - Choose the menu item **Task Manager - Stop**.
   - Choose the menu item **Exit**.

   The following confirmation prompt is shown; do not click **OK** or **Cancel** yet:



   Note:

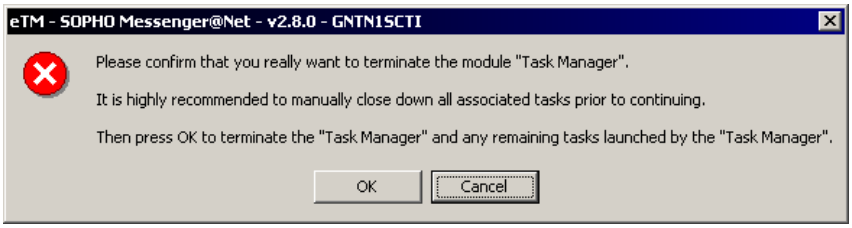
   The application also responds to a system Log off or Shut down event.

3. Shut down the applications.

   Close down all programs using the program specific instructions. In most cases this means closing the main form of each application by clicking the Close box on the top right of each form. However, some applications require specific shutdown procedures.

4. Confirm the eTM termination warning dialog box.

   Click **OK**.



   Because all associated tasks were already manually ended gracefully, no more processing is involved.

   Any associated tasks still running are terminated through a Terminate process API-call for each task that is launched from within the eTM and finally shuts down the eTM module too.

Chapter 5: Module - eTM_HA

Important:

Setting up the eTM_HA module in a networked environment is a complex task, and requires training to set up, maintain, and use in the DECT Messengerenvironment. Read the following documentation closely, and refer to the training session on eTM_HA for more details.

Overview

The module eTM_HA is the high-availability implementation of the eTM module. If you wish to migrate your system from eTM to eTM_HA, you must update the system registry.

The module eTM_HA is an application that is represented as a small icon in the system tray on the bottom right-hand side of the desktop. This tray is usually populated with other applications, as shown in Figure 68: Windows System Tray on page 67, where the eTM_HA icon is shown to the immediate left of the clock.

Figure 68: Windows System Tray

Move the mouse over the icon in the system tray, then right-click to open the menu shown in Figure 69: Open Task Manager on page 67.

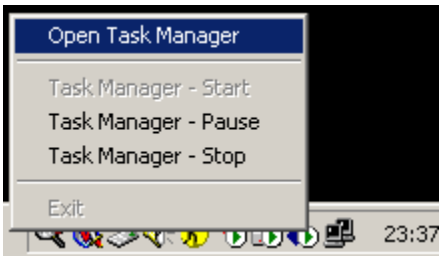


Figure 69: Open Task Manager

The menu option **Open Task Manager** restores the main menu, and can be opened to monitor the tasks in detail. This menu also provides options to **Start**, **Stop**, or **Pause** processing. Use the **Exit** menu option to terminate the eTM_HA module and all associated tasks.

When the **Open Task Manager** menu option of the pop-up menu is selected, a window similar to Figure 70: eTM-HA Task Manager - Overview tab on page 68 opens. The **Overview** tab shows the configuration, which is fetched from the registry.

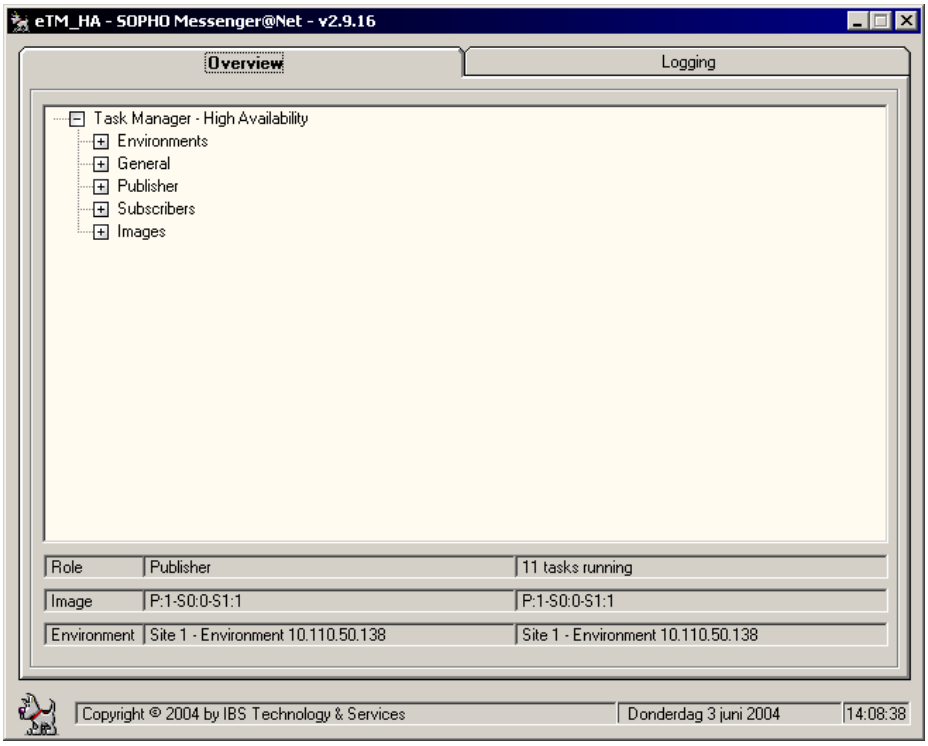


Figure 70: eTM-HA Task Manager - Overview tab

The **Logging** tab provides data as shown in Figure 71: eTM-HA Logging tab on page 69.

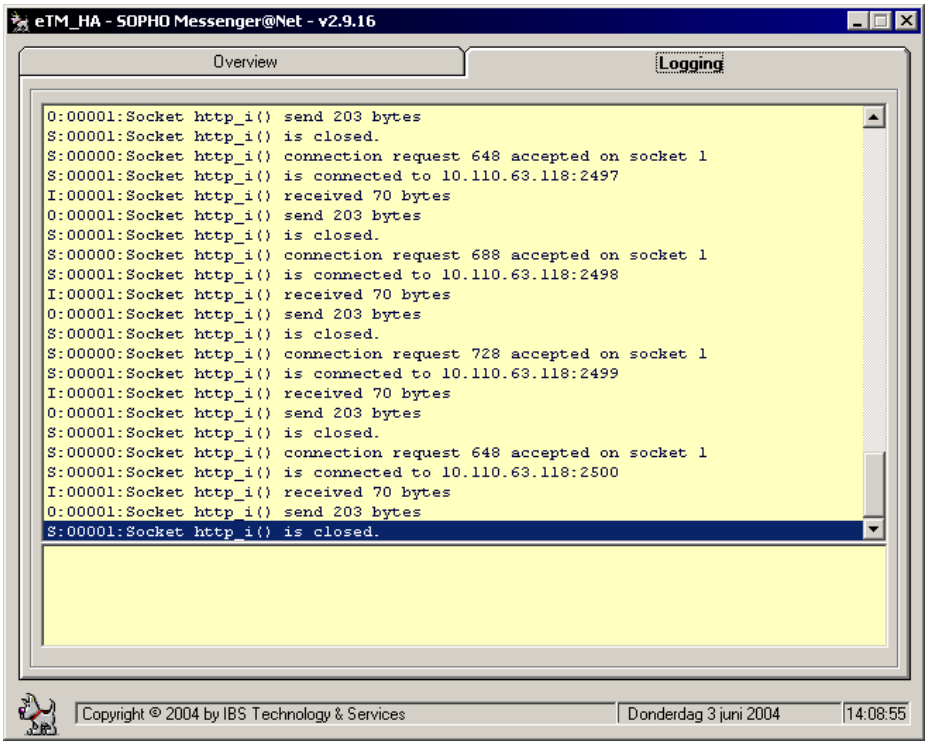


Figure 71: eTM-HA Logging tab

Note:

The information shown in Figure 69: Open Task Manager on page 67 is intended as an example. The exact information for your system differs according to your configuration settings.

Publisher and Subscriber

A typical eTM_HA environment involves one system configured be Publisher, and one or more system configured as Subscribers. Although eTM_HA can run stand-alone (just one publisher), there is no value in activating an eTM_HA when there are no Subscribers. If there are no Subscribers, use eTM instead of eTM_HA.

In the Publisher and Subscriber model, the Publisher is the site where the Messenger_CFG configuration database is centralized. This is often called the main site. All configuration must reside on this centralized database only, so eCONFIG maintenance and eKERNEL must all reside on this same site.

The eTM_HA software can also be installed on distributed systems, intended to launch tasks on the distributed system. These systems launch, for instance, eCAP and eDMSAPI modules, all of them referring to the central eKERNEL residing on the Publisher site.

The eTM_HA software must be installed on both the Publisher and the Subscriber site. Based upon configuration settings in the registry, the instance behaves as Publisher or as Subscriber.

The following functionality is available:

- eTM functionality
  - Launch tasks associated with an environment
  - Keep track of running processes of an environment
  - Restart tasks that are missing
- eTM_HA specific functionality on Publisher
  - TCP server, listening on an administration port (default 7000)
  - Handling KeepAlive requests from Subscriber
  - Handling GetImage requests from Subscriber
  - Keeping track of state of Publisher and Subscriber
  - Changing the environment depending on the Publisher and Subscriber states
- eTM_HA specific functionality on Subscriber
  - TCP client, connecting to Publisher administration port
  - Sending KeepAlive requests to Publisher
  - Sending GetImage requests to Publisher
  - Keeping track of state of Publisher and Subscriber
  - Changing the environment depending on the Publisher and Subscriber states

During a change of environment, all running tasks of a previous environment are ended, and new tasks of the new environment are launched. During such an event, the Subscriber applies the last database image received from the Publisher and optionally applies changes defined in an SQL Script.

Registry settings eTM

The configuration of environments and tasks is stored in the following section:

```
[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM]
```

This section contains definitions of environments and tasks, as described in the documentation of module eTM, Module - eTM on page 53. These settings can be entered manually of can be generated by the eCONFIG or eGRID modules.

Important:

If the environment names for eTM_HA are not defined with a name containing the local IP address, rename the registry structure generated by eGRID or eCONFIG, so that the IP address is available in the name.

The eTM structure can contain one or multiple environments. If you launch the eTM_HA.exe without additional parameters, the program analyzes the available environments of the registry, and prompts for an initial environment at startup.

Figure 72: Example of two environments on page 71 shows an example, with two environments defined. One environment is called Site 1 – Environment 10.110.50.138. The other is called Site 1 – Environment 10.110050.138 (backup).

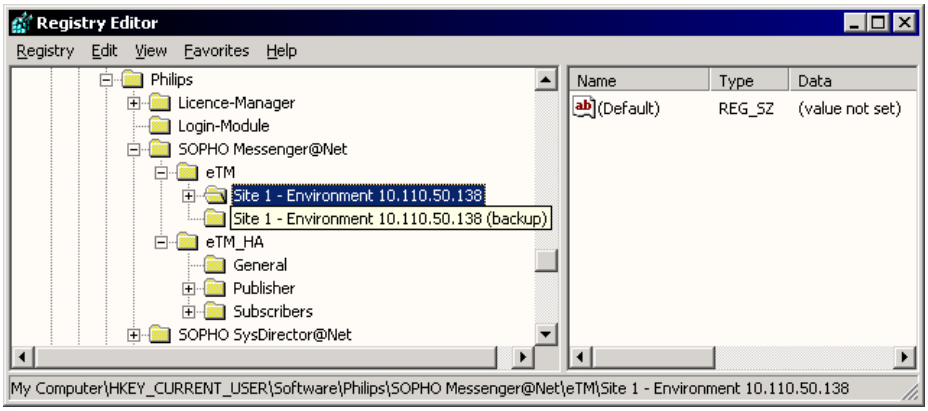


Figure 72: Example of two environments

If eTM_HA.exe is launched without additional parameters on a system with local IP address 10.110.50.138; then a prompt appears as follows:

```
"C:\SOPHO Messenger@Net\Exe\eTM_HA.exe"
```

```
"C:\SOPHO Messenger@Net\Exe\eTM_HA.exe"
```

Figure 73: Selecting an environment when more than one is defined

Note:

Because the objective of this module is to provide high availability, Avaya recommends that you suppress this prompt. This can be accomplished by adding a parameter on the command line of the shortcut specifying the initial environment to select. This is performed by means of the optional keyword **/Environment**.

Create a shortcut for eTM_HA in the startup group specifying the initial environment, as follows:

```
"C:\SOPHO Messenger@Net\Exe\eTM_HA.exe" / Environment:Site 1 – Environment 10.110.50.138
```

Note:

In eTM.exe a similar function existed, but the keyword was called **/Site**. In eTM_HA the keyword is renamed to **/Environment**.

eTM registry entries accept the following parameters:

- **PID**

  The keyword PID denotes the process identifier of the task. This identifier is formatted as a 10-digit numeric value. The PID is also shown when Microsoft Task Manager is used to represent the processes. A special value 0000000000 is shown when the task is not running.

- **Windowstyle**

  The keyword Windowstyle denotes the style of the window of the task. The supported values are shown in Table 4: Supported window styles on page 72.

  Table 4: Supported window styles

Value	Description
0	Window is hidden and focus is passed to the hidden window.
1	Window has focus and is restored to its original size and position.
2	Window is displayed as an icon with focus.
3	Window is maximized with focus.
4	Window is restored to its most recent size and position. The currently active window remains active.
6	Window is displayed as an icon. The currently active window remains active.

- **Shortcut**

  The keyword Shortcut denotes the command line parameter that is used to launch the process.

Figure 74: Sample registry file of the eTM, illustrating a Publisher site — part 1 on page 73 shows a sample (exported) registry file of the eTM section, and refers to a Publisher site, usually containing an eKERNEL reference.

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.76.255]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.76.255\SOPHO Messenger@Net - CSTA Service]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\CSTA_Service.exe\""
"Windowstyle"="1"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.76.255\SOPHO Messenger@Net - eAPI - Port 3212]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eAPI.exe\" /Site:1 /eKernel
      port:3212 /eKernel address:147.93.76.255 /Log drive:C"
"Windowstyle"="1"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.76.255\SOPHO Messenger@Net - eCAP - Port 3202]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eCAP.exe\" /Site:1 /eKernel
      address:147.93.76.255 /eKernel port:3202 /Log drive:C"
"Windowstyle"="1"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.76.255\SOPHO Messenger@Net - eDMSAPI - Port
      3201]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eDMSAPI.exe\" /Site:1 /
      eKernel address:147.93.76.255 /eKernel port:3201 /Log drive:C"
"Windowstyle"="1"

continued on next page...
```

Figure 74: Sample registry file of the eTM, illustrating a Publisher site — part 1

```
[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.76.255\SOPHO Messenger@Net - eKernel - Site 1]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eKERNEL.exe\" /Site:1 /
      licence:*NONE /keepalive:60"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.76.255\SOPHO Messenger@Net - eSMTP - Port 3211]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eSMTP.exe\" /Site:1 /
      eKernel address:147.93.76.255 /eKernel port:3211 /Log drive:C"
"Windowstyle"="1"
```

Figure 75: Sample registry file of the eTM, illustrating a Publisher site — part 2

Figure 76: Sample registry file, illustrating a Subscriber section in production mode on page 74 shows another example, illustrating a Subscriber section in production mode. There

is no eKERNEL reference in this example, as all modules refer to the eKERNEL on the publisher system.

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130\SOPHO Messenger@Net - CSTA Service]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\CSTA_Service.exe\""
"Windowstyle"="1"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130\SOPHO Messenger@Net - eCAP - Port 3403]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eCAP.exe\" /Site:1 /eKernel
      address:147.93.76.255 /eKernel port:3403 /Log drive:C"
"Windowstyle"="1"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130\SOPHO Messenger@Net - eDMSAPI - Port
      3401]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eDMSAPI.exe\" /Site:1 /
      eKernel address:147.93.76.255 /eKernel port:3401 /Log drive:C"
"Windowstyle"="1"
```

Figure 76: Sample registry file, illustrating a Subscriber section in production mode

Figure 77: Sample registry file, illustrating a Subscriber section in backup mode on page 75 shows another example, illustrating a Subscriber section in backup mode. Here an eKERNEL reference is shown, as the environment runs when the publisher is unavailable. All modules refer to the local eKERNEL on the subscriber system.

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130 (backup)\SOPHO Messenger@Net - eKernel
      - Site 1]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eKERNEL.exe\" /Site:1 /
      licence:*NONE /keepalive:60"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130 (backup)]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130 (backup)\SOPHO Messenger@Net - CSTA
      Service]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\CSTA_Service.exe\""
"Windowstyle"="1"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130 (backup)\SOPHO Messenger@Net - eCAP -
      Port 3403]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eCAP.exe\" /Site:1 /eKernel
      address:147.93.169.130 /eKernel port:3403 /Log drive:C"
"Windowstyle"="1"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130 (backup)\SOPHO Messenger@Net - eDMSAPI
      - Port 3401]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eDMSAPI.exe\" /Site:1 /
      eKernel address:147.93.169.130 /eKernel port:3401 /Log drive:C"
"Windowstyle"="1"
```

Figure 77: Sample registry file, illustrating a Subscriber section in backup mode

Registry settings eTM_HA

The configuration of environments and tasks is stored in the following section:

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM_HA]

This section contains additional configuration settings that are needed for configuring the high-availability functionality that is added in eTM_HA.

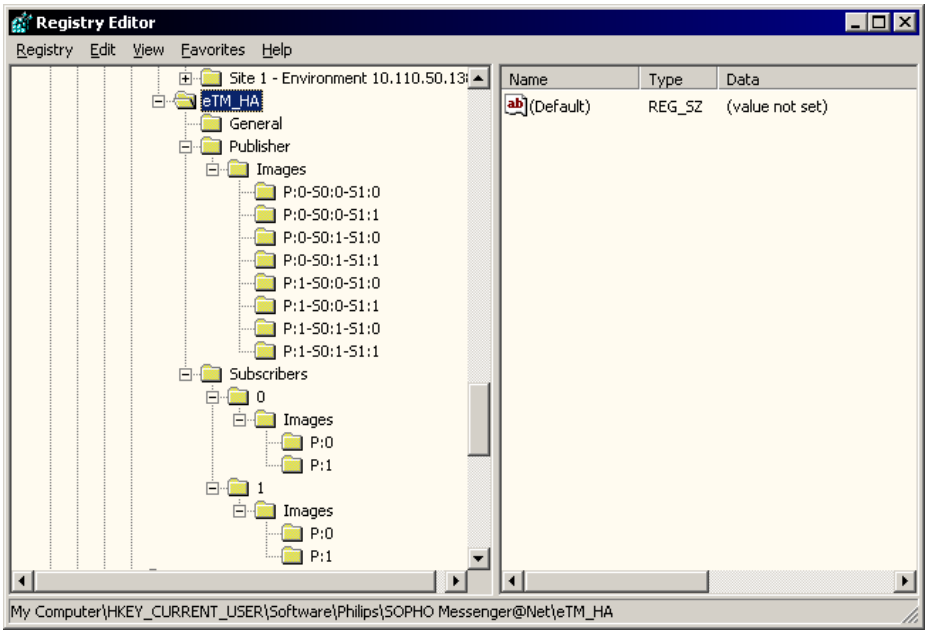


Figure 78: Registry settings: General section

The General section defines the following parameters:

- Interval CheckAvailability
- Interval CheckTasks
- Interval KeepAlive
- Interval GetImage
- Timeout KeepAlive
- Timeout GetImage
- Timeout Task
- Log days
- Publisher database
- Subscriber database
- Subscriber workspace
- Subscriber image

The Publisher section contains a structure as shown Figure 79: Registry settings: Publisher section on page 77:

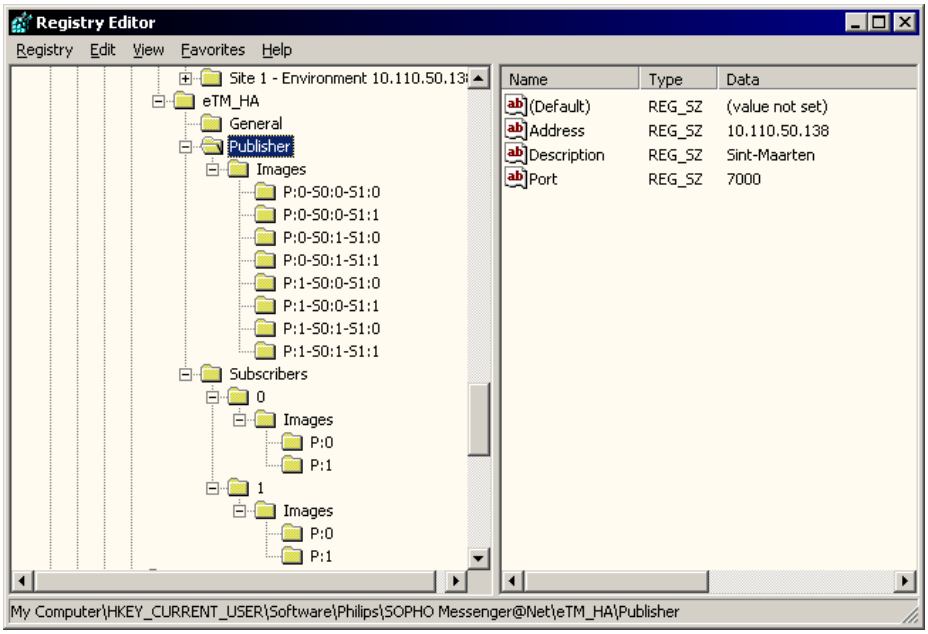


Figure 79: Registry settings: Publisher section

The same information is represented in the eTM_HA Overview tab, as illustrated in Figure 80: Registry settings: Publisher overview in eTM_HA on page 78.

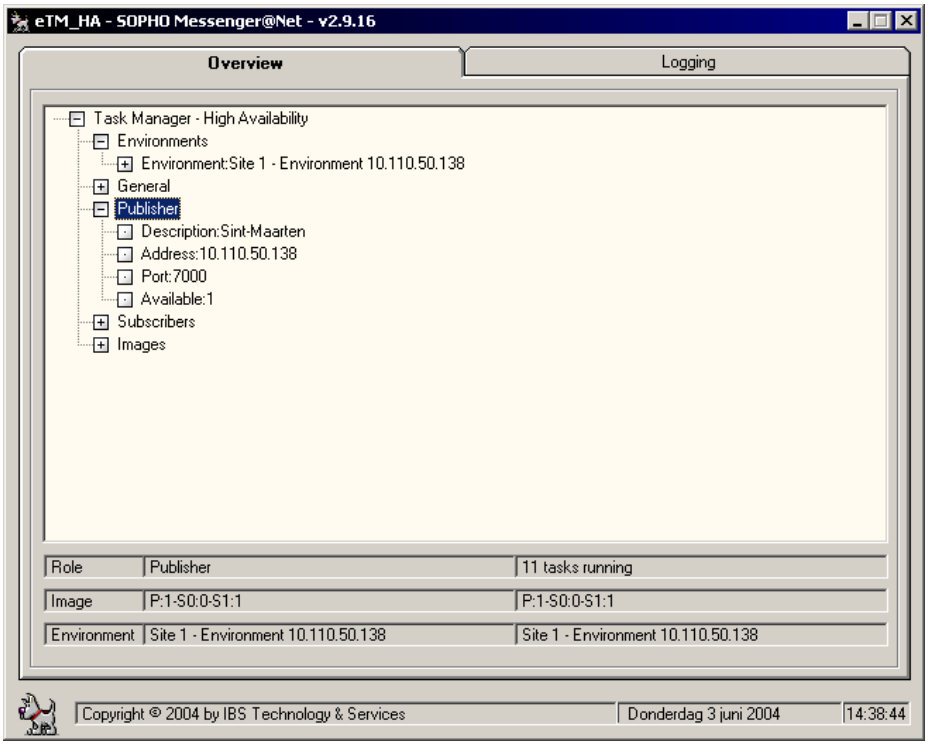


Figure 80: Registry settings: Publisher overview in eTM_HA

The Subscribers section contains a structure as illustrated in Figure 81: Registry settings: Subscribers (0) section on page 78 and Figure 82: Registry settings: Subscribers (1) section on page 79.

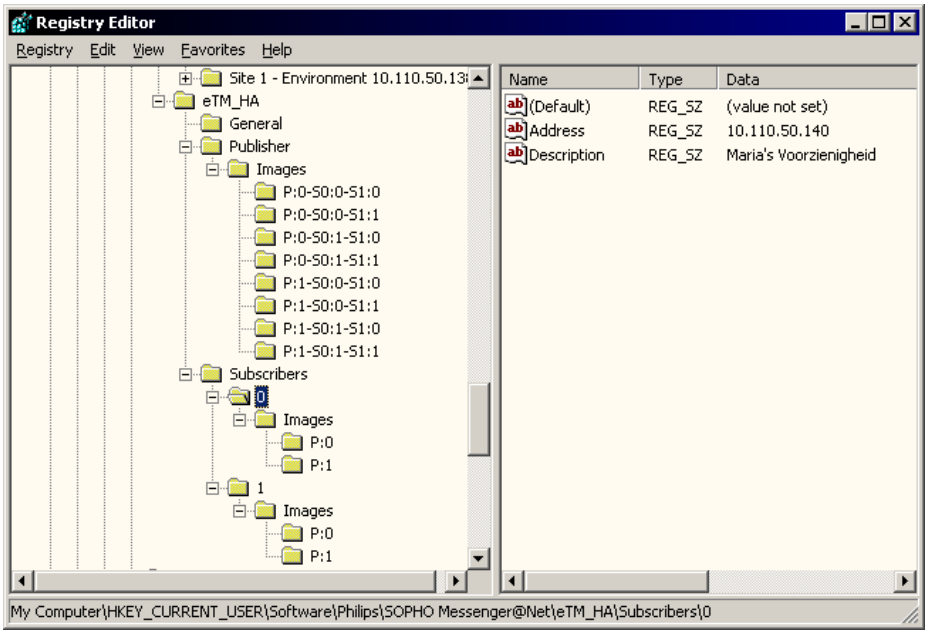


Figure 81: Registry settings: Subscribers (0) section

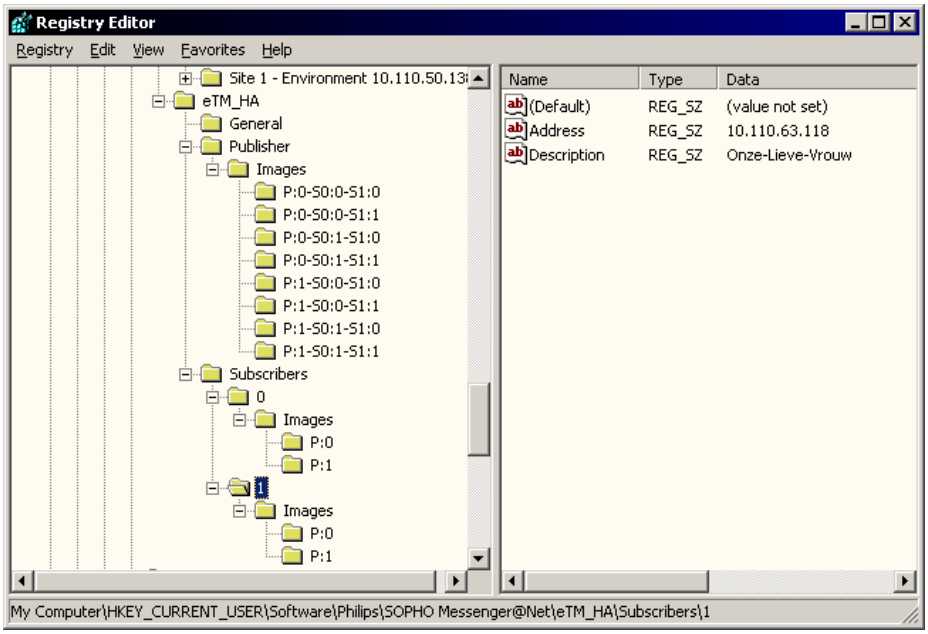


Figure 82: Registry settings: Subscribers (1) section

The same information is represented in the eTM_HA Overview tab, as illustrated in Figure 83: Registry settings: Subscribers overview in eTM_HA on page 80.

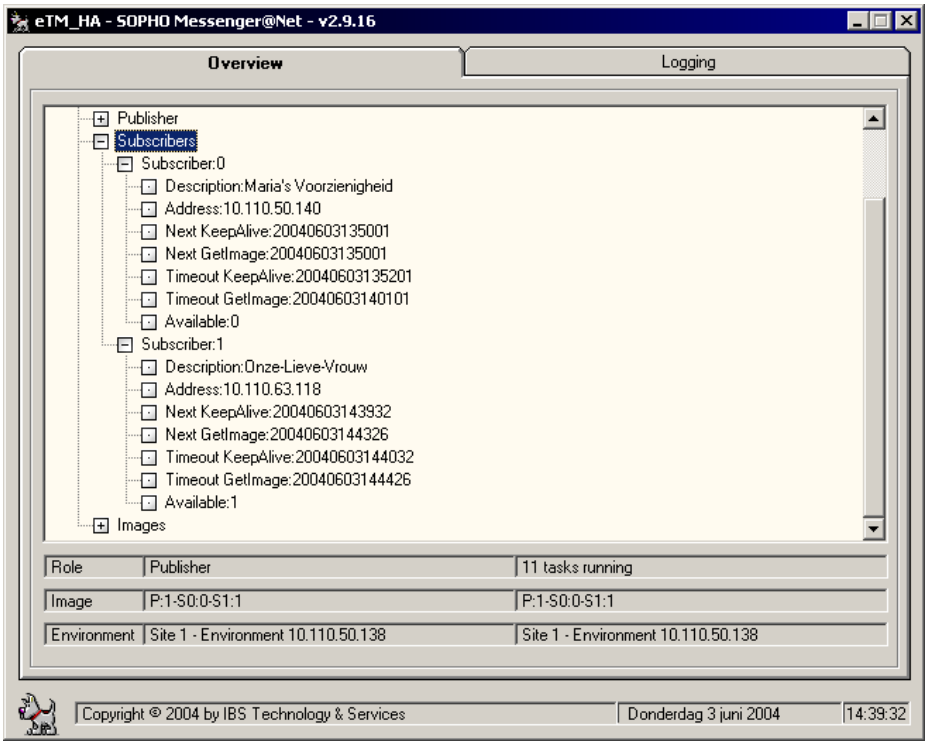


Figure 83: Registry settings: Subscribers overview in eTM_HA

The General section contains a structure as illustrated in Figure 84: Registry settings: General on page 80.

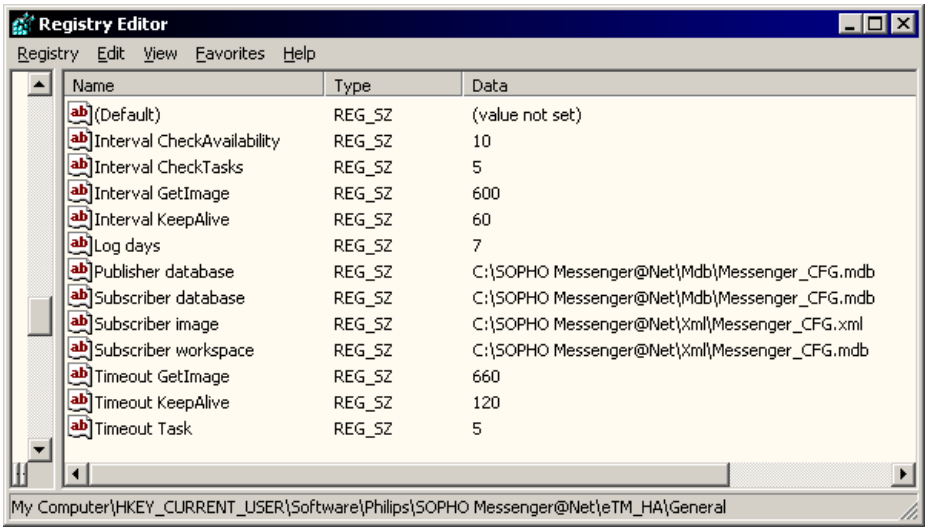


Figure 84: Registry settings: General

The same information is shown in the eTM_HA Overview tab, as shown in Figure 85: Registry settings: General in eTM_HA on page 81.

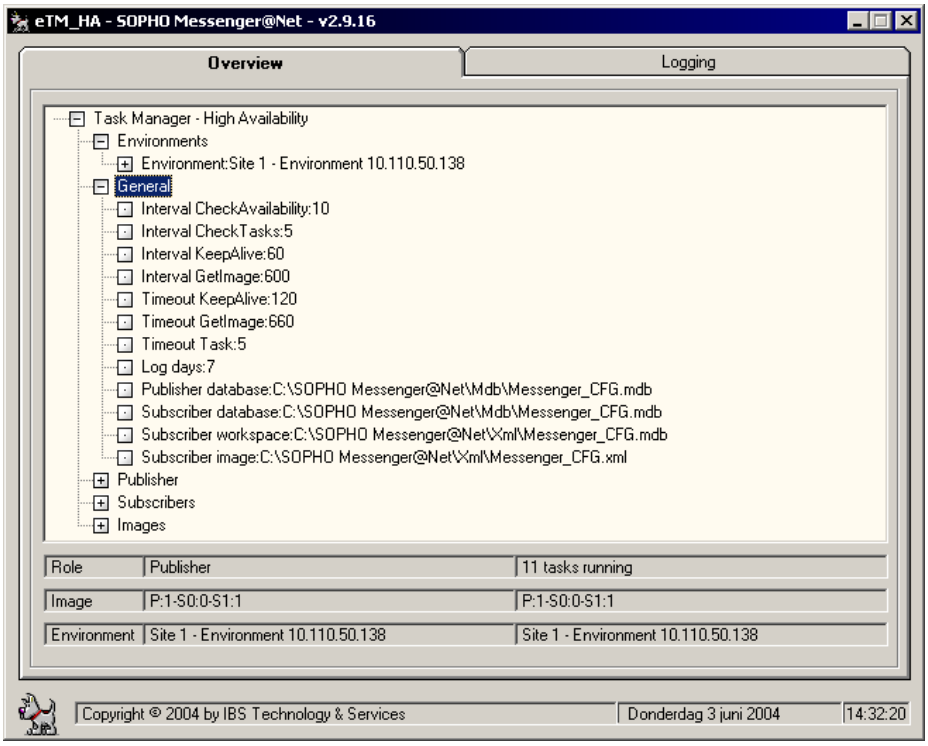


Figure 85: Registry settings: General in eTM_HA

Merging registry files

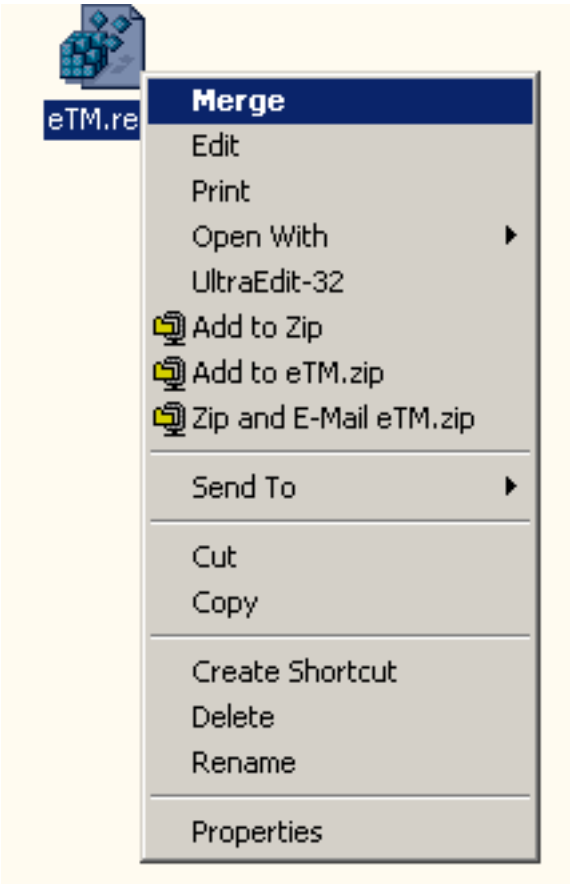
Use the steps in Merging .reg files on page 81 to merge registry files.

Merging .reg files

1. Select the Merge command.

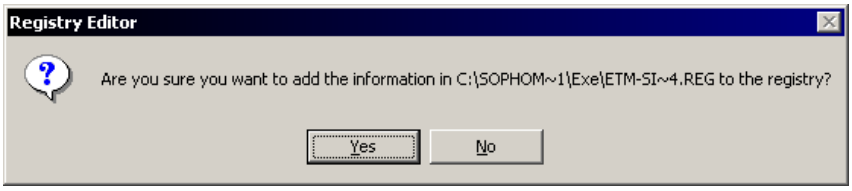
   In Windows Explorer:

   - Locate the file you wish to merge.
   - Right-click the file.
   - choose **Merge** from the pop-up menu, as follows:



2. Confirm that you wish to merge the registry.

   Choose **Yes** to continue.



3. Confirm completion of the registry merge.

   Click **OK**.

The command RegEdit or RegEdt32 can be used to verify the configuration, or to apply changes to an existing configuration.

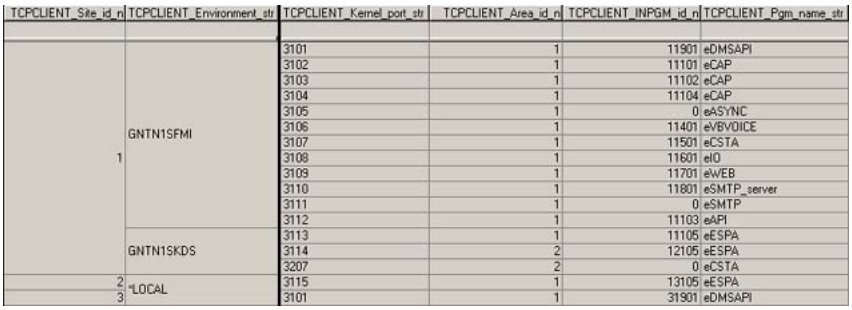
Future releases of DECT Messenger will provide automatic procedures for configuring the Task Manager from the Configurator module.

The eGRID module features a command button **Generate registry files for eTM**. Click this button to read the eKERNEL_TCPCLIENT table and automatically generate the required shortcuts for each site and environment, as shown in Generate shortcuts on page 83.

Generate shortcuts

1. Use eGRID to generate registry files for eTM.

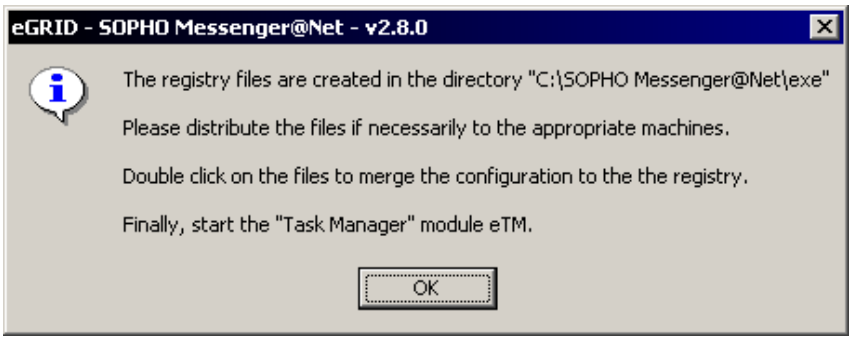
   Launch eGRID and click **Generate registry files for eTM**.



TCPCLIENT_Site_id_n	TCPCLIENT_Environment_str	TCPCLIENT_Kernel_port_str	TCPCLIENT_Area_id_n	TCPCLIENT_INPGM_id_n	TCPCLIENT_Pgm_name_str
1	GNTN1SFMI	3101	1	11901	eDMSAPI
		3102	1	11101	eCAP
		3103	1	11102	eCAP
		3104	1	11104	eCAP
		3105	1	0	eASYNC
		3106	1	11401	eVBVOICE
		3107	1	11501	eCSTA
		3108	1	11601	eIO
		3109	1	11701	eWEB
		3110	1	11801	eSMTP_server
		3111	1	0	eSMTP
		3112	1	11103	eAPI
	GNTN1SKDS	3113	1	11105	eESPA
		3114	2	12105	eESPA
		3207	2	0	eCSTA
2	*LOCAL	3115	1	13105	eESPA
3		3101	1	31901	eDMSAPI

2. Review the information provided, and acknowledge completion of the process.

   Click **OK** to continue.



Note:

Do not forget to verify that the names of the environments in the eTM registry keys contain the IP address; if not, rename the key to include the IP address. Avaya recommends that you use the following naming conventions in the registry: **Site n - Environment x.x.x.x** and **Site n - Environment x.x.x.x - backup**.

Check tasks

The program verifies all tasks with a time interval specified in the registry (usually 5 seconds).

When the eTM_HA is paused or stopped, the routine that verifies and restarts the process is temporarily interrupted.

This usually occurs during maintenance of one of more of the programs that are guarded by the eTM_HA. This temporary condition is shown in the logging.

A system administrator can also terminate a task from within the eTM_HA environment using a Terminate process API-call.

Note:

Using the Terminate process API-call can cause data loss, as this does not provide any graceful cleanup or shutdown of the associated program.

To terminate a process in the Task Manager, use the **Kill task menu** option as shown in Figure 86: Kill Task on page 84. The **Kill task** option is available only when the tree-view is expanded and the mouse is right-clicked on the PID:xxxxxxxxxx line.

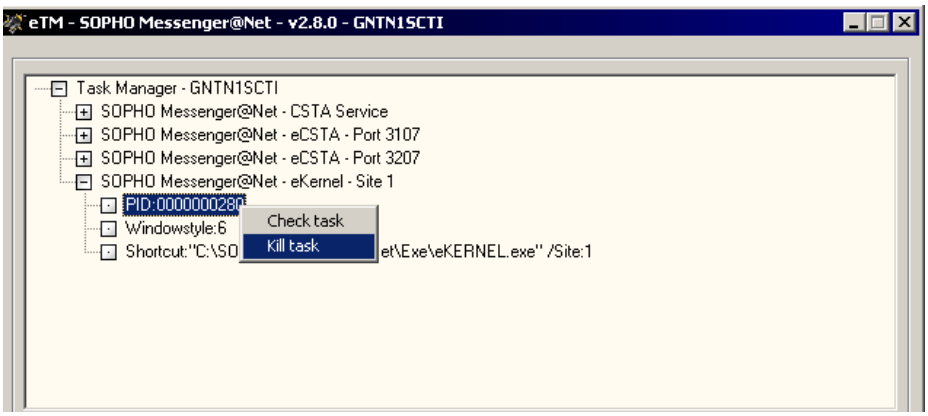


Figure 86: Kill Task

Note:

When the eTM is running, the system relaunches the terminated tasks within 10 seconds.

When the eTM form is closed through the control box on the right top of the form, the application does not shut down, but is instead minimized to an icon in the system tray. This function is designed to prevent the user from accidentally closing the eTM and associated tasks. This approach is similar to monitoring applications of other vendors, such as the Apache Monitor or the SQL Server Service Manager.

Shutting down eTM_HA

The eTM can be shut down by means of the pop-up menu shown in Figure 69: Open Task Manager on page 67, using the **Exit** menu option.

Important:

Avaya recommends that you close applications using shut down or exit/close options in the applications, to ensure a clean shutdown. This helps to protect volatile data, properly close

down serial and sockets communications, free resources, clean up garbage, and so on. To stop the processes gracefully, follow the steps in Shutting down eTM_HA on page 85.

Shutting down eTM_HA

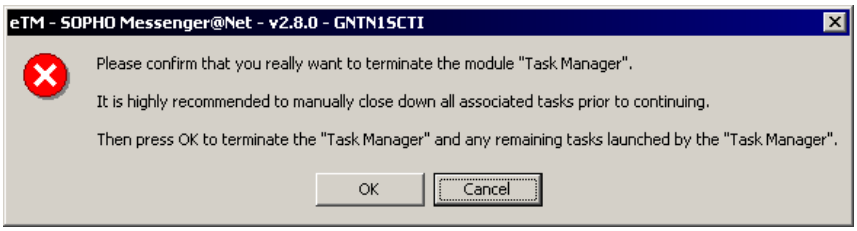
1. Open the eTM_HA pop-up menu.

   Right-click the eTM_HA icon in the system tray.

2. Stop the eTM_HA.

   - Choose the menu item **Task Manager - Stop**.
   - Choose the menu item **Exit**.

   The following confirmation prompt is shown; do not click **OK** or **Cancel** yet:



   Note:

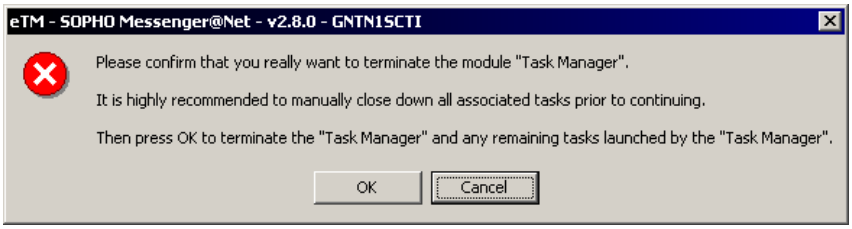
   The application also responds to a system Log off or Shut down event.

3. Shut down the applications.

   Close down all programs using the program specific instructions. In most cases this means closing the main form of each application by clicking the close box on the top right of each form. However, some applications require specific shutdown procedures.

4. Confirm the eTM termination warning dialog box.

   Click **OK**.



   Because all associated tasks were already manually ended gracefully, no more processing is involved.

   Any associated tasks still running are terminated through a **Terminate process** API-call for each task that is launched from within the eTM and finally shuts down the eTM module as well.

Publisher

The publisher instance of eTM_HA features a TCP Server listing on a port specified in the registry. Typically, port 7000 is used as the default port. The TCP Server is a multiple-accept model, so multiple clients can connect at the same time. The number of simultaneous connections is also defined in the registry. Specify a number at least as great as the number of subscribers. Avaya recommends specifying a value that equals the number of subscribers multiplied by three, to provide room for recovery in case of bad connection attempts.

The netstat command can be used on the Publisher to verify that the TCP Server is listening (sample data is shown in Figure 87: Sample netstat command and returned data on page 86).

```
C:\>netstat -a -n

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:21             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:25             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:90             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:443            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:7000           0.0.0.0:0              LISTENING
:
```

Figure 87: Sample netstat command and returned data

Important:

The TCP Server is used for internal processing only. Do not attempt to access the server unless instructed to do so.

The TCP Server is to be accessed from the Subscribers only. You can test this connection (from the subscriber PCs only) with Internet Explorer. An HTTP request to port 7000 must reply with the error code shown in Figure 88: TCP Server Error response on page 87.

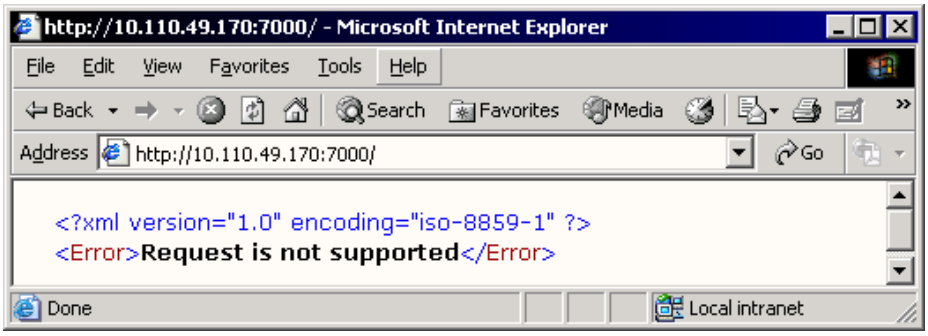


Figure 88: TCP Server Error response

In an operational environment, the eTM_HA instances of the Subscribers send these two requests to the publisher on a regular basis: KeepAlive, and GetImage.

- **KeepAlive**

  A KeepAlive request is exchanged between subscriber and publisher, and allows both parties to verify the presence of the other. Interval and timeout between attempts are defined in the registry.

  Figure 89: TCP Server Keep Alive response on page 87 shows an example of what is sent during this exchange. To test this response, Avaya recommends using Internet Explorer on the Subscriber.

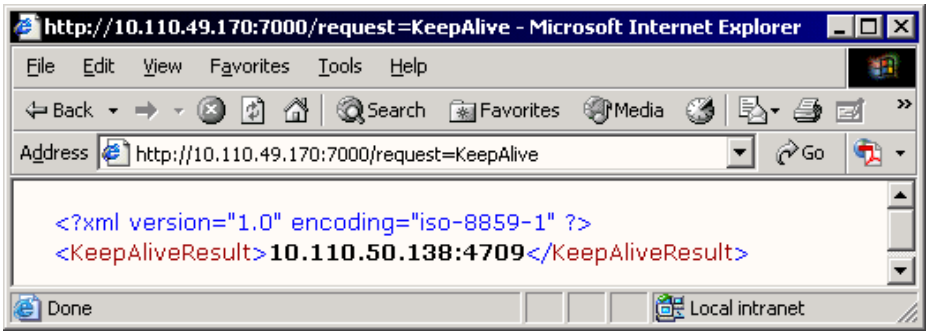


Figure 89: TCP Server Keep Alive response

- **GetImage**

  The GetImage request is sent from each Subscriber to the Publisher on a regular time interval, as specified in the registry. The publishing system responds to such a request with an XML image of the Messenger_CFG database. Figure 90: TCP Server Get Image response on page 88 shows an example.

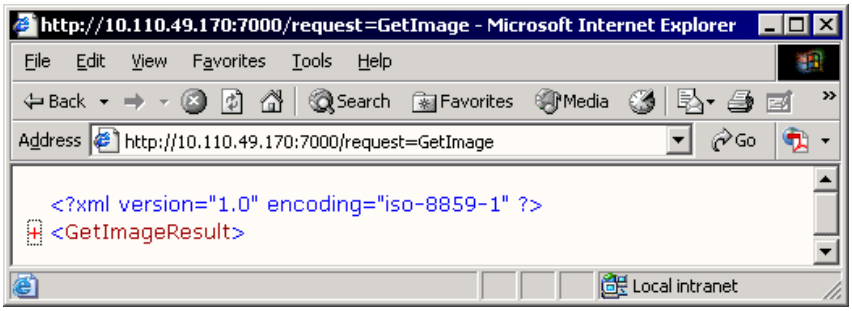


Figure 90: TCP Server Get Image response

The XML image file provided by the GetImage request can be expanded and collapsed with the plus (+) and minus (-) signs, as shown in Figure 91: Expanded information on page 88. For more information on the XML image, see XML image on page 92.

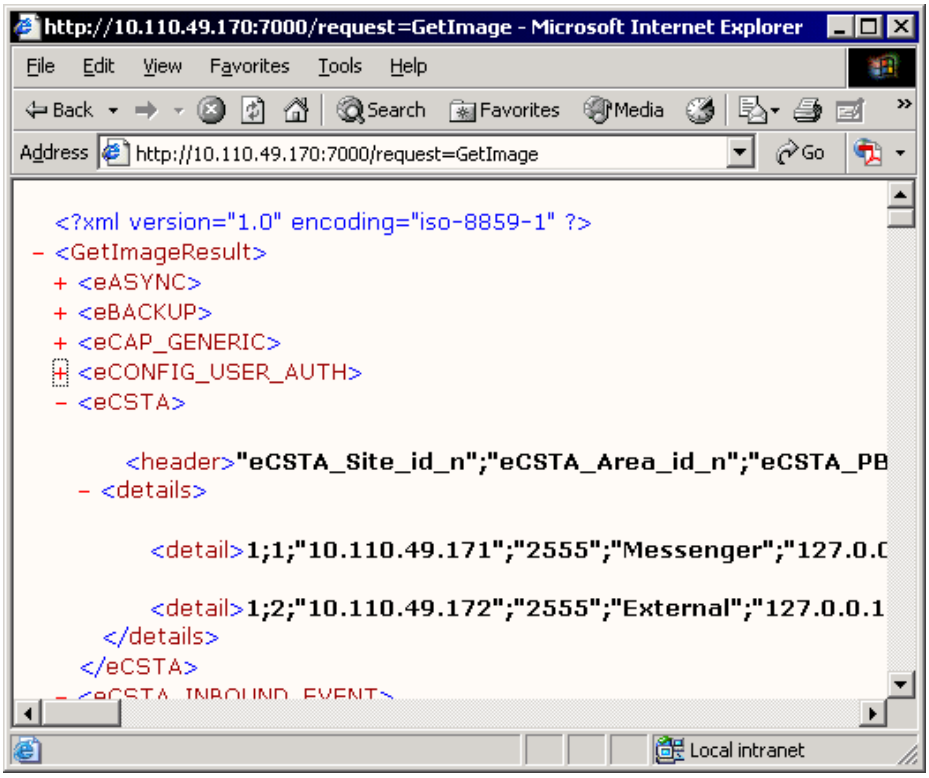


Figure 91: Expanded information

Note:

Messenger_CFG contains sensitive data, and is exchanged as plain text in XML format. To prevent security exposure, HTTP requests from external systems are rejected with an authentication error. This test is performed based upon the IP address of the requester.

Note:

In a WAN environment, a test with a Browser can lead to rejection, even from the subscriber system. The most common cause is a proxy server that masks the IP address of the

subscriber. During tests with Internet Explorer, you must disable the proxy server for local addresses or specify the IP address of the publisher in the bypass list.

Keeping track of states

Both publisher and subscriber keep track of the state of the other party. This leads to a so-called "image" of Boolean settings of publisher and subscriber.

Subscriber

On the subscriber level, there is a state represented by P:0 and P:1, indicating whether the publisher can be reached. P:1 denotes the publisher is available, P:0 denotes the publisher is unavailable.

Appropriate registry settings associate an environment to each image. Optionally an SQL script can be defined to run during switching environments.

The registry definitions are shown in Figure 92: Registry definitions on page 89.

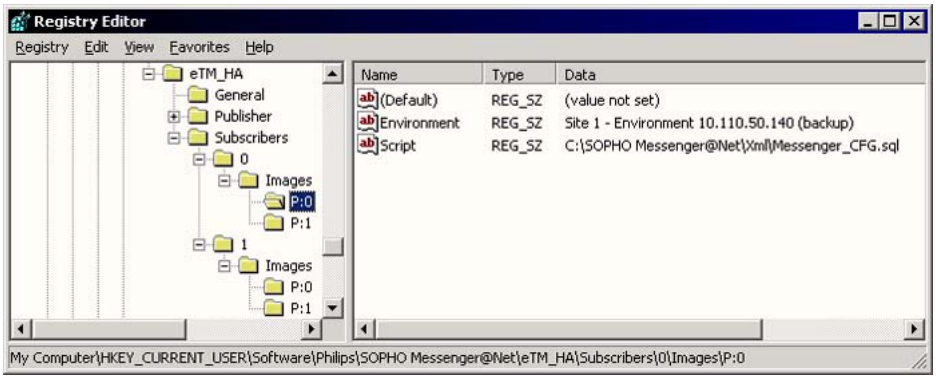


Figure 92: Registry definitions

In Figure 92: Registry definitions on page 89, an example is shown with one Publisher and two Subscribers; Subscriber 0 has an image for handling P:0 and P:1. This example shows the settings when the subscriber cannot connect to the publisher. The environment Site 1 – Environment 10.110.50.140 (backup) is associated and SQL script Messenger_CFG_sql is defined.

Publisher

On the publisher level, a similar registry image is used. However, as a Publisher is often in contact with multiple subscribers, the available images grow exponentially, as the state of publisher and every subscriber forms a number of combinations for each Boolean state.

Figure 93: Example of an image at the publisher level on page 90 shows a network where one Publisher and two Subscribers lead to eight images, and depend on each Boolean state of available (1) or unavailable (0).

The syntax for images on publisher level are similar to P:1:S0:1-S1:1. Each section is separated by a minus sign (-).

- The P:1 or P:0 denotes the state of the publisher
- The S0:1 or S0:1 denotes the state of first subscriber
- The S1:1 or S1:1 denotes the state of second subscriber

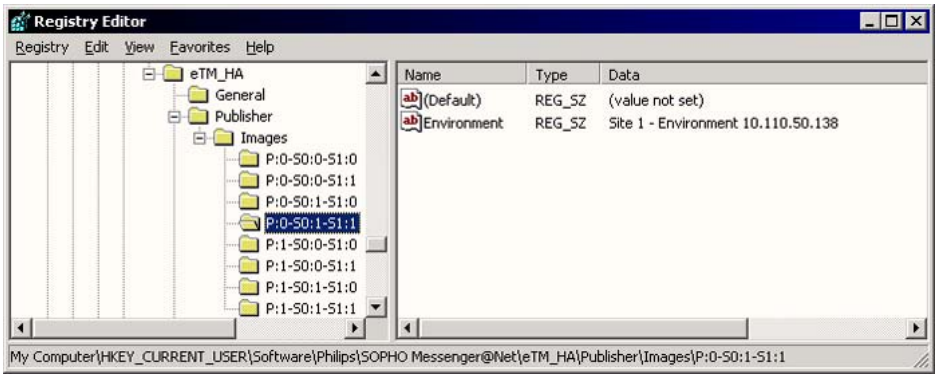


Figure 93: Example of an image at the publisher level

The registry keys are to be entered manually.

Recommendation

Avaya recommends that you begin with a definition on the Publisher level that refers to the same environment for each image, and with a definition on the Subscriber level that refers to the same environment for each image.

In this initial setup no environment changes occur, and initial testing can take place.

In a later stage you can modify environments. A copy of the production environment is usually made at the Subscriber sites, for example, Site 1 – Environment 10.110.50.140 and Site 1 – Environment 10.110.50.140 (backup). In this backup environment, the tasks can be altered, for example, an eKERNEL instance can be added, and eKERNEL_address refers to a local instance of eKERNEL.

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130\SOPHO Messenger@Net - CSTA Service]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\CSTA_Service.exe\""
"Windowstyle"="1"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130\SOPHO Messenger@Net - eCAP - Port 3403]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eCAP.exe\" /Site:1 /eKernel
      address:147.93.76.255 /eKernel port:3403 /Log drive:C"
"Windowstyle"="1"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130\SOPHO Messenger@Net - eDMSAPI - Port
      3401]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eDMSAPI.exe\" /Site:1 /
      eKernel address:147.93.76.255 /eKernel port:3401 /Log drive:C"
"Windowstyle"="1"
```

Figure 94: Example: Site 1 - Environment 147.93.169.130

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130 (backup)\SOPHO Messenger@Net - eKernel
      - Site 1]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eKERNEL.exe\" /Site:1 /
      licence:*NONE /keepalive:60"
"Windowstyle"="6"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130 (backup)]

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130 (backup)\SOPHO Messenger@Net - CSTA
      Service]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\CSTA_Service.exe\""
"Windowstyle"="1"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130 (backup)\SOPHO Messenger@Net - eCAP -
      Port 3403]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eCAP.exe\" /Site:1 /eKernel
      address:147.93.169.130 /eKernel port:3403 /Log drive:C"
"Windowstyle"="1"

[HKEY_CURRENT_USER\Software\Philips\SOPHO Messenger@Net\eTM\Site 1 -
      Environment 147.93.169.130 (backup)\SOPHO Messenger@Net - eDMSAPI
      - Port 3401]
"Shortcut"="\"C:\\SOPHO Messenger@Net\\Exe\\eDMSAPI.exe\" /Site:1 /
      eKernel address:147.93.169.130 /eKernel port:3401 /Log drive:C"
"Windowstyle"="1"
```

Figure 95: Example: Site 1 - Environment 147.93.169.130 (backup)

XML image

The Subscriber receives the result of the GetImage in a flat-file repository, located in the following directory:

C:\SOPHO Messenger@Net\Xml

Important:

This directory must be created manually on Subscriber systems. Also a copy of the Messenger_CFG.mdb with the exact layout of the database of the publisher must be created in this directory. If the database is missing or has incorrect layout, system malfunction results. An update of eKERNEL on the Publisher site must always be synchronized with the

same update on subscribers systems, and the eKERNEL can automatically add changes to the database at startup. Therefore, after applying a new version of eKERNEL, you must first start eKERNEL, and then copy the Messenger_CFG.mdb database.

If you install new versions of eKERNEL, you must synchronize the eKERNEL modules on all systems. Also the latest layout of Messenger_CFG of publisher (after automatic upgrade changes at first run) must be manually placed in the directory of the Subscribers.

GetImage puts a file Messenger_CFG.xml in the same directory, and replaces this file on receipt of a GetImage result.

If you want to review this file, make a copy in another location before doing so, for example, C:\Temp. You can, for example, launch the Internet Explorer and associate XML files to this program, as Internet Explorer has built-in functionality to parse XML documents.

⚠ Warning:

Do not open the file in the C:\SOPHO Messenger@Net\Xml, because the file can be allocated by the viewer, and must be replaced when the next KeepAlive result is received.

SQL script

When a Subscriber detects a change between P:1 and P:0:

1. The Subscriber ends all running tasks associated with the current environment.

   At this time, Messenger functionality is disrupted, and pending and new alarms can be lost.

2. The SQL image in C:\SOPHO Messenger@Net\Xml repository Messenger_CFG.xml is imported to the workspace C:\SOPHO Messenger@Net \Xml access database Messenger_CFG.mdb. For this reason, the Messenger_CFG.mdb on Publisher and Subscriber sites must always use the same layout.

Thus the following repositories exist:

- (original database on publisher)

  C:\SOPHO Messenger@Net\Mdb\Messenger_CFG.mdbat publisher

- (transferred as XML images through HTTP)

  C:\SOPHO Messenger@Net\Xml\Messenger_CFG.xmlat subscriber

- (converted into MDB on subscriber)

  C:\SOPHO Messenger@Net\Xml\Messenger_CFG.mdbat subscriber

- (processed through optional SQL script, described in Figure 96: Sample SQL script on page 94)

```
C:\SOPHO Messenger@Net\Xml\Messenger_CFG.mdbat subscriber
```

- (activated on subscriber)

```
C:\SOPHO Messenger@Net\Mdb\Messenger_CFG.mdbat subscriber
```

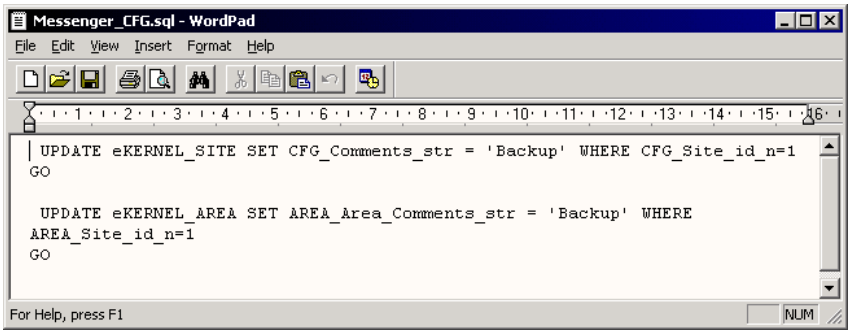


Figure 96: Sample SQL script

Use this (optional) SQL script to modify the contents of the database, as arrived from the operational publisher system. In some environments no changes are required; in more demanding customer environments complex scenarios can be set up to allow applying changes to the database. This can include changing .COM ports, IP address, group members, and so on. Review your SQL changes carefully.

Switch back

When Publisher or Subscribers detect a change in the availability image, a switch to another environment – or switch-back to the original environment – can be appropriate.

Conclusion

Careful planning and testing is required. Avaya recommends you simulate every configured scenario, and analyze in detail the possible impact of every scenario. An off-site testing procedure in a lab environment is usually appropriate, to prevent loss of alarms (during change in image, eKERNEL and other modules are stopped and all alarms can be lost).

Also, take into account that having a high-availability solution in place affects change management. Changes applied to eKERNEL must be synchronized, and (automatic) database upgrade changes to Messenger_CFG.MDB must be handled manually by the system administrator.

Finally, note that configuration changes with eCONFIG, eGRID, eWEB, or another configuration tool can affect the total environment. For example, a divert to another device

does not work in a backup environment if the destination device in not available, or the module is unavailable. Therefore, due to the nature of such an architecture, and maintenance issues and customer specific factors that are beyond our control, the authors of eTH_HA cannot accept responsibility for malfunction of the software.

Chapter 6: Module - eVBVOICE

Important:

Due to the ongoing development of the DECT Messenger product suite, some modules that provide additional functionality may become available after the initial release of DECT Messenger 4.0.

The following modules are described in this document but are not available at initial General Availability of Release 4.0.

- eFR
- eLICENSE
- eLOCATION
- eSMS
- eSNMP
- eVBVOICE

The eFR module is an add-on module and is licensed separately through the eLICENSE module. Some of the modules listed in this attention box are available only on a site-specific basis.

Introduction

Within every eVBVOICE application there are two different call types.

- Inbound calls

  In an inbound call where the user dials a specific number related to the eVBVOICE application, messages can be set (*SET), reset (*RESET), and confirmed (*CONFIRM). You can also record (*RECORD) wave files related on a specific menu option or recording general wave files.

- Outbound calls

  The eVBVOICE application can set up a call to a device and play a wave file linked to a specific alarm. The eVBVOICE application is used to inform the device of a user that a specific alarm is activated. The alarm is audio and not visual.

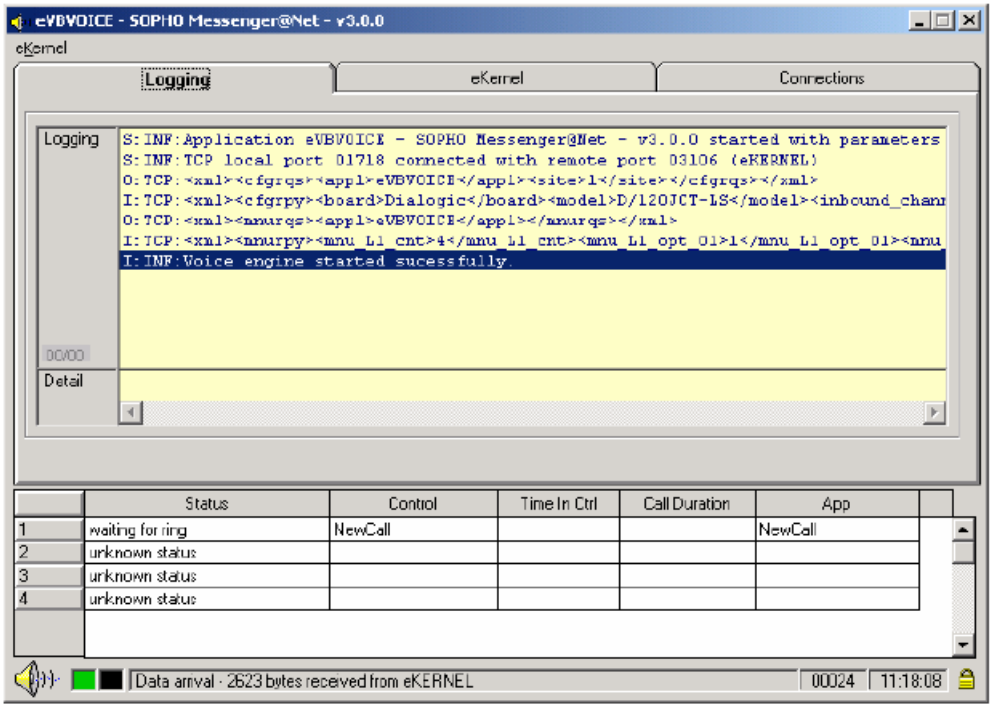


Figure 97: eVBVOICE inbound call

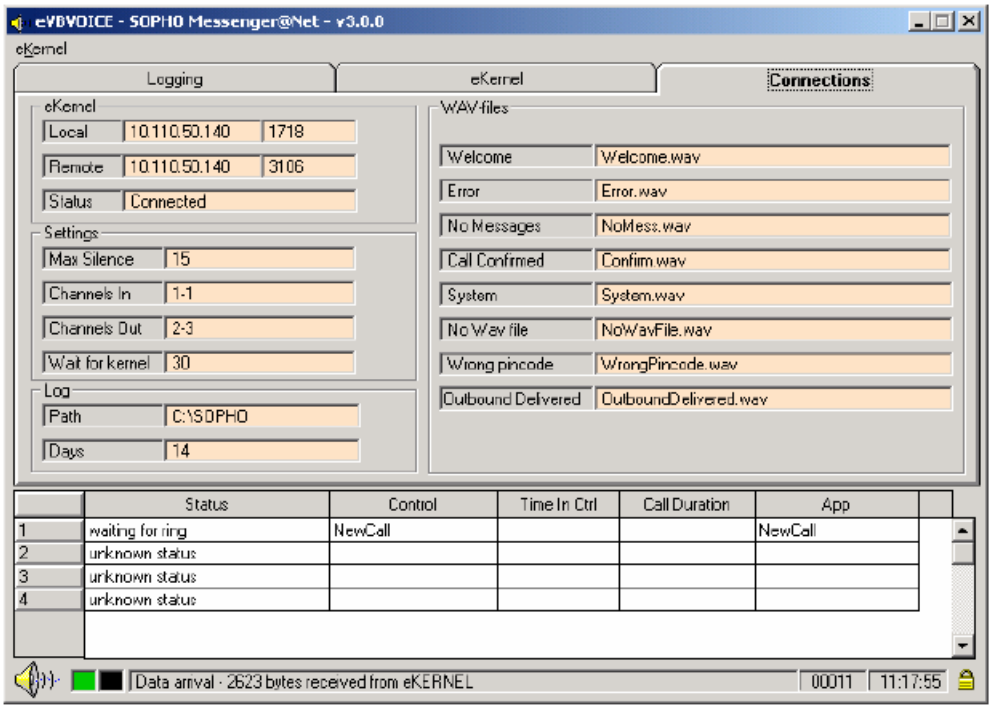


Figure 98: eVBVOICE outbound call

Inbound calls

General configuration parameters

General parameters for each eVBVOICE application are specified in the eVBVOICE table.

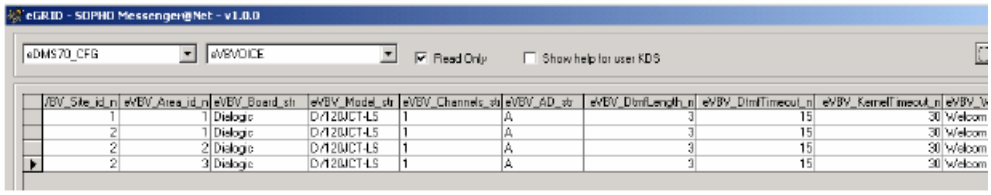


Figure 99: General configuration parameters

An eVBVOICE application must be unique for every site + area.

Functionality

The telephone number when a user calls eVBVOICE can be a group number, or a direct number, as defined in the telephone switch.

The users hears the Welcome message, eVBV_Welcome_str and then can select specific menu options. The menu options are described in the following configuration tables. All menu selections are generated with DTMF keystrokes, and every option must be terminated by the pound (#) key.

eVBVOICE menu option level 1

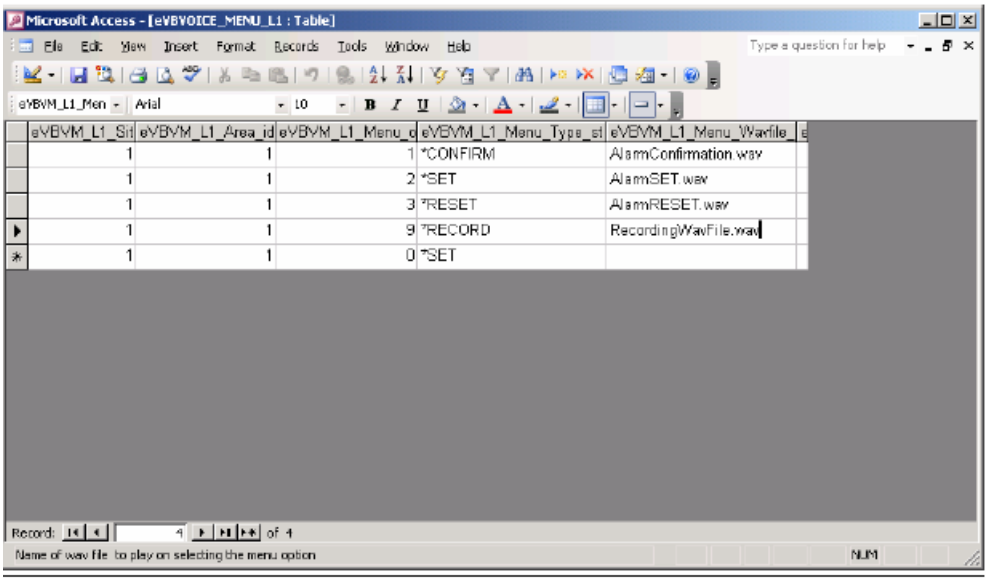


Figure 100: Menu option level 1

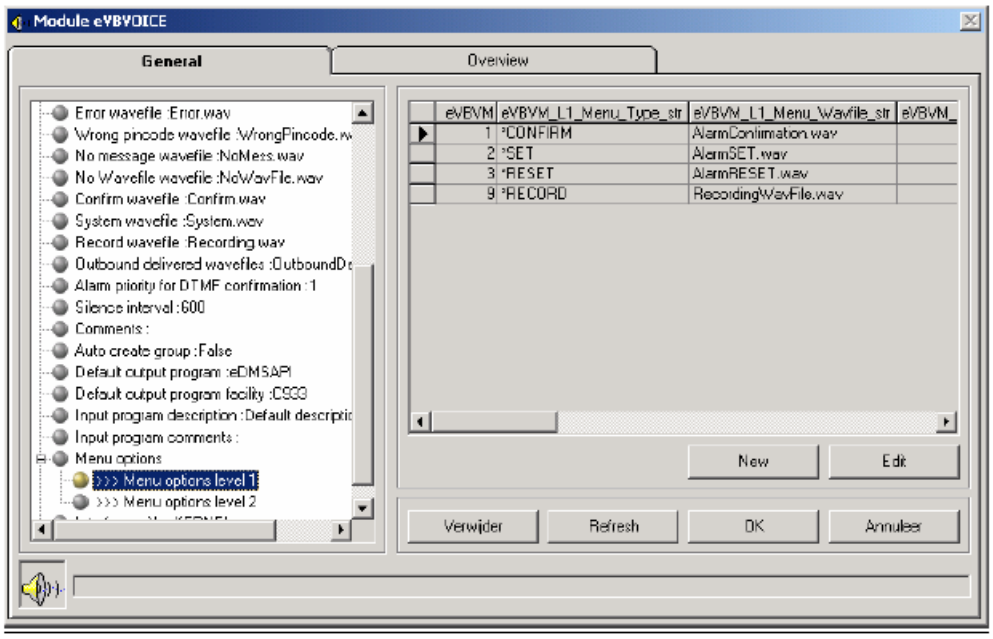


Figure 101: Menu option level 1 - General tab

Depending on the data entered in the configuration table for the first menu, eVBVOICE_MENU_L1, the user can select an option by pressing the corresponding keystroke or dtmf tone. The users hears the wave file linked with that option.

When a user hears the Welcome message, and presses 1 and #, for example, the users hears the wave file AlarmConfirmation.wav.

There are 4 different types of menu options possible for eVBVM_L1_Menu_Type_str =

- *CONFIRM
- *SET
- *RESET
- *RECORD

CONFIRM

For menu type = *CONFIRM for confirmation of alarms, there are no entries possible in the eVBVOICE_MENU_L2 table.

With the eVBVOICE application, users can confirm messages with a personal pincode (DEV_PinCode_str in eKernel_device).

If a message that is sent contains a message that must be confirmed, the user can make a call to the eVBVOICE application to confirm the alarm.

The user hears a welcome message when connected and must select the menu option for confirmation of alarms (*CONFIRM). The user is prompted to enter his pincode (eVBVM_L1_Menu_Wavfile_str).

This pincode is related to a device configured in the DEVICE table in the configuration database (DEV_PinCode_str in eKernel_device table).

The eKernel application checks the database for the devices related to this pincode, and informs the eVBVOICE application of the number of devices. If the pincode is unknown in the database, the "WrongPincode message" (eVBV_WrongPincode_str) plays.

If there are no alarms active for this pincode, the user hears a message that no messages are active for this pincode (VBV_Nomess_str).

If the confirmation is successful, all the active alarms for all the devices related to this pincode are cleared. The wave file entered in the eVBV_Confirm_str field plays.

If a wave file entered in the configuration tables does not exist, the wave file entered in the eVBV_NoWavFile_str field plays.

It is not possible to confirm an alarm during an outbound call in the current release. When a person receives a message through eVBVOICE, for example a call at home, the user cannot confirm the alarm or alarms during the same call. In the current release you must make a new call, particularly to confirm alarms.

SET and RESET

For menu type (eVBVM_L1_Menu_Type_str) = *SET and *RESET (set and reset of alarms), a link to the eVBVOICE_MENU_L2 table is necessary.

Menu option level 2

SET and RESET

For menu types *SET and *RESET, the related alarms must be specified in the eVBVOICE_MENU_L2 table.

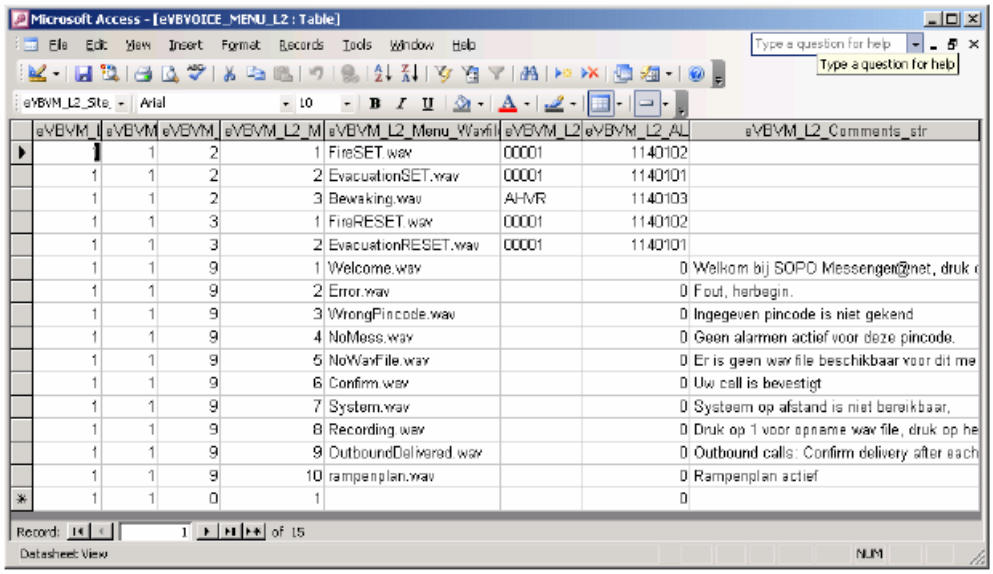


Figure 102: eVBVOICE option level 2

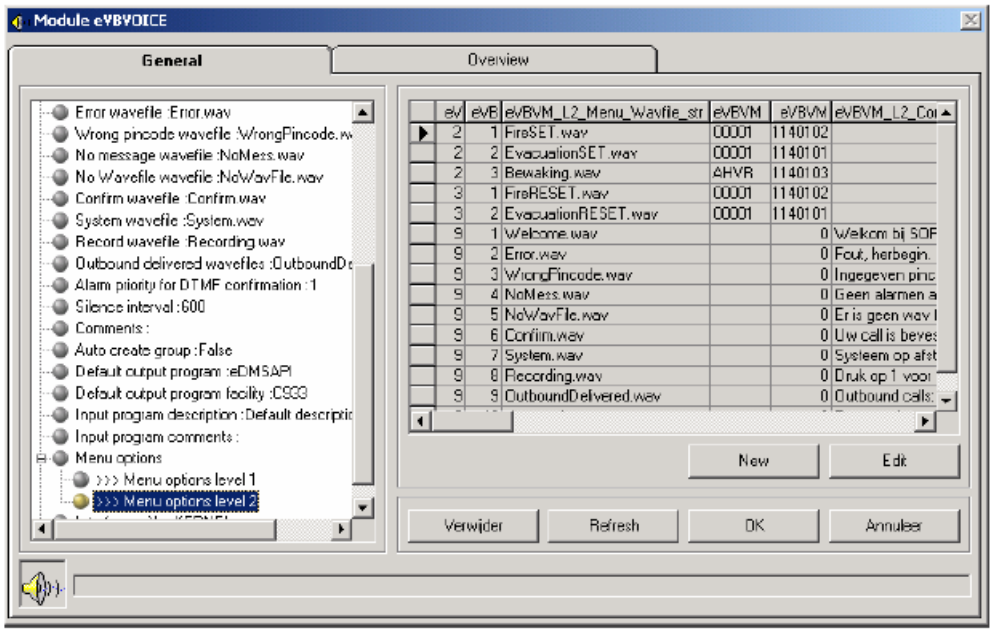


Figure 103: Menu options level 2 - General tab

All possible alarms that can be set (*SET) or reset (*RESET) with a dtmf keystroke must be entered in this table.

Although there is no password access protection, you can secure the activation of an emergency by choosing a very large menu option. Choosing a large menu option is very difficult to guess by an unauthorized person.

When a message is set or reset, and accepted by the eKernel (remote system), the user hears the message on the menu option level 2 (eVBVM_L2_Menu_Wavfile_str). Otherwise, the Error message (eVBV_Error_str) plays.

Recording wave files

RECORD

Through a specific option specified in the Menu Level 1 table, some users can record the wave files linked to the different options. For example, when a user selects option 9 (see configuration in the Menu L1 table), the user who records the wave file enters the menu option L1 and terminates with the # key. For recording wave files specified in menu L2 table, the user enters option L1, enters a * (as a separator between menu Level 1 and menu Level 2), enters option L2, and terminates by #.

Example site 1:

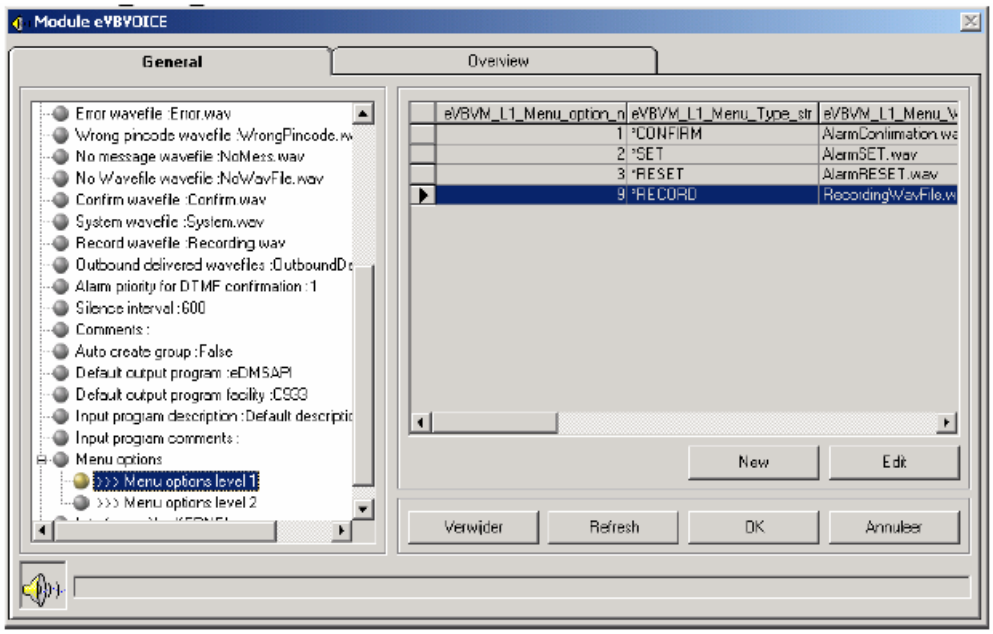


Figure 104: eVBVOICE_Menu_L1

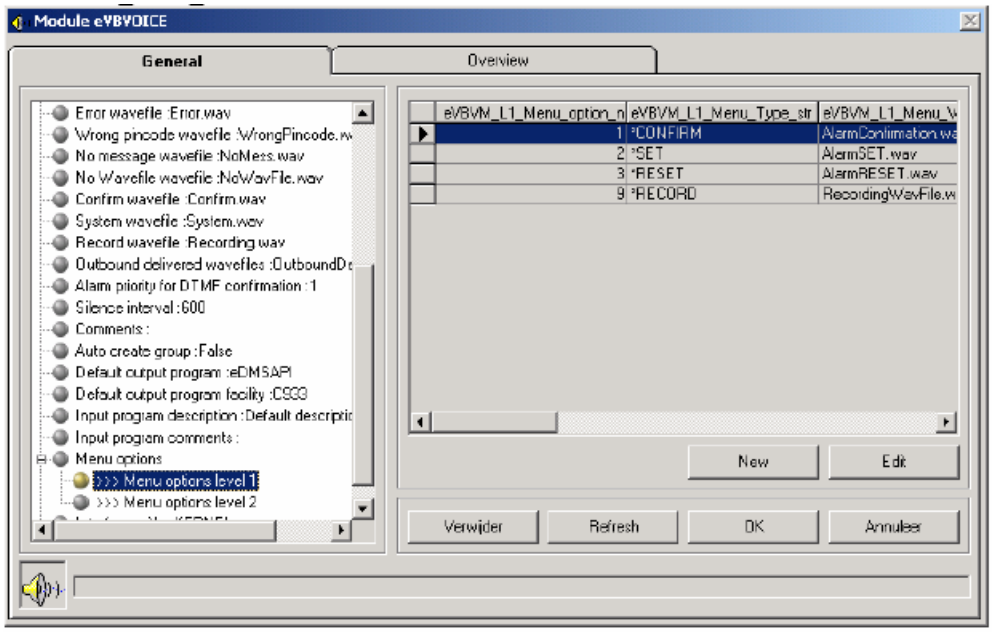


Figure 105: eVBVOICE_Menu_L2

To make a call to the eVBVOICE application, enter:

- 1#

  to record AlarmConfirmation.wav file

- 2#

to record AlarmSET.wav file

- 2*1#

  to record EvacuationSET.wav file

For recording the general wave files specified in the eVBVOICE table, the wave files must be specified in the eVBVOICE_MENU_L2.

Menu L1 option	Menu L2 option	Wav file	Contents of wave file
9	1	Welcome.wav	Welcome at SOPHO Messenger@net, press 1 to set an alarm, press 2 to …
9	2	Error.wav	Error, please retry
9	3	NoMess.wav	No active alarms for this pincode.
9	9	Rampenplan.wav	Emergency plan is activated!
9	…		

Figure 106: eVBVOICE MENU Options table

Also wave files specified in the eKernel_Message_format table can be entered in the eVBVOICE_MENU_L2 table. These wave files can be then be recorded by the eVBVOICE application.

A wave file can also be recorded by another infrastructure, but it is very important to be aware that the format must be compatible (Audio Format : CCITT u-Law 8,000 kHz; 8 bit; Mono).

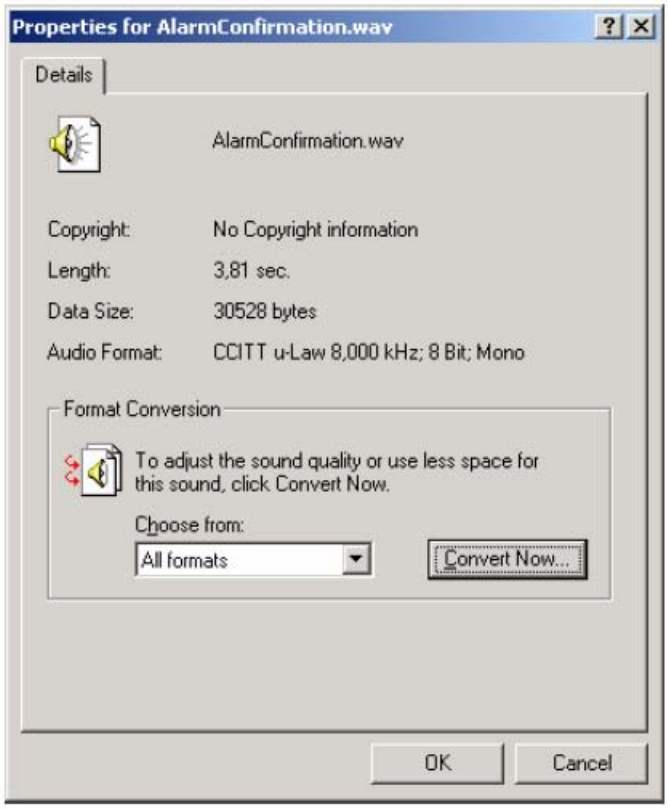
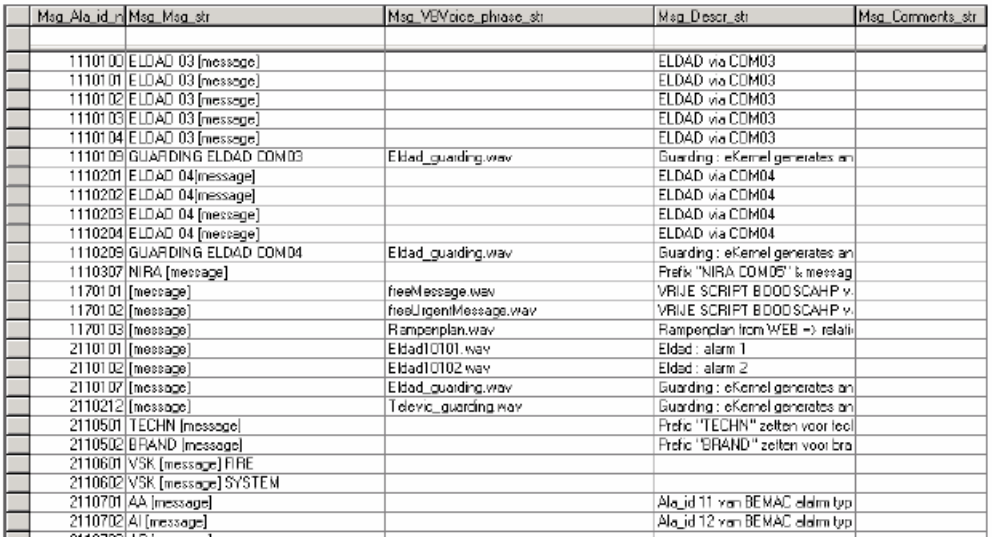


Figure 107: Properties for AlarmConfirmation.wav

Outbound calls

An alarm that is linked with a wave file, if the wave file exists, can be sent to a device with the eVBVOICE module.

Once all the wave files for a specific call are played, you enter a keystroke to ensure messages are delivered. This functionality is necessary to be sure the user hears the messages. Answering machines and voice mails perform in the same manner.



Msg_Ala_id_n	Msg_Msg_str	Msg_VBVoice_phrase_str	Msg_Descr_str	Msg_Comments_str
1110100	ELDAD 03 [message]		ELDAD via COM03	
1110101	ELDAD 03 [message]		ELDAD via COM03	
1110102	ELDAD 03 [message]		ELDAD via COM03	
1110103	ELDAD 03 [message]		ELDAD via COM03	
1110104	ELDAD 03 [message]		ELDAD via COM03	
1110109	GUARDING ELDAD COM03	Eldad_guarding.wav	Guarding : eKernel generates an	
1110201	ELDAD 04 [message]		ELDAD via COM04	
1110202	ELDAD 04 [message]		ELDAD via COM04	
1110203	ELDAD 04 [message]		ELDAD via COM04	
1110204	ELDAD 04 [message]		ELDAD via COM04	
1110209	GUARDING ELDAD COM04	Eldad_guarding.wav	Guarding : eKernel generates an	
1110307	NIRA [message]		Prefix "NIRA COM05" & messag	
1170101	[message]	freeMessage.wav	VRIJE SCRIPT BOODSCAHP v	
1170102	[message]	freeUrgentMessage.wav	VRIJE SCRIPT BOODSCAHP v	
1170103	[message]	Rampenplan.wav	Rampenplan from WEB => relati	
2110101	[message]	Eldad10101.wav	Eldad : alarm 1	
2110102	[message]	Eldad10102.wav	Eldad : alarm 2	
2110107	[message]	Eldad_guarding.wav	Guarding : eKernel generates an	
2110212	[message]	Televic_guarding.wav	Guarding : eKernel generates an	
2110501	TECHN [message]		Prefix "TECHN" zetten voor tecl	
2110502	BRAND [message]		Prefix "BRAND" zetten voor bra	
2110601	VSK [message] FIRE			
2110602	VSK [message] SYSTEM			
2110701	AA [message]		Ala_id 11 van BEMAC alarm typ	
2110702	AI [message]		Ala_id 12 van BEMAC alarm typ	

Figure 108: eVBVOICE Outbound calls

Note:

The keystroke you enter to ensure messages are delivered does not confirm an alarm. If you want to confirm an alarm, you must call the eVBVOICE application and enter your pincode to confirm the active alarms.

Important:

It is very important that you link all possible alarms with a wave file in the eKernel_Message_format table. If there is an alarm which must be sent to a eVBVOICE device, a wave file related to this alarm id must be defined in the eKernel_Message_Format table. If you do not define the wave file, the alarm is not processed.

If the wave file defined in the eKernel_message_format table does not exist, the eVBVOICE application always returns a ‘NACK^NOWAV’ return code to the eKernel application. A ‘NACK^NOWAV’ return code means that the end user never receives the alarm or alarms. After x retries (see DEV_Retry_count_ALT_DEV_id_n field in the eKernel_device table), the alarm is reset or set for an alternative device (only if DEV_Retry_count_ALT_DEV_id_n > 0).

Note: Remark : in the current release you cannot configure alternative devices with an outputpgm (ALT_Alt_OUTPGM_Appl_str in eKErnel_device _alt) for eVBVOICE devices (ALT_OUTPGM_Appl_str = eVBVOICE), because the alternative device otherwise receives the name of the wave file specified in the eKernel_Message_format table, instead of the original message.

VBV4.INI.File Settings

The vbv4.ini file is by default located in the c:\winnt directory.

Note:

Ini Settings use both upper and lower case for readability. However the entry in the file is not case-sensitive.

Voicecard-related settings are documented in the Hardware Installation manual, and in the online help.

[Conference]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
MVIPBridging	See Conference control	0/1	1
RestartForSecondMember		0/1	1
Timer			

Figure 109: Ini settings - Conference

[DataFind]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
DoODBCInMainThread	set to 0 for multi-threaded ODBC drivers	0/1	1

Figure 110: Ini settings - DataFind

[Dialogic]

Voicecard-related settings are documented in the Hardware Installation manual and in the online help.

[Directories]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
Voice	default directory for voice files		C:\VBV4\
Logs	default directory for log files		C:\VBV4\LOGS\
Help	default directory for help files		C:\VBV4\HELP

Figure 111: Ini settings - Directories

[Languages]

[Languages] controls which rules are used for each language. Settings are made by adding any combination of flag values; for example English = 0x1 + 0x20000 =0x20001. The following are the flags available.

- english 0x1
- usegender 0x4

- useordinalforfirstofmonthonly 0x100
- singularpluralhundred 0x102 - special version of "hundred" singular hundred and plural hundreds
- useitaliangender 0x40 - always use male one if saying 81 or 101 or something
- usethouforyear 0x80 - french say one thou, nine hundred, and so on, for year
- useofyear 0x200 - spanish say of between month and year
- dontuseordinalsfordays 0x400 - spanish
- usehundredgender 0x800 - not used
- usehoursandminutes 0x1000 - spanish - say 12 and 30 for 12.30
- fullordinal 0x2000
- usegenderhundreds 0x4000 - spanish - only works if full or fullordinal, or full and fullordinal
- usegender_twentyonethirtyone 0x8000 - spanish
- femalemoney 0x10000
- sayzeroinminslessthan10 0x20000
- japaneseflag 0x40000
- polishflag 0x80000

INI SETTING	DESCRIPTION	UNITS	DEFAULT
EnglishFormat	see description and tables below		
FrenchFormat			
GermanFormat			
ItalianFormat			
SpanishCFormat			
SpanishSAFormat			

Figure 112: Ini settings - Languages

englishformat	english + sayzeroinminslessthan10
italianformat	usegender + singularpluralhundred + useitaliangender + usethouforyear + useordinalforfirstofmonthonly + sayzeroinminslessthan10
frenchformat	singularpluralhundred + usethouforyear + useordinalforfirstofmonthonly + sayzeroinminslessthan10
spanishCformat	usegender + usehundredgender + usethouforyear + useordinalforfirstofmonthonly + useofyear + singularpluralhundred +usegenderhundreds + usegender_twentyonethirtyone + femalemoney + dontuseordinalsfordays
spanishSAformat	usegender + usehundredgender + usethouforyear + useordinalforfirstofmonthonly + useofyear + singularpluralhundred +usegenderhundreds + usegender_twentyonethirtyone + dontuseordinalsfordays
germanformat	singularpluralhundred
japaneseformat	japaneseflag
polishformat	polishflag

Figure 113: Languages - Default settings

[Layout]

A value of 0 means do not show the name/type. A value of 1 means show the name/type.

INI SETTING	DESCRIPTION	UNITS	DEFAULT
ShowNames	sets how controls are drawn		1
ShowTypes	sets how controls are drawn		0

Figure 114: Ini settings - Layout

[Logs]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
LocalSlot	controls which machine receives logging information. Format is \\\\.\\mailslot\\vbvlog machinename		
RemoteSlot	not used		

Figure 115: Ini settings - Logs

[PBX]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
AnswerDeglitch	answer size detect	ms	600
Connect			!,
DetectAnswerTime	silence in answer before stopping	ms	6000
DetectDialTone	duration of nonsilence to detect as dialtone during play operations. 0 to disable.	seconds	10
MaxDialToneWait	secs to wait for dialtone before abandoning	seconds	6
MinDialTone	minimum period of dialtone required to start dialing	seconds	2
Putonhold	string to put caller on hold		!,
ReconnectFromHold	dial string to reconnect caller		!,
ReconnectFromBusyNoans	dial string to reconnect caller from transfer-busy		!,
Xfer	dial string to initiate transfer		!,

Figure 116: Ini settings - PBX

[PlayMsgs]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
FwdDigitsVariable	set to 1 to allow variable number of mailbox digits (digit collection terminates on # or timeout)		0
FwdDigitsLen	number of digits when requesting mailbox number to forward to		3

Figure 117: Ini settings - PlayMsgs

[Record]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
RecChopSecs	deletes data from the end of recorded messages to delete tones	ms	0 (Dialogic) 100 (Tapi) 500 (Rhetorex,)
RecChopSil	deletes silence from end of recorded messages when ending in silence	0 / 1	0
RecChopDialTone	deletes dialtone from end of recorded messages when ending in dialtone	0/1	0
IgnoreTermDigit	if 1, will discard digit that terminates a recording, if 0, will leave digit in buffer for option digit processing	0/1	1

Figure 118: Ini settings - Record

[Rhetorex]

Refer to VBV4.INI Hardware-specific settings on page 116.

[SAPI_TTS]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
ConvertPCMToULaw	used for cards that do not support 8 bit 8Khz PCM when used with TTS engines (Dialogic only	0/1	0
TTSLines	Maximum number of lines to open for TTS - overrides licensed number		1
uLawTTS	Sets the voice format mode for SAPI engines. 1 if the engine provides u-law format	0/1	1 (Lucent engine) 0 (Watson engine)

Figure 119: Ini settings - SAPI_TTS

[SAPI_ASR]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
ASRLines	Maximum number of lines to open for ASR - overrides licensed number		1

Figure 120: Ini settings - SAPI_ASR

[System]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
DeleteOldLogs	Sets the number of logfiles to keep. Files older than the specified number of days are deleted at midnight.	days	0
LineStat	sets the application that contains the default LineStatus window - used in StopSystem		c:\vbv4\vbvlog.exe
Logs	set to 1 to enable event logging in your .exe		0
IgnorePhraseErrors	prevents call drop on phrase error		0
OffHookIdle	busy out lines on shutdown		
PlayEntryAfterError	changes error handling to stop play of Entry Greeting after Invalid or Silence greetings		1
SecureLog	set to 1 to flush event logs after each write.		0

Figure 121: Ini settings - System

[VoiceCard]

VBV4.INI Hardware-specific settings on page 116

INI SETTING	DESCRIPTION	UNITS	DEFAULT
ConfVolDownDigit	sets a digit to decrease listener volume	0-9,#,*	-
ConfVolResetDigit	sets a digit to reset listener volume	0-9,#,*	-
ConfVolUpDigit	sets a digit to increase listener volume	0-9,#,*	-
DefaultFormat	default voice format	FileFormatConstants	0 (8K-ulaw)

Figure 122: Ini settings - VoiceCard

DIDDigitDelay	max silence while waiting for DID digits	seconds	2
DisableLCDetect	disables detection of loop current drop	0/1	0
Flash_character	the character to be used in dial strings for a hook flash		!
Flashtime	duration of a hook flash	ms	500
Ground_character	character used to signal an earth recall		G
LoopCurrentDropTime	duration of loop current drop	ms	Voice card dependent
MaxDialToneWait	maximum time to wait for dialtone	seconds	6
MinDialTone	minimum duration of dialtone	seconds	2
PauseTime	duration for a pause character (comma) during dialing	ms	2000
Ring_cnt_reset	minimum time between calls	seconds	10000
Type	selects the voicecard type in use	one of: RHETOREX DIALOGIC DIALOGIC_ISDN ACULAB_RHET ACULAB_RHET_DPN TAPI SIMULATOR NONE (default)	
UseAlaw	use A-law compression instead of u-law for sound card and Tapi cards	0/1	0
Voxconvert	if set to 1, uses format conversion by VBVoice instead of Windows ACM when playing to sound card or tapi	0/1	0

VBV4.INI Hardware-specific settings

Note:

Ini settings use both upper and lower case for readability; however, the entry in the file is not case-sensitive.

[AccuCall]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
DefaultTable	path of tone table to be loaded by Accuload, and used by Brooktrout / Rhetorex driver	path name	c:\rhet32\acucal32\default.ton

Figure 123: Ini settings - AccuCall

[Dialogic]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
ConfEntryTone	enables a tone to all conferees upon entry of another conferee	0/1	1
ConfTariffTone	enables a tone every 5 minutes to all conferees	0/1	0
InboundProtocol	sets GlobalCall protocol, for use with Type = DIALOGIC_ISDN	ISDN	no default
ListenMonitor	listeners in a DCB conference share the same monitor port	0/1	0
NoAnswer	# of seconds of ringback before "no answer" returned. This setting is overridden by any loaded TSF file. See Dialogic Call Progress.	seconds	30
OutboundProtocol	sets GlobalCall protocol, for use with Type = DIALOGIC_ISDN	ISDN	no default

Figure 124: Ini settings - Dialogic

[Rhetorex]

INI SETTING	DESCRIPTION	UNITS	DEFAULT
SystemInputGain	increase inbound volume (helps call progress & record		0x1000
PCPMInputGain	increases inbound vol during call progress		3
(see also VoiceCard - ConnectTimeout below)			

Figure 125: Ini settings - Rhetorex

[Voicecard] - (ACULAB specific settings)

INI SETTING	DESCRIPTION	UNITS	DEFAULT
ConnectTimeout	max time for call progress analysis	seconds	40
FlexMapping		0/1	0
MinCLI	number of Originating Number (CLI) digits		0
Rhet_Number	Number of rhetorex cards in use in conjunction with Aculab cards		1
Rhet_Type	used to define card	one of: RDSP24000 (default)	
	type to use in conjunction with Aculab cards	RDSP20000 RDSP16000 RDSP8000 RDSP4000 VANTAGE_VRS_32 VANTAGE_VRS_24 VANTAGE_VRS_16 VANTAGE_VRS_8 VANTAGE_VRS_4	
RingNoAnswerSecs	max number of seconds for ringback	seconds	15
Trace		0/1	
UseCallID	enable/disable CLID	0/1	0

Figure 126: Ini settings - [Voicecard] (ACULAB specific settings)

[Voicecard] - (Dialogic specific settings)

INI SETTING	DESCRIPTION	UNITS	DEFAULT
PulseDetection	Enables pulse detection	0 = no DPD 1 = normal DPD 2 = DPD with cut-through	

INI SETTING	DESCRIPTION	UNITS	DEFAULT
LoopStart	Overrides T1 bit pattern settings (see below) to use loopstart protocol	0/1	0
Wink	Selects number of winks after incoming ring	0 = no wink (immediate) 1 = single wink (default) 2 = double wink	
WinkTime	Time in ms for wink duration	ms	150
PreWinkDelay	Prewink transmit delay in ms	ms	0

Figure 127: Ini settings - [Voicecard] (Dialogic specific settings)

[Voicecard] - (Dialogic T1 specific settings)

T1 bit pattern settings are used to configure the T1 bit patterns for the A & B bits for non-standard T1 configurations. The first digit is the value for 'A' bit, the second is the value for 'B' bit. The values can be "00", "01", "10" or "11", where 00 means look for both A and B bits off, 01 means look for A bit off, B bit on, and so on.

Valid values for t1hookflash_start and t1hookflash_end are as follows:

- 0 = set the bit specified in t1hookflash, clear any others
- 1 = set the bit specified in t1hookflash, leave others unchanged
- 2 = clear the bit specified in t1hookflash, leave others unchanged

The difference between t1answer and t1hangup settings is used to test for hangup notification from the switch. For example, if t1answer is 11 and t1hangup is 10, the B bit going to 0 is used to detect hangup.

The difference between t1ringing and t1idle is used to test for incoming ring indication in the same way. Loopstart overrides the t1answer settings to monitor B-bit changes only, and forces the other settings to use the following values.

- t1idle = 01
- t1ringing = 00
- t1hangup = 00 (no change)
- t1hookflash = 01

INI SETTING	DESCRIPTION	UNITS	DEFAULT	
			Loopstart = 0	LoopStart = 1
t1idle	bits to set for idle line	AB	00	01
t1ringing	bits to detect ring indication	AB	11	00
t1answer	bits to set for IVR to answer call	AB	11	11
t1hangup	bits to detect hangup	AB	00	00
t1hookflash	bits to change for a hookflash	AB	10	10
t1hookflash_start	how to start a hookflash	0/1/2	2	
t1hookflash_end	how to end a hookflash	0/1/2	1	

Figure 128: Ini settings - [Voicecard] (Dialogic T1 specific settings)

VBVoice uses the following events to determine start and end of an incoming call.

	LoopStart=0	LoopStart=1
Ring event	Transition from t1idle to t1ringing	State change on bit B
Hangup event	Transition from t1answer to t1hangup	Transition from t1idle to t1hangup

Figure 129: VBVoice events to start and end an incoming call

[Voicecard] - Rhetorex specific settings

INI SETTING	DESCRIPTION	UNITS	DEFAULT
ConnectTimeout	maximum duration of PCPM on no answer (Rhetorex only)	seconds	40

Figure 130: Ini settings - [Voicecard] (Rhetorex specific settings)

[Voicecard] - (Rhetorex T1 specific settings)

When the first seizure attempt fails, a random back-off is performed followed by another seizure attempt. If this second attempt fails, a failure is reported. Most T1 settings for Brooktrout/ Rhetorex cards are now set in the RealCTdrivers.

INI SETTING	DESCRIPTION	UNITS	DEFAULT
AllDigitalConnect	Identifies if the system is a digital end-to-end connection	0/1	0
T1SeizeBackOffMax	Upper end of random retry back-off for T1 line seizure	ms	T1SeizeTimeout
T1SeizeBackOffMin	Lower end of random retry back-off for T1 line seizure	ms	500
UseATIOnly	Forces ATI use in system with other hardware installed	0/1	0
UseCallID	Enables or disables CallerID	0/1	0
UseT1Only	Forces T1 use in system with other hardware installed	0/1	0

Figure 131: Ini settings - [Voicecard] (Rhetorex T1 specific settings)

From VBVOICE4.4 a Pronexus VbvConfig tool is installed for maintenance of the VBV4.ini file. See the following figure.

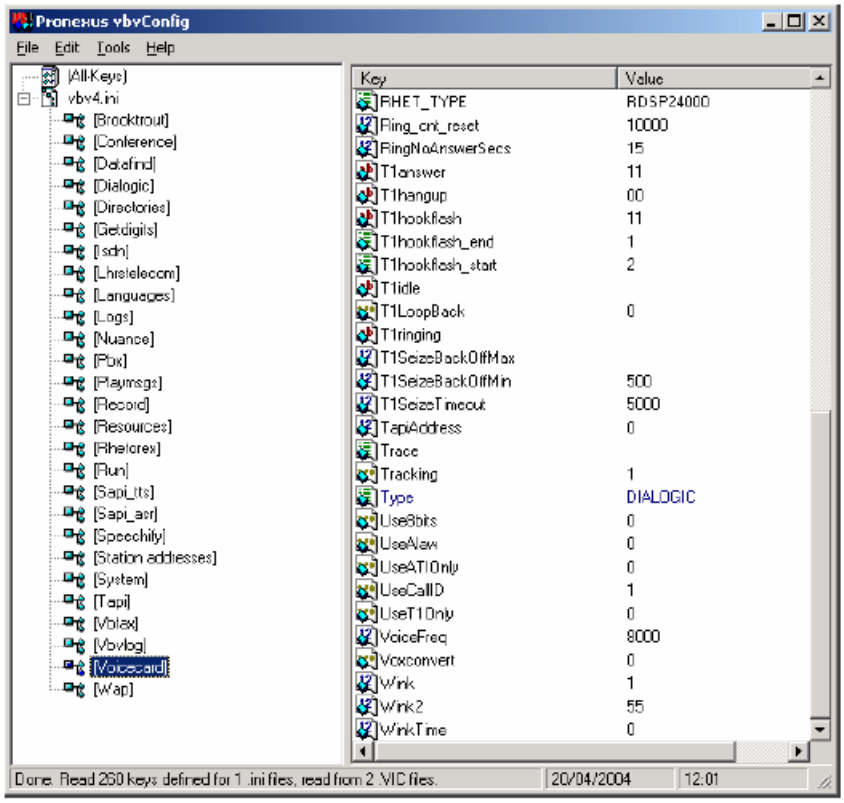


Figure 132: vbvConfig tool - Voicecard

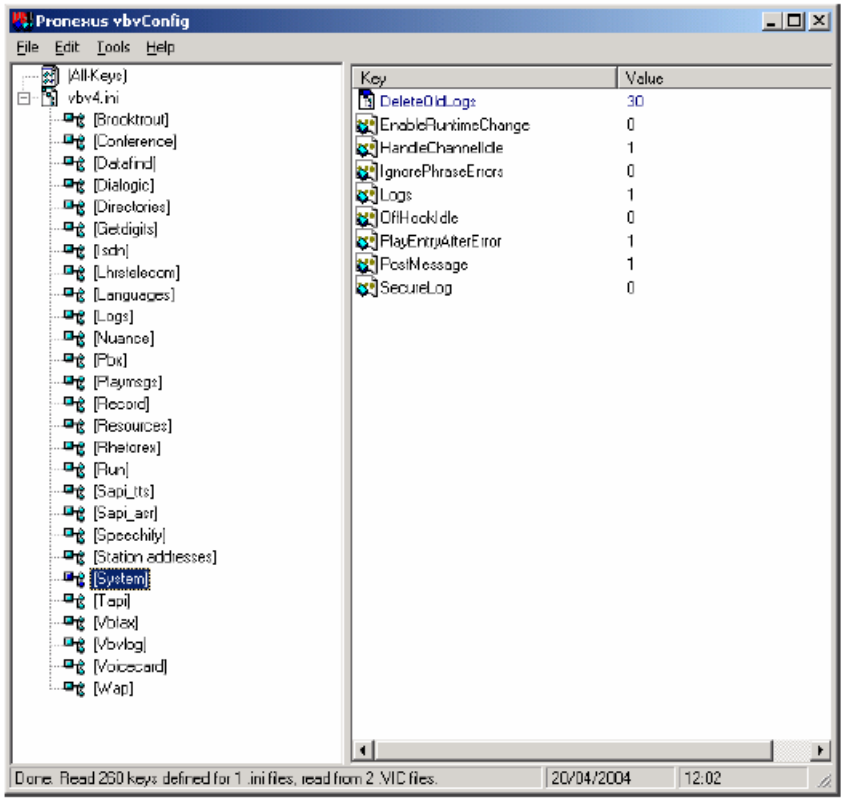


Figure 133: vbvConfig - System

Chapter 7: Module - eVBVOICE AHVR

Important:

Due to the ongoing development of the DECT Messenger product suite, some modules that provide additional functionality may become available after the initial release of DECT Messenger 4.0.

The following modules are described in this document but are not available at initial General Availability.

- eFR
- eLICENSE
- eLOCATION
- eSMS
- eSNMP
- eVBVOICE

The eFR module is an add-on module and is licensed separately through the eLICENSE module. Some of the modules listed in this attention box are available only on a site-specific basis.

Introduction

With the current eVBVoice module, you can generate, confirm, and reset an alarm.

When an alarm is set, a pre-recorded message can be distributed to a number of recipients using Ad Hoc Voice Recording (AHVR).

You can use eVBVOICE AHVR to perform the following actions.

- Generate an alarm
- Record a specific alarm message
- Distribute the ad hoc recorded message

The eVBVoice module operates basically as before the addition of AVHR. You SET an alarm in the same way as before AVHR,, using the 2 level menu structure. However, you can only record a message after you receive the level 2 prompt. The recording stops when you press a dtmf key, or when the connection is broken. The recorded message is then distributed in the same way as pre-recorded messages.

Configuration

In eConfig you define the menu structure. One of the options on level 1 is to ‘SET’ an alarm. See the following figure.

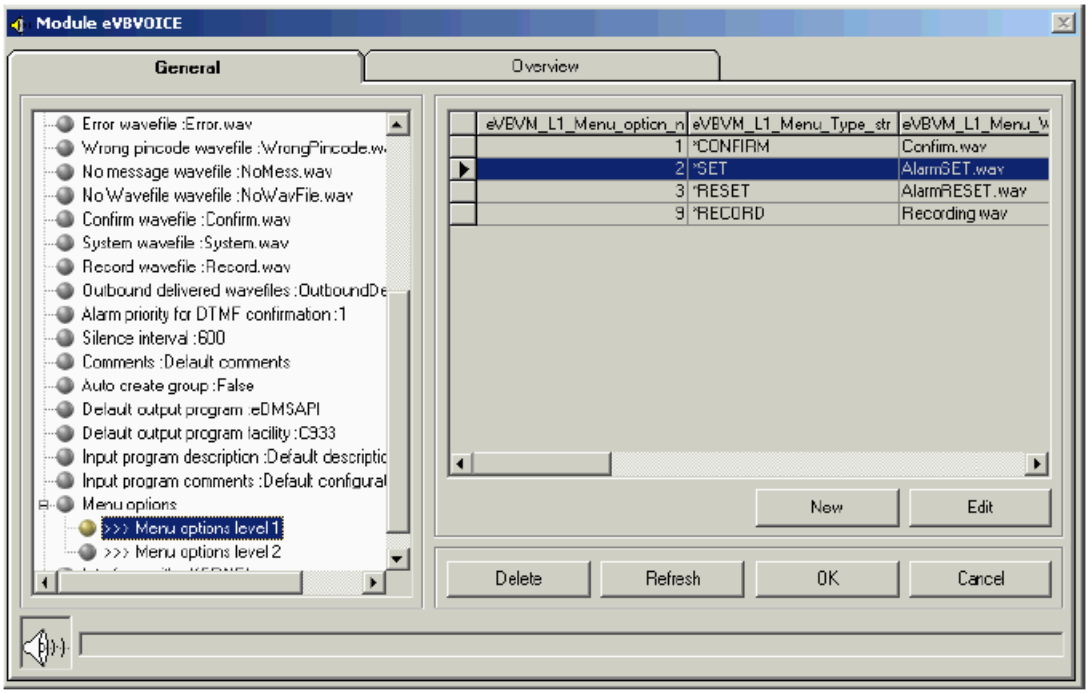


Figure 134: Level 1 - Set alarm

On level 2, you define the alarm ID and the group. See the following figure

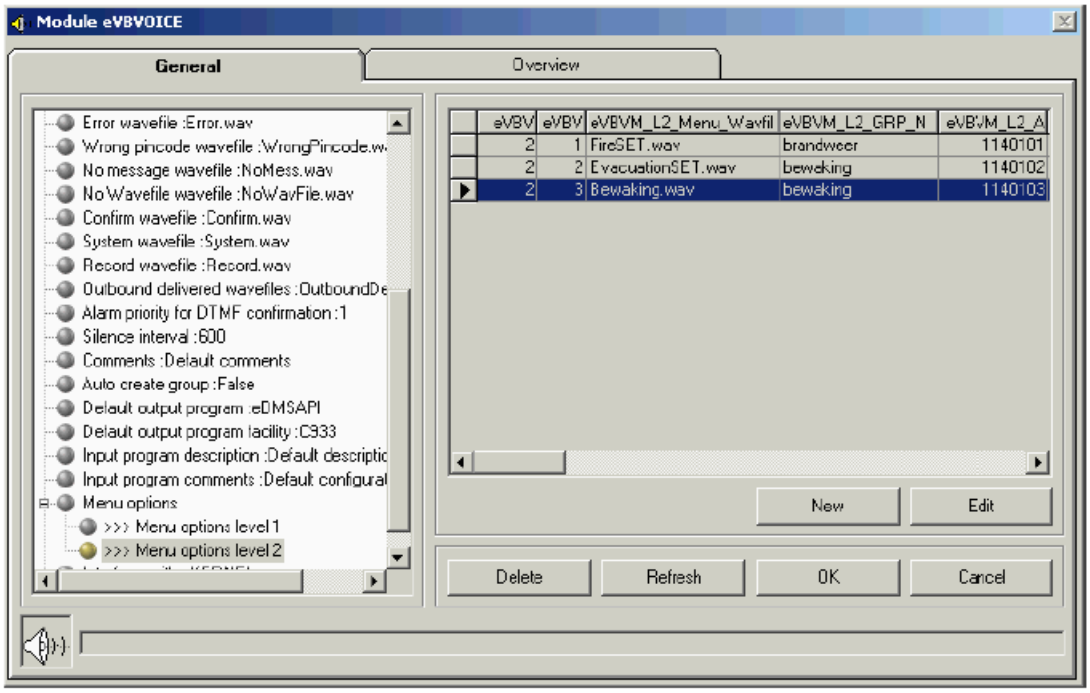


Figure 135: Level 2 - Define alarm ID and group

You need to set the only visible deviation from normal alarm handling. Instead of identifying the prerecorded message, use the keyword *RECORD. See the following figure.

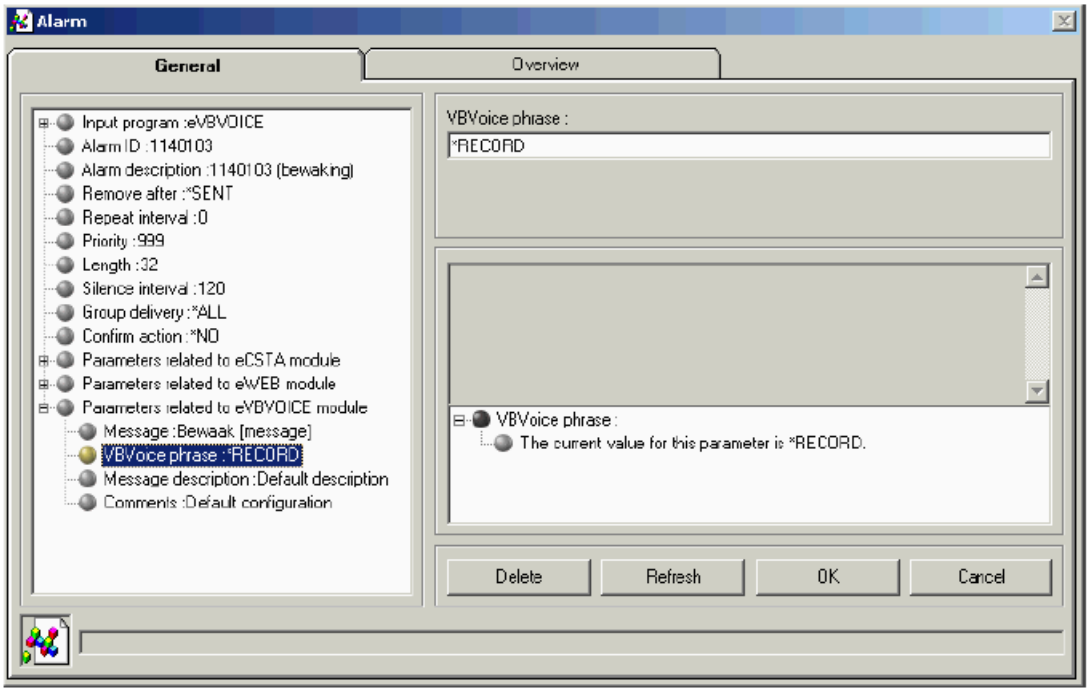


Figure 136: RECORD

The following figure shows how *RECORD looks In the Access tables.

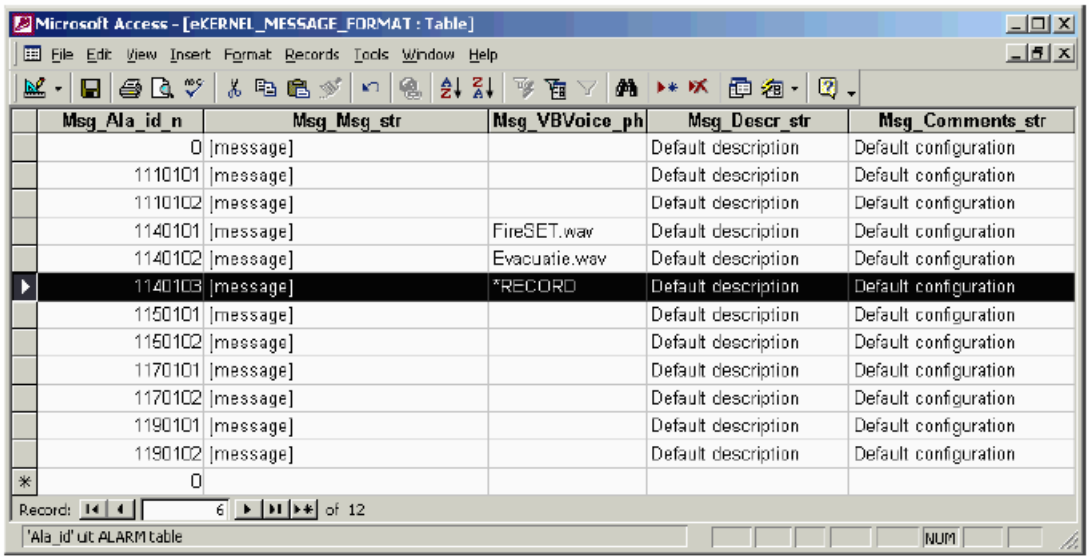


Figure 137: RECORD in the access tables

Example 1

Consider the customer site shown in the following figure.

For the example, assume there are 3 types of alarms: fire, reanimation and security. Also assume that the alarm recipients are located in all of the iS3070 locations.

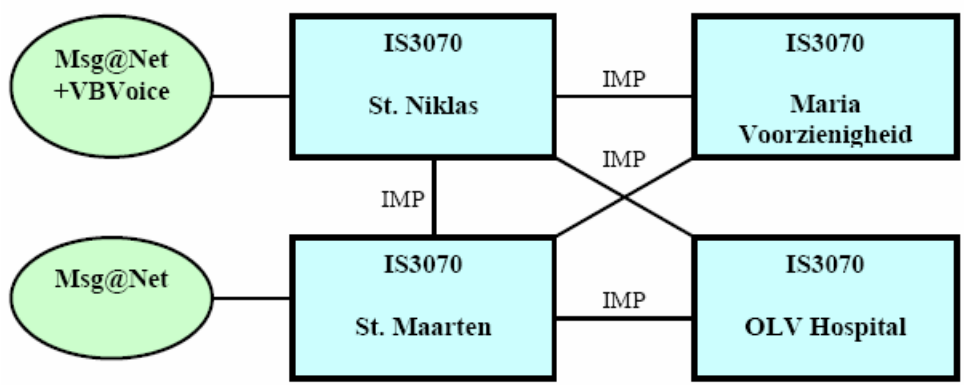


Figure 138: Example 1 - Customer site

Also assume for Example 1 that there is one eVBVoice module on site. Your goal is to add ad hoc voice messages to the reanimation alarms.

The IVR menu could look like the following figure.

Menu level 1	Menu level 2	Alarm	Group	Message
1 Set an alarm	1 Fire alarm	1140101	Fire dep.	FireSet.wav
	2 Reanimation	1140102	Medical	*RECORD
	3 Security	1140103	Security	SecuritySet.wav
2 Confirm an alarm	-			
3 Reset an alarm	1 Fire alarm	1140101	Fire dep.	FireReset.wav

Figure 139: Example 1 - IVR menu

Example 2

for Example 2, assume the location is the same site as in Example 1. In Example 2, however, a reanimation alarm is set for a specific node or group. The IVR menu could look like the one shown in the following figure.

Menu level 1	Menu level 2	Alarm	Group	Message
1 Reanimation alarr	1 st. Niklas	1140102	Med. Niklas	*RECORD
(=Set)	2 st. Maarten	1140102	Med. Maarten	*RECORD
	3 Maria Voorz.	1140102	Med. Maria V	*RECORD
	4 OLV	1140102	Med. OLV	*RECORD
2 Set an alarm	1 Fire alarm	1140101	Fire dep.	FireSet.wav
	2 Security	1140103	Security	SecuritySet.wav
3 Confirm an alarm	-			
4 Reset an alarm	1 Fire alarm	1140101	Fire dep.	FireReset.wav

Figure 140: Example 2 - IVR menu

Voice messages are stored on the system where the eVBVoice module is located. This means that voice messages can only be distributed by the eVBVOICE module on the same system as the voice messages. The voice messages are not automatically removed. For this reason Avaya suggests that you manually remove old and handled message from time to time.

Chapter 8: Module - eWEB

When you start your web browser application and navigate to the DECT Messenger system that has the eWEB module operational, a sign-on window opens. Contact the system administrator to obtain the URL address assigned to the system.

Sign-on procedure

A sign-on is required; because you are not yet authenticated to the application, this window is presented in English.

Figure 141: Sign on information

On the lower left-hand side of the window, user status is displayed, indicating that you are not logged in at this time, similar to the following:

To start working, you must log in with a valid user and password combination. The password is displayed as a series of asterisks (*) during entry.

The user and password is checked against the eWEB_USER_AUTH table. Refer to the documentation for Table: eWEB_USER_AUTH on page 393 for more information.

When a valid user and password is found, you are able to continue working in the eWEB module.

It is important to know that during the sign on procedure, two additional parameters are fetched: the language code and the security level.

The language code determines the language of the forms that are presented to the user. If for example the language code is 2909 – Belgian English, the panels are in English. If the language code is 2963 – Belgian Dutch, the panels are in Dutch.

The security level determines the table-of-contents options that are presented to the user. A user with a limited security level has only a small number of options available, whereas a user with a high security level has many options available. Refer to the documentation of Table: eWEB_TOC on page 387 for more information on the table-of-contents mechanism.

In the illustrations on the following pages, the user shown has a language code that refers to English forms, and a security level that gives access to all available options. The information displayed varies depending on your security level and language code.

Sign-off procedure

To log off, choose the last option in the list on the left side of the window, Sign off. You are also automatically logged off when either of the following occurs:

- Twelve hours elapse after initial sign-on.
- You leave the eWEB web site, for instance by selecting another URL in the address field of your browser or selecting another web site through Favorites, Home, Back, and so on.

Figure 142: Expired login on page 131 shows an expired state, requiring a new logon.

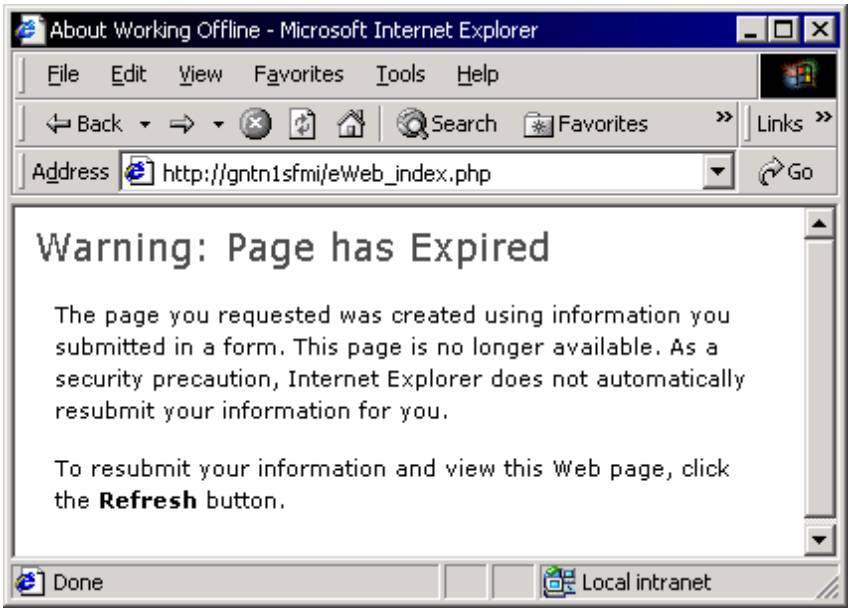


Figure 142: Expired login

Send DMS-API Message

This function allows you to send an E2-data message to a peripheral that is capable of receiving messages through this technology. The web interface presents all DECT extensions that are defined in the eKERNEL_DEVICE table for the local site and area and the output program eDMSAPI.

Important:

The eWEB application is configured in the eWEB table, and identifies its site and area based upon the IP address of the Apache Web Server.

Therefore, it lists only those devices that are defined for that same site and area. In a multi-area environment, you can access the devices that belong to another area. You can assign these remote area devices on device level in the eKERNEL_DEVICE table, where the DEV_Ras_Area_b value must be set to **True**.

Figure 143: Local extensions on page 132 shows the list of extensions that all reside locally on the same site and area.

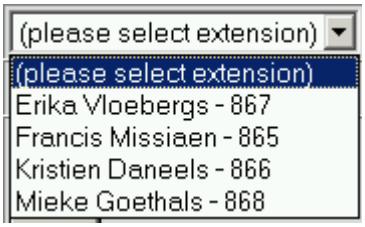


Figure 143: Local extensions

Figure 144: Local and remote extensions on page 132 shows the list of extensions that all reside locally on the same site and area, but also displays an extension that resides on another area, which is made available through the DEC_Ras_Area_b value in the eKERNEL_DEVICE table.

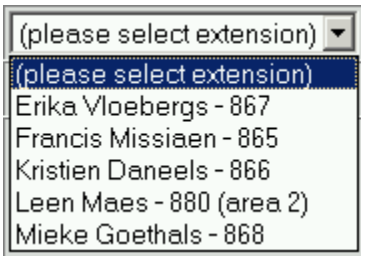


Figure 144: Local and remote extensions

Note:

The Send DMS-API Message form always contacts that local DMS-API Service of the same site and area as the Apache Web Server. In a multi-area environment, where there are possible multiple eDMSAPI applications defined, the local DMS-API service contacts all peripherals.

The user can enter a message and the message type (normal or urgent) and click **Enter** to transmit the message. The application waits for message delivery or failure. In the case of urgent messages, this delay can sometimes be quite long because the application waits for the user to acknowledge receipt of the message by pressing **OK** on the DECT handset.

Send SMTP Message

This function allows the user to send a message to a mail address destination, by means of an SMTP connection between the Apache Web Server and the SMTP server of the mail server. In this process, no eKERNEL activity takes place, because the transaction is executed directly.

The list of available addresses is limited to the devices that are defined in eKERNEL_DEVICE table, and defined for the same site and area as the eWEB application, and with output program eSMTP.

You can also make devices that are allocated to a remote area available through the DEV_Ras_Area_b value in the eKERNEL_DEVICE table. An example is shown in Figure 145: Sending messages to remote addresses on page 133:

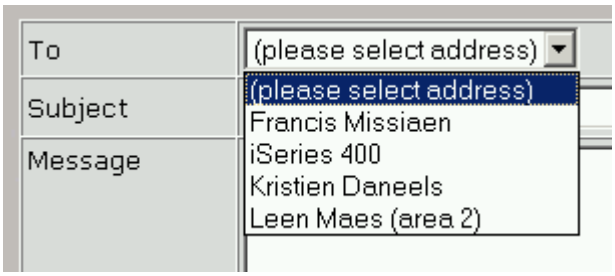


Figure 145: Sending messages to remote addresses

As a result, the SMTP server is contacted, and a message is sent. The IP address and port number is retrieved for the server defined in the eSMTP_CLIENT table, with a matching site and area as used by the Apache Web Server.

The mail is sent following the specs of RFC821. In the composed mail, the MAIL FROM: keyword is automatically retrieved from the definition in the eWEB_USER_AUTH table. As a result, when the destination user replies to the mail, the reply arrives in the correct mailbox of the sender.

Send Server Message

Send Server Message is a function that communicates to the eKERNEL module.

This is the opposite of the Send DMS-API Message and Send SMTP Message, both of which directly access the underlying services and ignore the eKERNEL module for processing. A major advantage of using Send Server Message is that it utilizes more product features, including: logging, sending to a group of users, assigning alarm types, priorities, addressing any kind of peripherals, implementing confirmation procedures, implementing alternatives devices, and so on.

Because Send Server Message communicates with eKERNEL, a number of configuration actions are required. One of them is specifying alarm identifiers in the eKERNEL_ALARM table, for the input program that is assigned to the eWEB instance. At this time, you can define for instance alarm types with different lengths (for example, short messages of 8 bytes, medium messages of 16 bytes and long messages of 32 bytes).

Important:

Because the Send Server Message is designed only to set a message, and cannot reset a message, you must always specify remove after *SENT in the eKERNEL_ALARM table, otherwise the message remains active forever.

In the example shown in Figure 146: Alarm types on page 134, you can choose between three alarm types, which are defined in the eKERNEL_ALARM table.

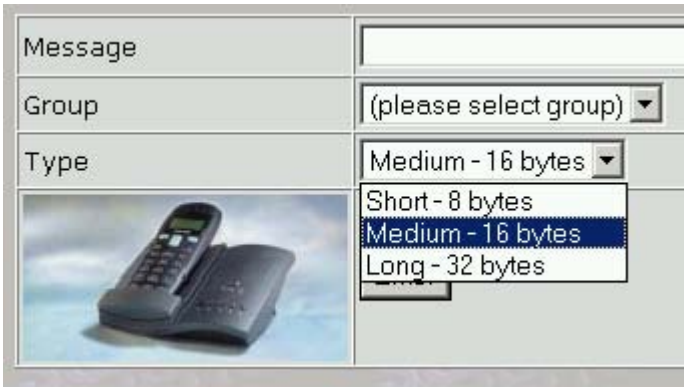


Figure 146: Alarm types

Important:

You can only access alarm types in the eKERNEL_ALARM table with field ALA_Trace_b equal to False. While assigning alarm types, always make a distinction between alarms for Send Server Message (False) and Send Script Message (True).

The destination of the message is also defined in the database. The eWEB module has an input program identifier, and one or more alarm definitions. For the same input program, you also must predefine the group, group members and group authorities in the corresponding tables eKERNEL_GROUP, eKERNEL_GROUP_MEMBER and eKERNEL_GROUP_AUTH.

The web user is able only to select from the list of groups that are configured for that input program.

Note:

The web user can submit a message to the eKERNEL, but is not able to verify that the message actually arrives at the destination address. One potential issue is that a message can be sent to a group that is empty (it has no peripherals defined as group members). Another issue can arise if a group is configured in such way that, due to the definitions in eKERNEL_GROUP_MEMBER, no one is active in the group, based upon hour, day, holiday and activation interval issues. eWEB users must be aware of these possibilities.

In the sample shown in Figure 147: Group list on page 135 (it is advisable to use more descriptive group names than those shown in the example).

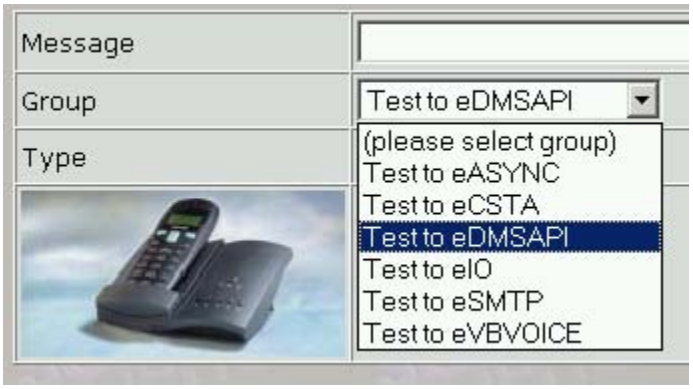


Figure 147: Group list

Send Group Message

In step 1, shown in Figure 148: Select a group on page 135, you can choose from a list of groups. These groups are retrieved from the eKERNEL_GROUP and eKERNEL_GROUP_MEMBER tables. All the groups that use a matching input program with the appropriate eWEB module (site/area) are shown to the user. Collapse or expand the group to see the group members.

Click the arrow to select the group.

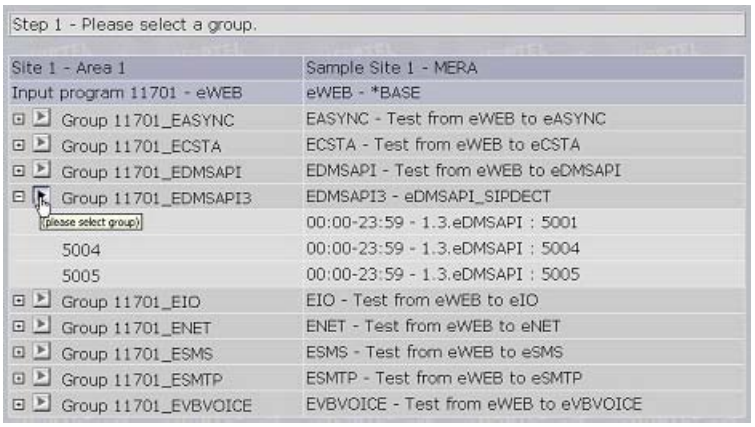


Figure 148: Select a group

The next step offers an overview of the group messages that are preconfigured for the selected group. As shown in Figure 149: Select a message on page 136, the eWEB_SNDGRPMSG table can define private messages per group, shared messages for all groups and also user messages.

In the example shown in Figure 149: Select a message on page 136, the administrator has configured four private messages, one fixed message and one user-specified message:

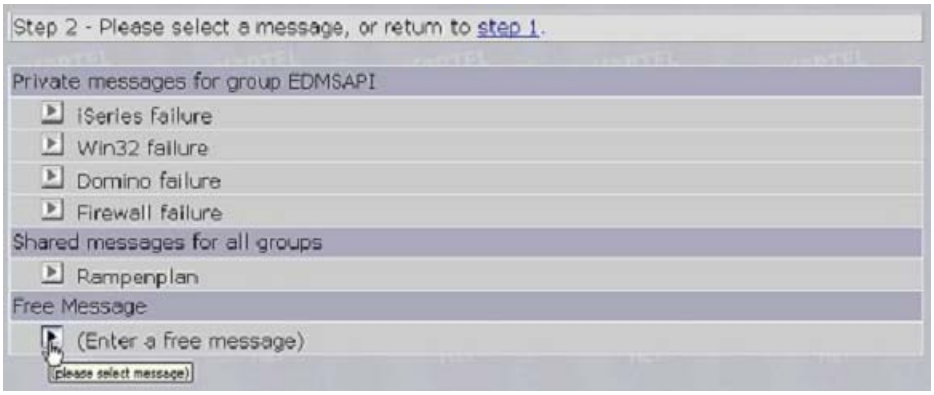


Figure 149: Select a message

Finally, you can send the request to eKERNEL and submit the request for further processing. The example shown in Figure 150: Confirm and send message on page 136 shows a situation in which a user-defined message has been selected, so you must enter the message text manually.

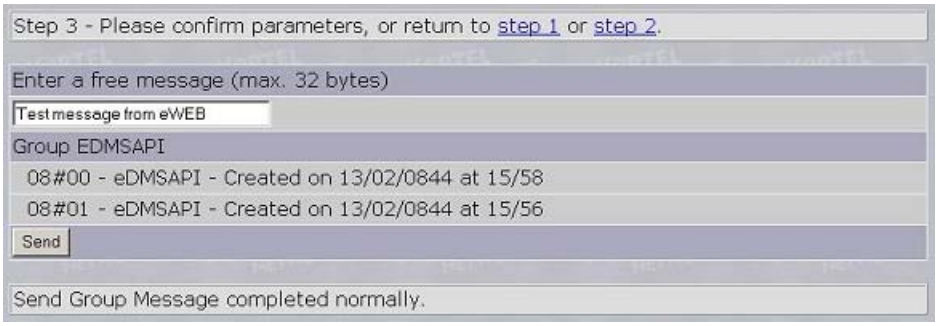


Figure 150: Confirm and send message

The Send Group Message completed normally message indicates the message has been submitted to eKERNEL. Final message delivery depends on a number of factors and are beyond control of the eWEB user.

Send User Message

In step 1, a list of groups is presented, as shown in Figure 151: Select the group on page 137. These groups are retrieved from the eKERNEL_GROUP and eKERNEL_GROUP_MEMBER tables. All the groups that use a matching input program with the appropriate eWEB module (site/area) are shown to the user. Collapse or expand the group to see the group members.

Click the arrow to select the group.

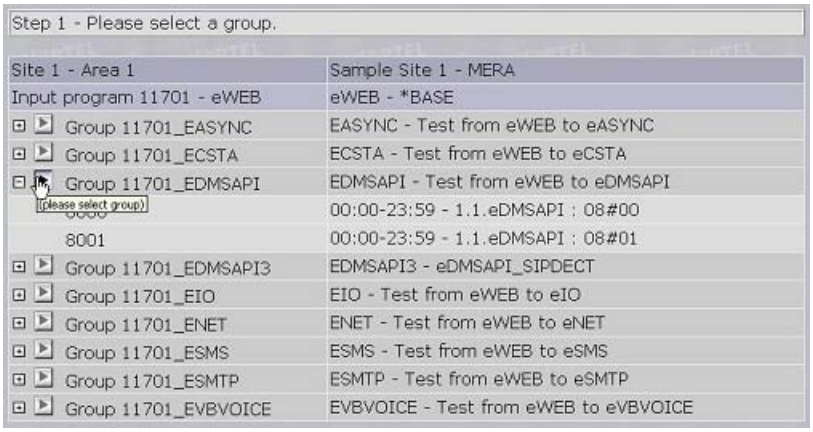


Figure 151: Select the group

Step 2 provides an overview of the user messages that are preconfigured for your current user profiles, which is used during the login procedure in the initial window of eWEB. As shown in Figure 152: Select a message on page 137, the eWEB_SNDUSRMSG table can define private messages per user, shared messages for all users, or user-defined entered messages.

In the example shown in Figure 152: Select a message on page 137, the administrator has configured four private messages, six fixed messages, and a user-defined message.

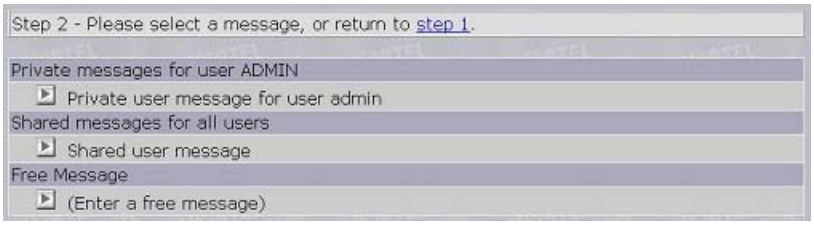


Figure 152: Select a message

Finally you can send the request to eKERNEL and submit the request for further processing. Note that the example in Figure 153: Confirm your choices on page 137 shows a fixed message and therefore message text need not be entered.

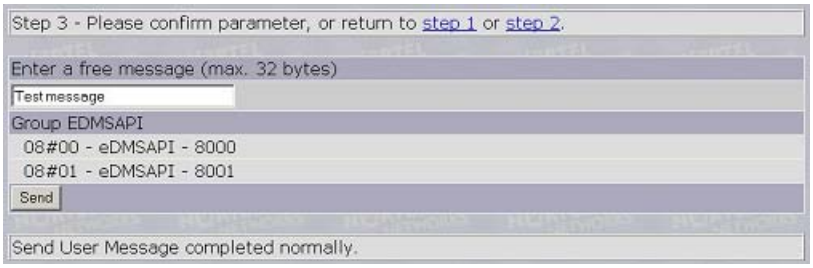
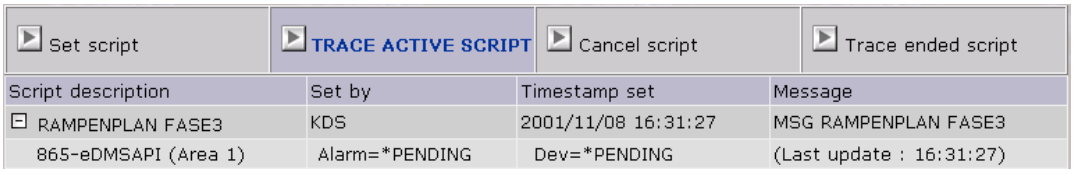


Figure 153: Confirm your choices

When the message is submitted to eKERNEL, the message "Send User Message completed normally" is displayed. Final message delivery depends on a number of factors and are beyond control of the eWEB user.

Send Script Message

You can choose from the following sub-functions:



- **Set Script** is used to activate a script. The scripts are defined the eWEB_SCRIPT table.
- **Trace Active Script** is used to see an overview of activated scripts. These scripts are still running.
- **Cancel Script** is used to abort a script that has been activated.
- **Trace ended Script** is used to see an overview of these scripts that are completed.

For more information, refer to:

- Table: eWEB_SCRIPT on page 369
- Table: eWEB_SCRIPT_SET_AUTH on page 373
- Table: eWEB_SCRIPT_TRACE_AUTH on page 375
- Table: eWEB_SCRIPT_CANCEL_AUTH on page 377

Set Script

Choose Set Script to browse an overview of defined scripts, as shown in Figure 154: Overview of defined scripts on page 139. A green or red icon indicates if the eWEB user is authorized to activate the script. The window also shows additional information; as follows:

- The identifier of the group.
- The message that is sent to the group members.
- The current number of instances of the script currently active.
- The maximum number of instances of the script that can be active.

The illustration in Figure 154: Overview of defined scripts on page 139 shows that the current user is authorized to set the first seven scripts, but not authorized for the last script. No script is currently active.

Script description	Group	Message	Cur active	Max active
RAMPENPLAN FASE1	eDMSAPI	MSG RAMPENPLAN FASE1	0	10
RAMPENPLAN FASE2	eDMSAPI	MSG RAMPENPLAN FASE2	0	1
RAMPENPLAN FASE3	eDMSAPI	MSG RAMPENPLAN FASE3	0	1
RAMPENPLAN FREE GROUP	*ALL	MSG RAMPENPLAN FREE GROUP	0	1
Short script	*ALL	This is short	0	*NOMAX
Medium script	*ALL	*FREE	0	*NOMAX
Long script	*ALL	*FREE	0	*NOMAX
MSG NEED RESET	eDMSAPI	MSG NEED RESET	0	*NOMAX

Figure 154: Overview of defined scripts

In Figure 155: Script details on page 139, the third script has been activated, and more detailed information on the script is provided (only one such script can be activated at a time). The window shows us that one device is a member of the group, and the device is configured to be available 24/24 hours and 7/7 days. A minimum of one device must confirm the alarm, therefore you must not clear the device selection.

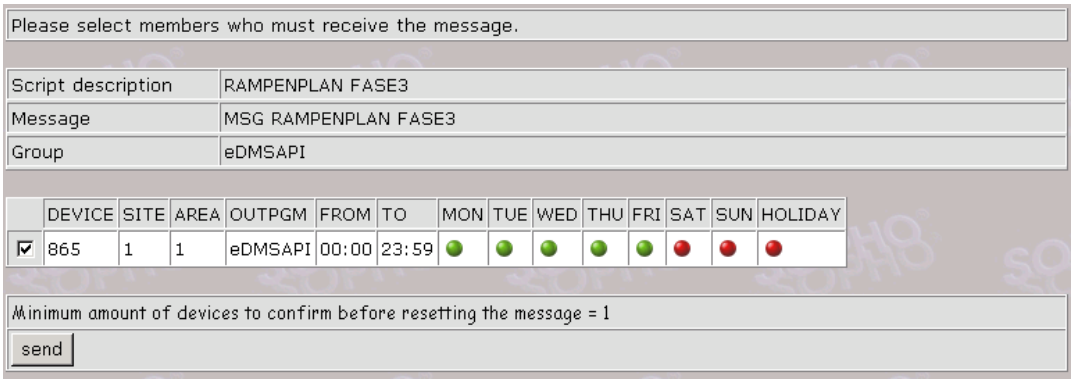


Figure 155: Script details

Trace Active Script

Use Trace Active Script, shown in Figure 156: Trace active script on page 139, to monitor the event handling of scripts that are active.

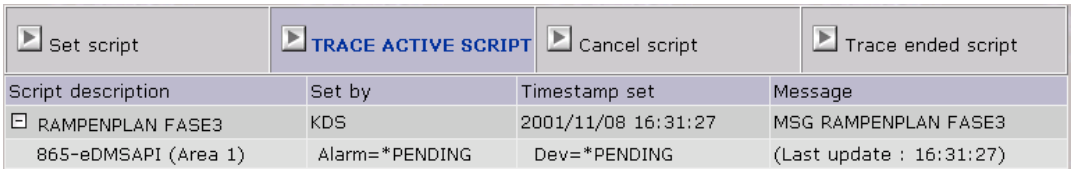


Figure 156: Trace active script

Cancel Script

Use Cancel Script to abort an active script. Figure 157: Cancel script on page 140 shows one active script.

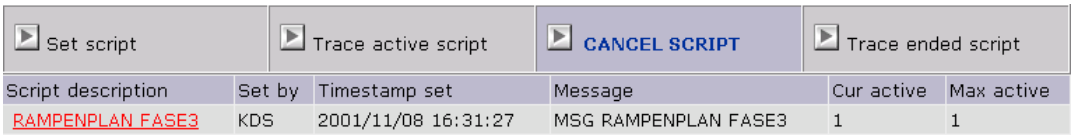


Figure 157: Cancel script

Cancelled scripts are removed from the list, as shown in Figure 158: Cancelled script removed from the list on page 140.

Figure 158: Cancelled script removed from the list

Trace Ended Script

Trace Ended Script, shown in Figure 159: Trace Ended Script on page 140, allows you to monitor the event handling of scripts that are finished.

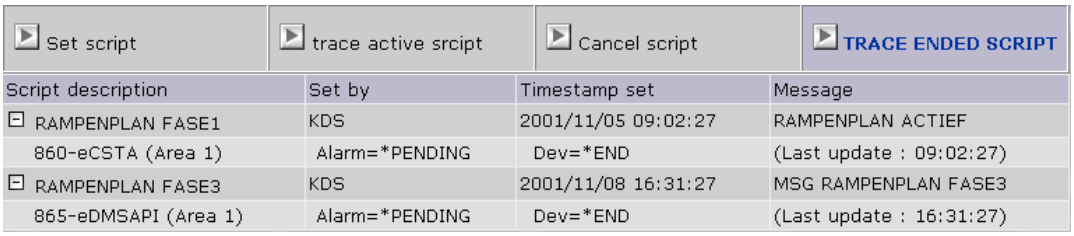


Figure 159: Trace Ended Script

Alarm Inquiry

Alarm Inquiry allows you to see all relevant parameters for the eKERNEL_ALARM file. Only those records are shown that refer to the site where the current eWEB instance resides.

The information is retrieved from two tables: eKERNEL_ALARM and eKERNEL_INPGM. You can either display data for all input programs (by specifying *ALL) or select one input program.

Device Inquiry

The device inquiry allows you to see all relevant parameters for the eKERNEL_DEVICE file. Only those records are shown that refer to the site where the current eWEB instance resides.

The information is retrieved from one table: eKERNEL_DEVICE. You can either display data for all output programs (by specifying *ALL) or select one output program.

Group Inquiry

The group inquiry allows you to see all relevant parameters for the eKERNEL_GROUP and eKERNEL_GROUP_MEMBER files. Only those records are shown that refer to the site where the current eWEB instance resides.

The information is retrieved from multiple tables: eKERNEL_GROUP, eKERNEL_GROUP_MEMBER, eKERNEL_INPGM, eKERNEL_DEVICE, eKERNEL_SITE and eKERNEL_AREA. You can select the data for each area.

Table View

The Table View function allows you to perform inquiry functions of all tables available in Messenger_CFG database. Only users with security administrator special authority rights can access the eWEB_USER_AUTH file. Users who lack security administrator special authority rights cannot access this table, which contains sensitive information such as passwords.

Work with Groups

Click **Work with Groups** to access group maintenance functions. Users with all object special authority can access all groups, while users without these rights can access only groups specified in eKERNEL_GROUP_AUTH.

Note:

If no groups are shown, the user has no all object special authority, or no access to any group. You must grant if necessary access to one or more groups through the eKERNEL_GROUP_AUTH table.

In step 1, shown in Figure 160: Select a group on page 142, select a group. You can collapse or expand a group to preview the group member information.

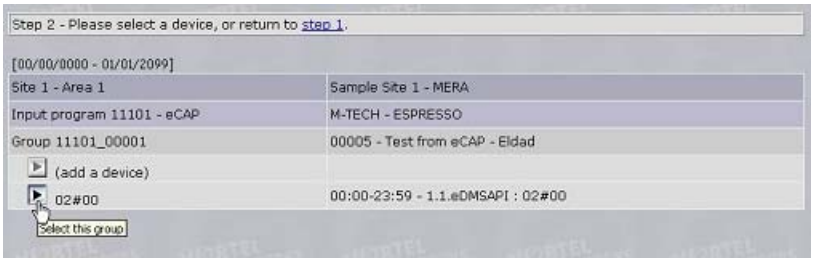


Figure 160: Select a group

Next, you can either maintain an existing device or add a new device. The example shown in Figure 161: Select a device on page 142 demonstrates selecting an existing device for maintenance (update or delete).

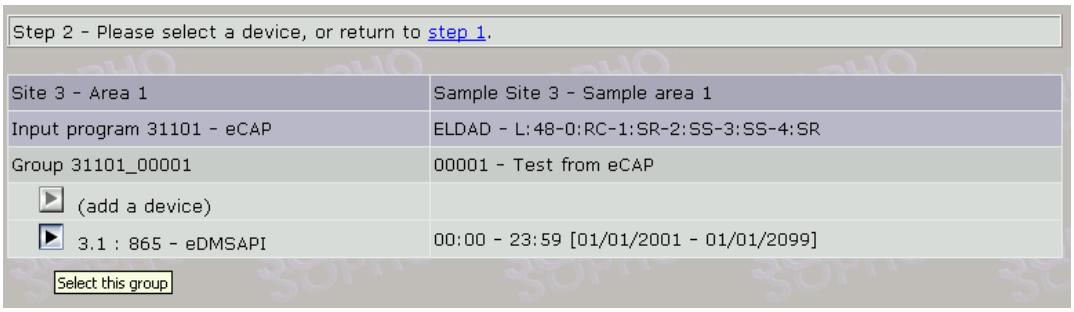


Figure 161: Select a device

The values displayed when you choose Work with Groups refer to the fields in the eKERNEL_GROUP_MEMBER table. Refer to Table: eKERNEL_GROUP_MEMBER on page 303 for details. The example shown in Figure 162: Confirm changes on page 143 defines the extension 865 to be available on working days only between 8:30 and 12:00. Note that the record is disabled on Saturdays, Sundays and holidays.

Important:

The last two fields (Activate definition and Deactivate definition) allow you to specify an interval during which the record is active. In the example shown in Figure 162: Confirm changes on page 143, the record is active from January 1, 2001 at 00:00 until December 31, 23:59. This functionality allows administrators and power users with group maintenance rights to predefine schedules that are activated and deactivated automatically. This functionality can add flexibility in your group maintenance in handling holiday planning, staff schedules, and so on.

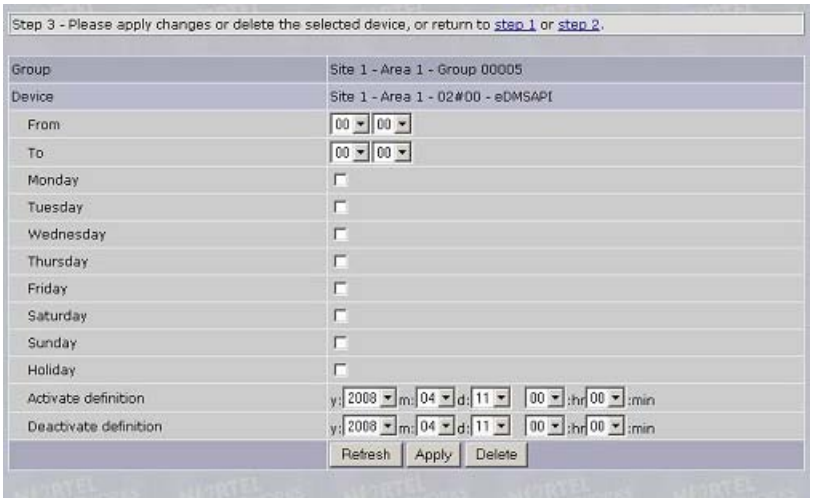


Figure 162: Confirm changes

If you select to add a new device, a window similar to the one in Figure 163: Select new device on page 143 is shown. Select one of the configured devices and specify the additional parameters prior to adding the device.

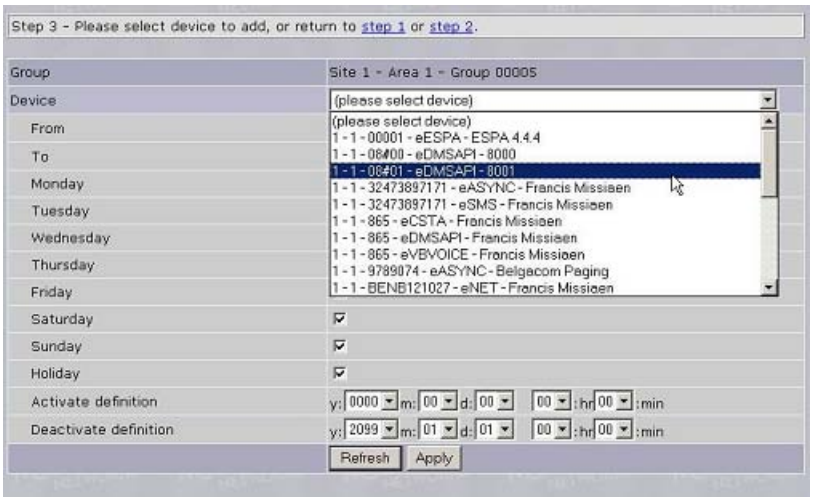


Figure 163: Select new device

Note:

You can access only the devices that belong to the same site as used by the eWEB module. Figure 163: Select new device on page 143 shows devices of site 3 because, in this example, eWEB is running in site 3 – area 1.

Change Password

Change Password allows you to enter a new password. You must enter:

- your User ID
- your old password
- your new password
- your new password (for verification)

This option eliminates the need for an eGRID-based administration of passwords of existing user profiles.

Note:

More advanced security settings or resetting passwords of users that forgot their password, still must be performed through eGRID in the eWEB_USER_AUTH table. Some additional tables (with extension _AUTH) are available for more detailed security implementation.

Info

The Info page provides web-based access to Adobe Portable Document Format (.PDF) files. The eWEB user must install on their desktop PC a suitable Adobe Acrobat Reader to open the .PDF files.

A .PDF reader is shipped on the CD-ROM, but if you can access the Internet, Avaya recommends you download the software from the Adobe web site.

There are .PDF files that handle installation issues, others that provide information on table configuration issues, and others that are more module functional.

Sign off

The sign-off link logs you out of the system. You must always sign off if you leave your browser unattended, to prevent other users from accessing eWEB functions.

Important:

Due to the users' ability to activate disaster scenarios, evacuation scenarios, and others, you must clearly inform all users of the risk they run by leaving their browser unattended. In

many situations, users who leave their browser unattended can be held personally responsible for actions that are taken with their authenticated session.

Plug-in Support

The DECT MessengereWEB module allows embedding plug-in modules that add additional functionality to the web interface. The plug-in modules can be integrated easily through the standard eWEB_TOC table. This is illustrated in Figure 164: Plug-ins added to eWEB on page 145, where additional table-of-contents entries are added for the plug-in MyPortal@Net.

	1	8	0	2909	MyPortal@Net		40
	1	8	0	2963	MyPortal@Net		0
	1	8	1	2909	MyPortal@Net	MyPortal@Net.php	40
	1	8	2	2963	MyPortal@Net	MyPortal@Net.php	40

Figure 164: Plug-ins added to eWEB

Plug-in module MyPortal@Net

An example of a plug-in module is MyPortal@Net. The interface is shown in Figure 165: MyPortal@Net plug-in on page 146. This module is not part of the base product, and is sold separately. The application provides a web interface for outbound voice-calls integrated in the eWEB module. This allows data retrieval from any data repository, including Sigma PhoneWare BTS_DIR directory. Other databases can be accessed as well through OLE/DB, ADO or sockets.

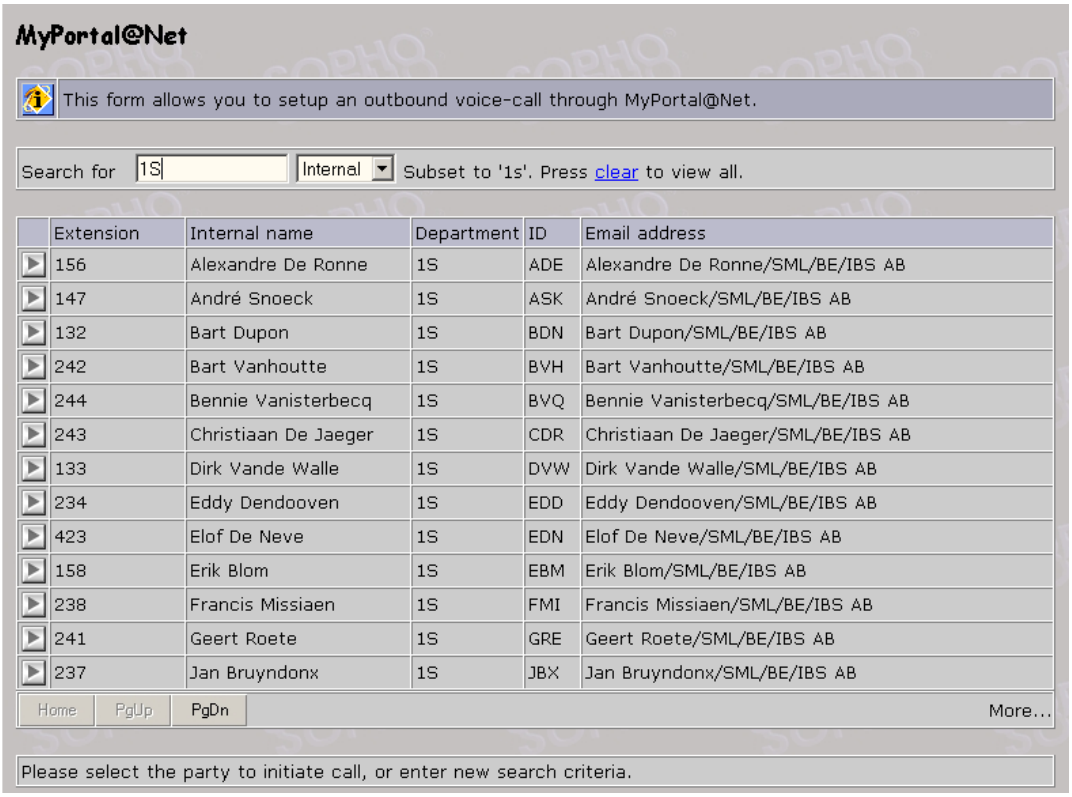


Figure 165: MyPortal@Net plug-in

The module uses native CSTA.DLL interfacing to handle voice-calls.

This option is currently not supported. Contact Avaya to determine if plug-in models are available.

Chapter 9: Module - Web Administrator

The module Web Administrator provides a web-based user interface. Web Administrator builds on the infrastructure of eWEB and depends on the same prerequisites, such as the Apache HTTP Server, PHP scriping engine, and ODBC DSN for Messenger_CFG and Messenger_DATA database.

To log on to Web Administrator, refer to Logging on to Web Administrator on page 147

Logging on to Web Administrator

1. To start the web interface, enter the URL: http://messenger/ez_index.php in Internet Explorer.
2. To log in to Web Administrator, a user name and password combination is needed. Enter the default user name admin and the default password admin.

Authentication

Authentication is based on eWEB configuration. The eCONFIG can be used to maintain users, as shown in Figure 166: eWEB user authority on page 147.

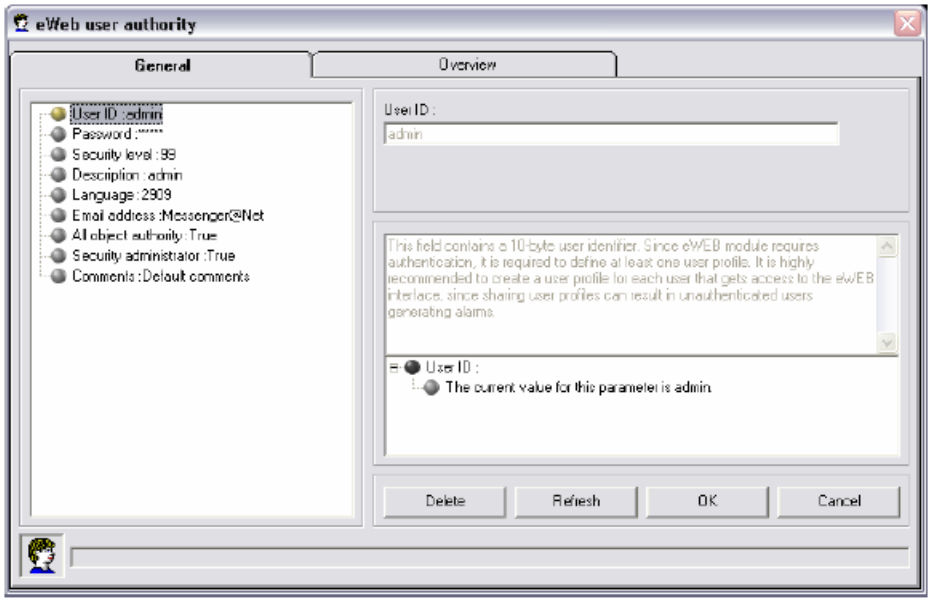


Figure 166: eWEB user authority

Work with Users

When you login with the default user name and password, you have the default administrative rights and full access to the Web Administrator. One of the features is Work with Users. Use this feature to maintain the users, as shown in Figure 167: Work with Users in Web Administrator on page 148.

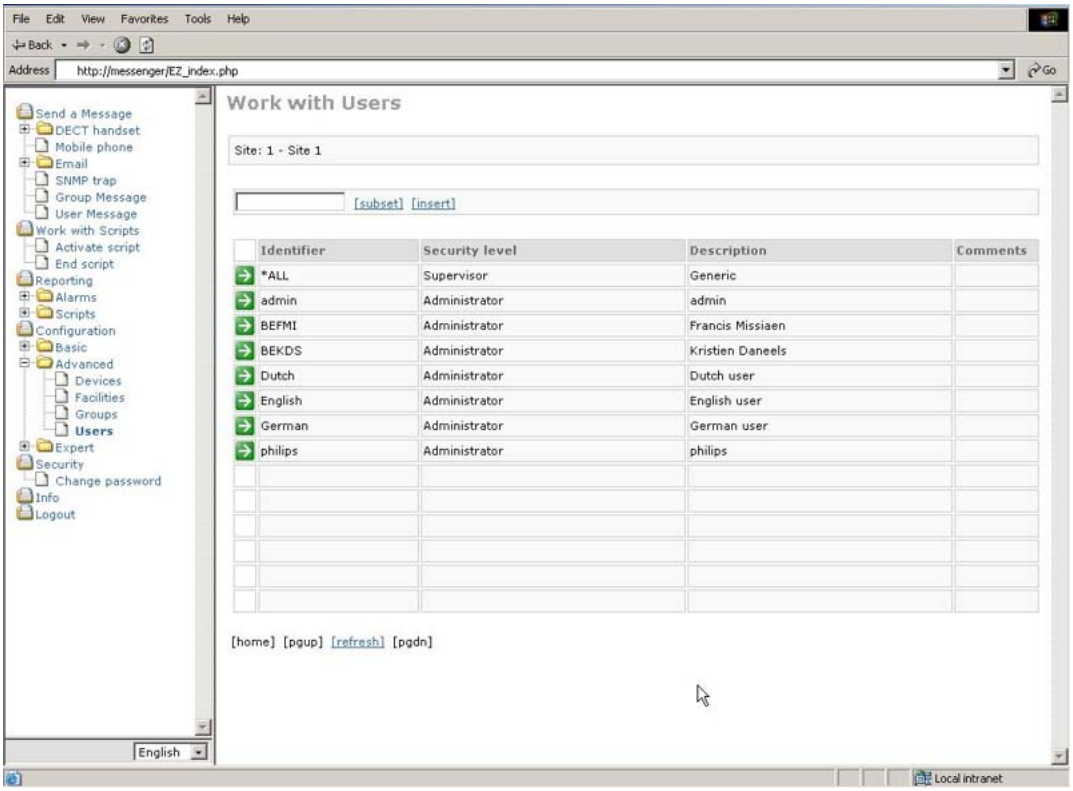


Figure 167: Work with Users in Web Administrator

Chapter 10: Module - Web Administrator User Guide

The Web Administrator provides a web-based user interface. Ensure that you have a compatible web browser, such as Internet Explorer 6.0 or Mozilla Firefox 2.0. In the web browser's Address field, enter the URL where Web Administrator is installed; for example, http://messenger/ez_index.php. When you enter this URL in your web browser, the Login page appears, as shown in Figure 168: DECT Messenger Login screen on page 149.

This sample URL applies to a Messenger@Net system that has the name messenger, and the host name messenger is a known host name in the network (such as through a DNS Server). Your administrator can optionally provide you a different URL, which can contain another host name or can consist of an IP address instead of a host name. Avaya recommends adding the URL to your favorites.

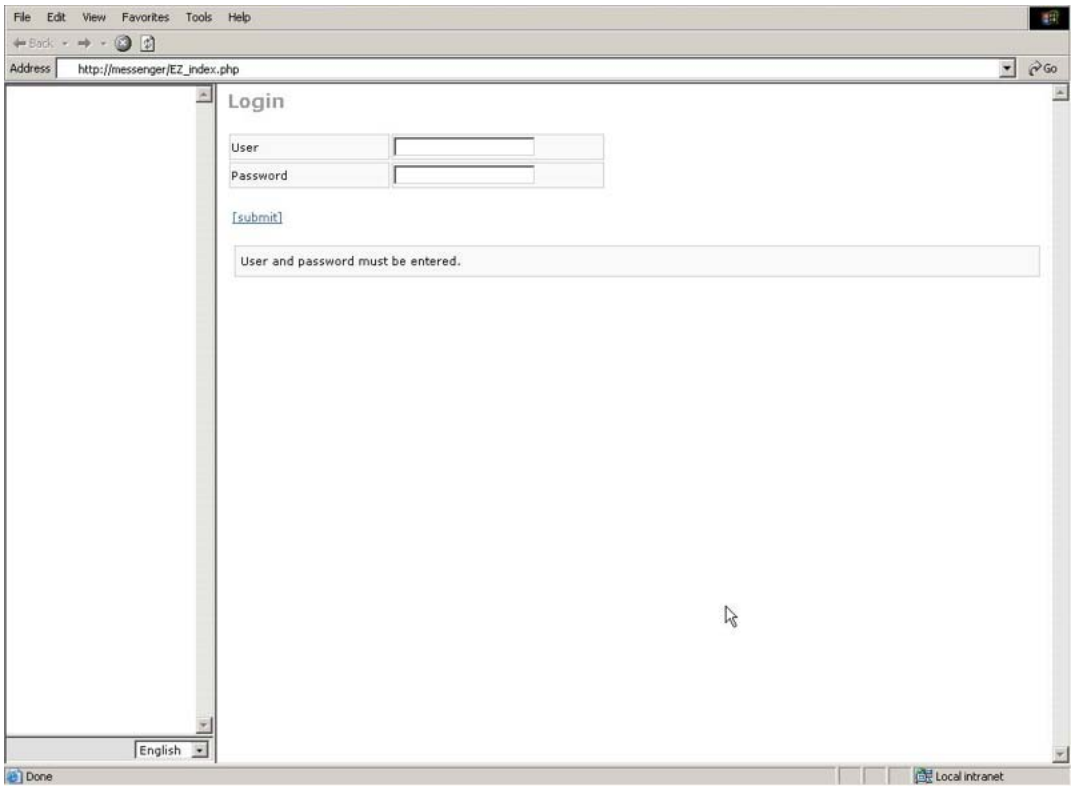


Figure 168: DECT Messenger Login screen

Authorization level

The contents of the navigation pane in the Web Administrator varies depending on the authorization granted to the user account that is used to log on. Figure 169: Navigation pane for Administrator user on page 150 shows an example of a navigation pane for a user with Administrator rights. Users with less authorization see a reduced number of links, restricting access to certain functions.

Figure 169: Navigation pane for Administrator user

DECT Messenger offers five levels of user authority, as described in <CR>.

Table 5: User authorization levels

Authorization level	Functionality
User (basic)	• Send a Message - DECT handset - Mobile phone - Email - Windows • Security • Logout

Authorization level	Functionality
User (advanced)	• Send a Message - DECT handset - Mobile phone - Email - Windows - Group Message - User Message • Security • Logout

Authorization level	Functionality
User (expert)	• Send a Message - DECT handset - Mobile phone - Email - Windows - Group Message - User Message • Reporting - Alarms • Active alarms

Authorization level	Functionality
	• Ended alarms • Report • Configuration - Basic • Group members • Alternative devices • Overview • Security - Change password • Logout

Authorization level	Functionality
Supervisor	• Send a Message - DECT handset - Mobile phone - Email - Windows - Group Message - User Message • Work with scripts - Activate script - End script • Reporting - Alarms • Active alarms • Ended alarms • Report - Scripts • Active alarms • Ended alarms • Configuration - Basic • Group members

Authorization level	Functionality
	• Alternative devices • Overview • Security - Change password • Logout

Authorization level	Functionality
Administrator	• Send a Message - DECT handset - Mobile phone - Email - Windows - Group Message - User Message • Work with scripts - Activate script - End script • Reporting - Alarms • Active alarms • Ended alarms • Report - Scripts • Active alarms • Ended alarms • Configuration - Basic • Group members • Alternative devices • Overview - Advanced • Devices

Authorization level	Functionality
	• Facilities • Groups • Users - Expert • Tasks • Configuration • PHP Info • Import • Security - Change password • Logout

The following section describes the functionality of accounts that have User authorization levels. Other accounts are described in Supervisor on page 169 and Administrator on page 174.

Log in

Your system administrator provides your user name and password. Keep your user name and password confidential; do not exchange this information with others, unless you are instructed to do so by your system administrator.

Logging in to Web Administrator

1. Open the your web browser, and in the Address field, enter the URL where Web Administrator is installed, for example: `http://messenger/ez_index.php`

   The Login page appears.
2. In the User field, enter the user name provided by your administrator.
3. In the Password field, enter the password provided by your administrator.

   The password appears as a series of asterisks (*).
4. Click **Submit**.

   If you enter an incorrect user name or password, an error message appears: Invalid user and password combination. Try again; if you still cannot log in, contact your system administrator.

The Web Administrator page

The top part of the left pane of the Web Administrator page shows graphics or images that are related to the environment or the functionality that appears on the page. When you select an option in Web Administrator the graphic changes to represent the selected function.

The middle of the left pane is a navigation menu that provides a hierarchical representation of the available functions. The available links depend on the authorization level of the user, and on what modules and infrastructure are installed or available.

The bottom part of the left pane contains a menu that allows you to change the language of the user interface. The default language is English; you can choose other languages, if they are installed on the system.

The contents of the right pane vary depending on the function you select in the navigation pane.

Log out

To log out of Web Administrator, click **Logout** in the navigation pane. Avaya recommends that you log out whenever you leave your computer unattended.

Send a message

When the Send a Message section appears in the navigation pane, you can send a message to one or more of the following, depending on what modules are available, and how the system is configured:

- DECT handset
- Mobile phone
- Email
- Windows
- Group Message
- User Message

When all DECT peripherals are considered one logical group, links are organized into a tree that is two levels deep: **Send a Message** > **DECT handset**.

In some larger environments with more than one PBX, DECT peripherals are organized depending on the PBX they are registered too. In such an environment, links are organized into a tree that is two levels deep: **Send a Message** > **DECT handset** > **Area**.

This would be the case in a environment with two areas; the DECT handsets from one campus can be logically assigned in one area, and the DECT handsets from another campus can be logically assigned in another area. The administrator defines the name for each area, such as

campus, address, building, and so on. When you send a message, you must first select the area.

Send a message to a DECT handset

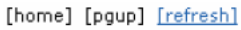
Use the following procedure to send a message to a DECT handset. You can only send messages to DECT Handsets that are configured on the system by the administrator.

Sending a message to a DECT handset

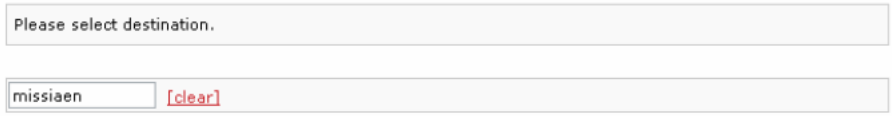
1. Log in to Web Administrator.
2. In the navigation menu, click **DECT handset**. **OR** If multiple areas are defined, click the name of the area, then click **DECT handset**.

   The Send a Message page appears, on which appears a list of all DECT handsets to which you can send a message.

   There is room to display 14 devices on the screen at any one time. If there are more than 14 handsets listed, click **[pgdn]** or **[pgup]** to navigate to the next page or previous page. To return to the first page, click **[home]**.

   In order to speed up the process of finding the destination, you can use the search field. If the search field is empty, all records are retrieved. If you enter subset criteria, only matching records are retrieved. You can click **[clear]** to clear the subset criteria.

3. Select a recipient by clicking the green arrow at the left of the entry.

   The Send a Message > Please enter message information page appears.
4. Enter the text of your message in the **Message** field.

   The length of the message you can enter is displayed in the field to the right of the Message field.
5. Use the Priority menu to assign a priority to the message; where the priority is one of Normal, Urgent, or Emergency.

6. Click **Submit**.

   Web Administrator indicates whether the message delivery succeeded or failed.

Table 6: Job aid: maximum message length

System	Maximum length of message
traditional DECT	up to 48 characters
SIP DECT	up to 160 characters

Table 7: Job aid: message priorities

Priority	Behavior
Normal	The system verifies delivery of the message without requesting confirmation from the recipient. If the message is successfully delivered, Web Administrator displays the result within a few seconds. If the message cannot be delivered, a longer delay occurs before Web Administrator displays the result.
Urgent	The system waits until the message is delivered, and confirmed by the recipient before Web Administrator displays the result. The recipient has 30 seconds in which to confirm receipt. During this time, the Web Administrator page is nonresponsive.
Emergency	The Emergency option is not available in some system configurations, and in some configurations where it is available, Emergency messages are handled in the same way as Urgent messages. Consult you system administrator for more information about the priorities that are available for your use.
Web Administrator reports whether the message was delivered successfully. If a Normal message cannot be delivered, or if an Urgent or Emergency message either cannot be delivered, or is not confirmed by the recipient, the following message appears: `Error. Send message ended abnormally .`	

Send a message to a mobile phone

You can send messages to a mobile phone only if the eSMS module is installed and licensed, and configured on the system. Use the following procedure to send an SMS message to a mobile GSM phone. You can only send messages to mobile phones that are configured on the system by the administrator.

Sending a message to a mobile phone

1. Log in to Web Administrator.
2. In the navigation menu, click **Mobile phone**.**OR** If multiple areas are defined, click the name of the area, then click **Mobile phone**.

   The Send a Message page appears, on which appears a list of mobile phones to which you can send a message.

   There is room to display 14 devices on the screen at any one time. If there are more than 14 phones listed, click **[pgdn]** or **[pgup]** to navigate to the next page or previous page. To return to the first page, click **[home]**.

   [home] [pgup] [refresh] [pgdn]

3. Select a recipient by clicking the green arrow at the left of the entry.

   The Send a Message > Please enter message information page appears.
4. Enter the text of your message in the **Message** field.

   The length of the message you can enter is displayed in the field to the right of the Message field.
5. Click **Submit**.

   Web Administrator indicates whether the message delivery succeeded or failed.

Web Administrator reports that the message was successfully delivered when the mobile provider accepted the message. If the mobile phone is powered off, the mobile phone user can receive the message only when the when the mobile phone is later switched on.

Send a message to an e-mail address

You can send messages to e-mail addresses only if the eSMTP module is installed, licensed, and configured on the system, and can send e-mail messages only to e-mail addresses that are configured in the system by the administrator.

Sending a message to an e-mail address

1. Log in to Web Administrator.
2. In the navigation menu, click **Email** .**OR** If multiple areas are defined, click the name of the area, then click **Email**.

   The Send a Message page appears.

There is room to display 14 devices on the screen at any one time. If there are more than 14 address listed, click **[pgdn]** or **[pgup]** to navigate to the next page or previous page. To return to the first page, click **[home]**.

3. Select a recipient by clicking the green arrow at the left of the entry.

The Send a Message > Please enter message information page appears.

4. Enter the text of your message in the Message field.

The length of the message you can enter is displayed in the field to the right of the Message field.

5. Click **Submit**.

Web Administrator indicates whether the message delivery succeeded or failed.

Important:

The e-mail is sent on behalf of the e-mail account that was configured by the administrator. Therefore, you have no record of this message, and any responses are not delivered to you.

Send a message using Group message

Group messages are messages you can send to previously configured groups of recipients. Group messaging is only available if the administrator has configured it, and has populated the eWEB_SNDGRPMSG table.

Sending a message using Group message

1. Log in to Web Administrator.
2. In the navigation menu, click **Group message**. **OR** If multiple areas are defined, click the name of the area, then click **Group message**.

   The Send Group Message > Please select group page appears.
3. Select the group to which to send a message.

   The Send Group Message > Please confirm group page appears. This page lists the group members.

   If you select the wrong group, click **back** to select a different group.
4. Click **continue** to confirm the group selection.

   The Send Group Message > Please select message page appears.

Up to three types of messages are supported, depending on your system configuration:

- Private messages
- Shared messages
- Free messages

5. Select the message to send by clicking the green arrow next to it. If you select **Free message**, type the text of your message.
6. Click **Submit**.

   Web Administrator indicates whether the message delivery succeeded or failed.

Table 8: Job aid: Group message types

Type	Description
Private messages	Are messages, defined by your system administrator, that you can send only to the selected group.
Shared messages	Are messages, defined by your system administrator, that you can send to any group.
Free messages	Are custom messages you type yourself.

Send a message using User message

User messages are messages you can send to previously configured groups of recipients. User messaging is only available if the administrator has configured it, and has populated the eWEB_SNDGRPMSG table.

Sending a message using User message

1. Log in to Web Administrator.
2. In the navigation menu, click **User message**. OR If multiple areas are defined, click the name of the area, then click **User message**.

   The Send User Message > Please select group page appears.
3. Select the group to which to send a message.

   The Send User Message > Please confirm group page appears. This page lists the group members.

   If you select the wrong group, **back** to select a different group.
4. Click **continue** to confirm the group selection.

   The Send User Message > Please select message page appears.

Up to three types of messages are supported, depending on your system configuration:

- Private messages
- Shared messages
- Free messages

5. Select the message to send by clicking the green arrow next to it. If you select **Free message**, type the text of your message.
6. Click **Submit**.

   Web Administrator indicates whether the message delivery succeeded or failed.

Table 9: Job aid: User message types

Type	Description
Private messages	Are messages, defined by your system administrator, that you can send only to the selected group.
Shared messages	Are messages, defined by your system administrator, that you can send to any group.
Free messages	Are custom messages you type yourself.

Change password

Use the information in this section to change your password. Avaya recommends that you change your password whenever you suspect someone else may have access to it.

Changing your password

1. Log in to Web Administrator.
2. In the navigation menu, click **Change password**.

   The Change password page appears.
3. In the **Old password** field, enter your existing password.

   If you do not know your existing password, contact your system administrator.
4. In the **New password** field, enter your new password.
5. In the **New password (confirm)** field, reenter your new password.

Reports of active alarms

The system generates reports of active alarms that are processed through eKERNEL, and makes them available to users with User (expert), Supervisor, and Administrator privilege.

Reports are generated from the information that is stored in the Messenger_DATA database, which is an internal repository that temporarily stores active alarms.

Alarms are organized according to output program, and the number of active alarms appears next to each output program.

The report provides a snapshot of an instant in time, and is not updated in real-time. Click **refresh** to update the information in the report.

Click on a Module to see the details of active alarms for that program.

When you select a module in the Work with Active Alarms page, the active alarms associated with the selected output module appear, and for each alarm the system displays the destination device Identifier, Output program, Message, and Next call timestamp. The search field can be used to subset the view to selected subset criteria. The navigation keys [home] and [pgup] and [pgdn] allow navigating through the list. The [refresh] allows you to take a new snapshot.

Click the green arrow to get additional details on the selected active alarm.

Note:

The [reset] link should only be used by Expert users to provide them with the ability to selectively set an alarm. Alarms are normally set automatically following normal call flow. Prior to release 4.0, removing an alarm can be done only through general Reset all alarms in eKERNEL or through low level database maintenance tools, such as SQL Server 2000 Enterprise Manager.

Reports of ended alarms

The system generates reports of ended alarms that are processed through eLOG, and makes them available to users with User (expert), Supervisor, and Administrator privilege.

You are provided access to the internal repository that is maintained through the eLOG module through Work with Ended Alarms. Work with Ended alarms writes information into a comma separated file for every input request, output request, and output response.

You can use the optional eLOG module to access the information provided by Work with Ended alarms.

Note:

eLOG is a an add-on module, which you can purchase separately as part of a Premium Package.

The eLOG repository is stored for a configurable number of days.

You choose a date in the first selection screen. The default is the current day.

You choose between Input request and Output request in the second selection screen.

The example in the following figure shows a number of input requests. Click the green arrow to view additional details.

You can also perform optional filtering, for example upon Message.

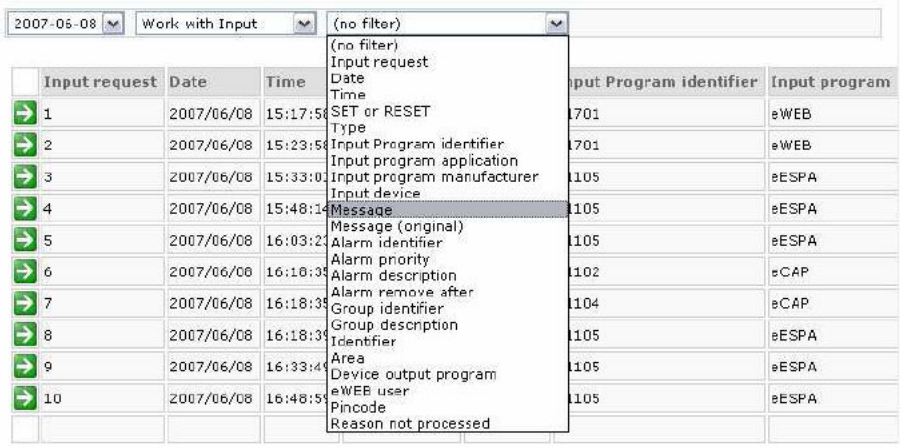


Figure 170: Work with Ended alarms - Input requests

Reports on alarms

Reports on alarms is available to User (expert) and Supervisor and Administrator.

Work with Report provides an easy web-based reporting function to combine the available information gathered by eLOG module. Work with Report also exposes the data available in comma separated files.

Unlike the functions discussed earlier in this chapter, the Input request, Output request, and Output response are automatically consolidated.

To access this information, you need the eLOG.

The first page of Work with Report shows a selection box with available dates. The current day is the default.

The following figure shows the second selection box. This selection box shows the available messages for that day, sorted alphabetically. Click a message twice to obtain the requested data.

The message shown in the report is the consolidated result of *INPUT, *OUTPUT, and *RESPONSE in order of date and time. Expert Users that need to answer inquiries on message arrival and notification can use the report to provide the answers.

Configuration of basic group members

Users (expert), Supervisors, and Administrators can use the configuration function Work with Group Members for web-based maintenance of group members. The function also internally maintains the content of the eKERNEL_GROUP_MEMBER table.

Follow the steps in the next procedure to configure Work with Group Members.

Configuring Work with Group Members

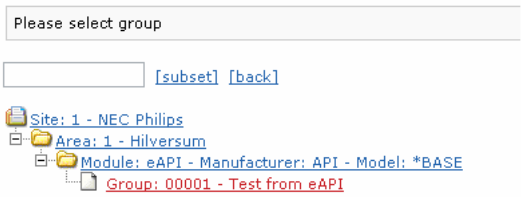
1. Select the input module you want to maintain on the first page.

   The first page provides a hierarchical overview of the available input modules.

   The following figure shows an example of a selected module. In the example, the configuration contains two areas. The eAPI input module on area 1 is selected.

2. Select a group for the chosen input module.

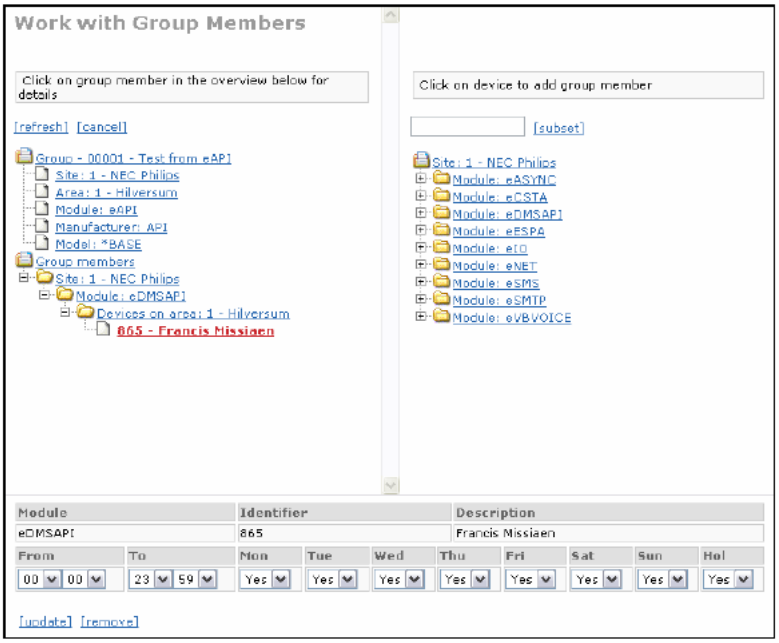
   In the example shown in the following figure, group 00001 is chosen.



   The left-hand section provides the current group members The right-hand section provides a list of all available peripherals. The bottom area provides space to show details.

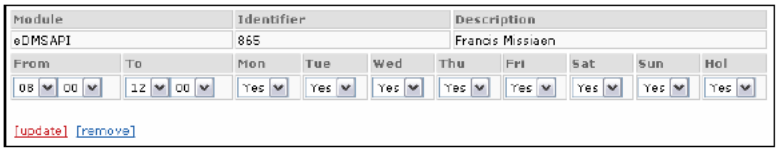
3. Click on an existing group member to see details.

   The following figure shows the details after clicking on 865 – Francis Missiaen. The bottom area shows details such as the start hour, end hour, and weekly presence.



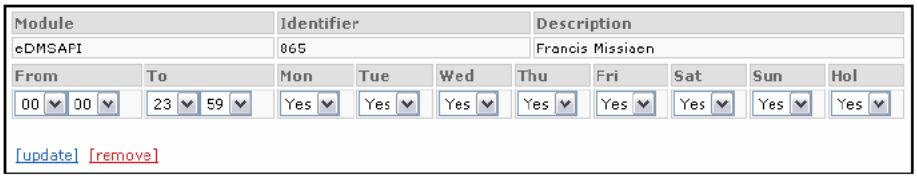
4. To update the details, change the start hour and end hour and click the [update] link.

   In the example shown in the next figure, an update is done by changing From into 08:00 and To into 12:00 and clicking the [update] link.



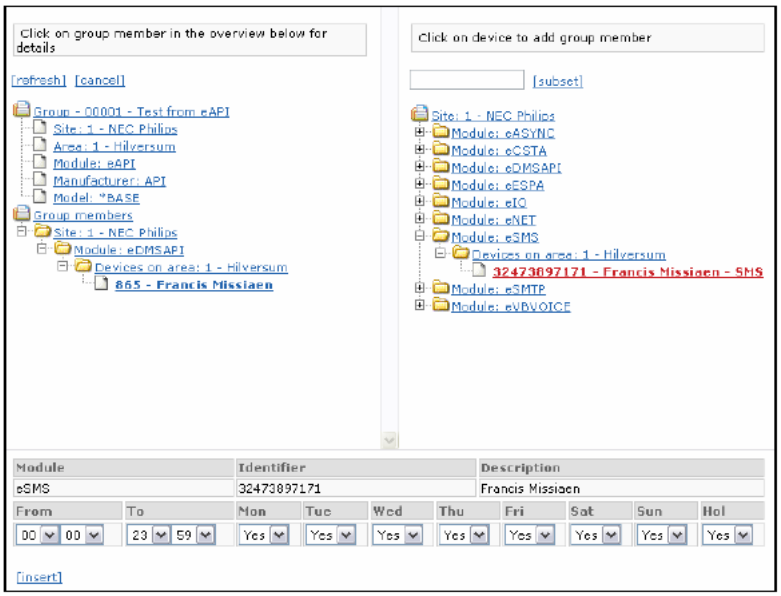
5. Click the [remove] link to remove a group member.

   See the example in the following figure.



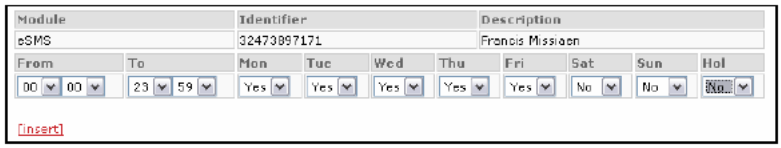
6. Select a destination device in the right-hand area to insert a new group member.

   In the example shown in the next figure, the eSMS peripheral 32473897171 belonging to Francis Missiaen is selected. The default details showing membership from 00:00 to 23:59 on all days is presented.



7. Optionally, adjust the default settings and click [insert].

   In the example shown in the following figure, the default settings Sat, Sun, and Hol are changed to No.



Configuration of basic alternative devices

User (expert), Supervisor and Administrator can use the function Work with Alternative Devices.

Work with Alternative Devices provide a web-based user interface to maintain the internal configuration table eKERNEL_DEVICE_ALT.

Follow the steps in the next procedure to configure alternative devices.

Configuring basic alternative devices

1. Log in to Web Administrator.
2. In the navigation menu, click **Work with Alternative Devices**.

   The Work with alternative devices page appears.
3. Under Configuration > Basic, click Alternative devices .

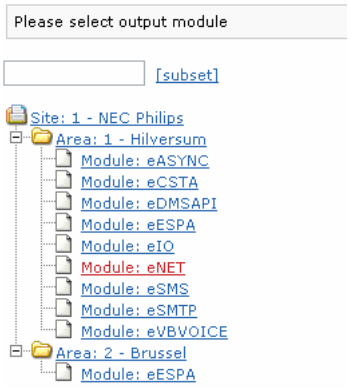
   The page Please select output moduleappears.

4. Select an output program.

   Peripherals are associated to output programs and therefore, Mobile phones resides under an instance of the eSMS module, and DECT handsets resides under an instance of eDMSAPI.

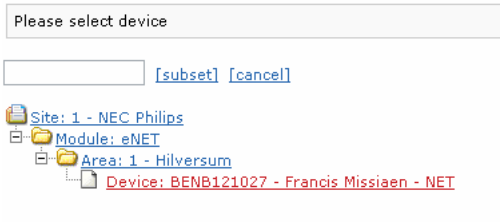
   In the example shown in the following figure, an alternative device for a Windows PC is required. The output module eNET is selected.

   The objective in the example is to define an alternative device so that if a message cannot be delivered using eNET; if for example portable PC is not online, the message is rerouted to a mobile GSM phone.



   After you select an output module, you are presented with an overview of the available devices for that output module.

   In the same example presented previously, the only available device listed for the output module eNET is a single PC named BENB121027 that belongs to user Francis Missiaen. See the following figure.



5. Select a device.

   A page appears that presents you with a number of options.

   The top-left-hand area of the page gives details on the selected device are shown as well as the available alternative devices. When no alternative devices are present, (none) is shown.

   The top-right-hand area shows all available devices. You can use the subset criteria to limit the list according to selection criteria.

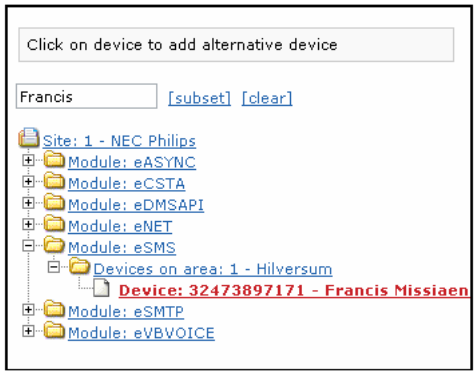
   The bottom of the page shows details that vary according to the options you choose in the other areas of the page.

6. Click on the device listed on the right of the page to add an alternative device.

To add the mobile phone of Francis Missiaen as an alternative device, Francis is typed into the subset field under Click on devices to add alternative device, and [subset] is clicked.

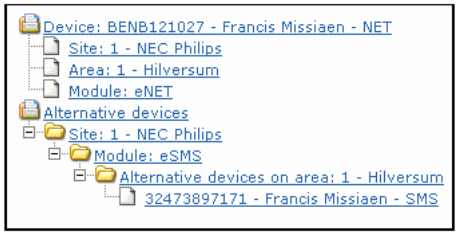
7. If you want to add the name of an alternative device, type the name you want to add in the [subset] field under the section on the right called **Click on device to add alternative device**.
8. Click on **[subset]**.

   In the example shown in the following figure, another option is illustrated; restricting the available devices. In the right hand section of the page, under [subset], Francis is typed and then [subset] is clicked. The mobile phone of Francis Missiaen appears after clicking on eSMS.



Then Device: 32473897171 – Francis Missiaen is clicked and the details shown in the next figure appear. When [insert] is clicked, the mobile phone is added as an alternative device.

As a result, the mobile phone is defined as alternative device for the device BENB121027, as shown in the following figure.



Configuration basic overview

The Configuration basic overview function is available to User (expert), Supervisor, and Administrator.

You can select a group, check the group members, and check the alternative devices with Overview. Overview basically responds to the question: "What happens to my alarms?"

Follow the steps in the next procedure to configure basic overview.

Configuring basic overview

Navigate to Site 1 – NEC Philips > Area 1 – Hilversum > Module eWeb – Manufacturer eWEB – Model *BASE > Group: ENET – Test from eWEB to eNET.

The contents of the selected group appear as a result of alarms that originate in the eWEB input module and addressed to group ENET.

In the example below, there appears to be one group member, the Device 1.1.eNET.BENB121027. One attempt (1x) is made to notify the device.

If that attempt is unsuccessful, the alternative device 1.1.eSMS.32472897171 is used. One attempt (1x) is made to notify the alternative device.

Figure 171: Contents of the selected group

Click on the group member to see the details on From, To, and Daily presence.

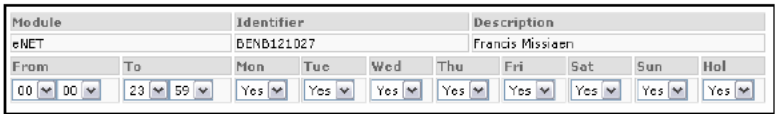


Figure 172: Group member details

Supervisor

A user with Supervisor authority is granted addition access to **Work with Scripts**. This function is accessed through the links **Activate script**and **End scrip**t.

Reporting is extended with **Scripts**, with access to link **Active scripts** and **Ended scripts**Ended scripts.

Figure 173: Supervisor access to Work with Scripts

Work with scripts - activate script

The Activate Script function in Work with Scripts is available to Supervisor and Administrator.

Follow the steps in the next procedure to use activate script.

Using Activate Script

1. Open **Work with Scripts** and click **Active script**.

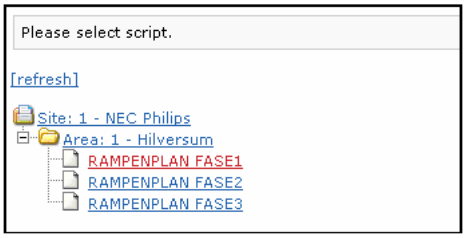
   You can only access Activate Script in this manner and it is only available if the administrator defined scripts in the internal table eWEB_SCRIPT.

   Note:

   The Web Administrator no longer implements the tables eWEB_SCRIPT_SET, eWEB_SCRIPT_TRACE, and eWEB_SCRIPT_END. If you want to implement details script authority, provide the original eWEB interface to your users through http://messenger/index.php.

   The previous figure shows the configured scripts.

2. Click on one of the configured scripts to select that script.



The next window shows a preview of the script.

In the top section, details are visible on the script, such as message and group and current user.

The bottom section shows the group members belonging to the selected group.

Your ability to check or un-check the check boxes in front of groups members depends on the configuration of the script. In some configurations, you cannot un-check group members as the check boxes are disabled.

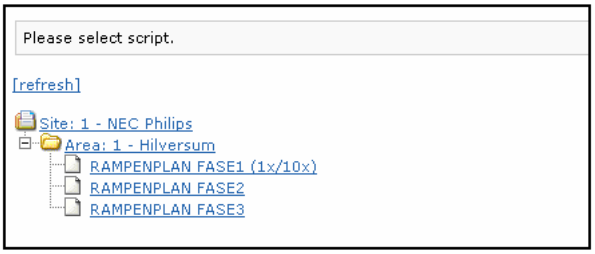
If you deselect too many members, the error below appears. A script defines the minimum amount of group members that are part of the script.

3. Click **[submit]** to activate the script.

   You then return to the main window.

4. Click **[refresh]** to see an update of the available scripts.

   In the example in the following figure, the script is activated once. In the example configuration, the script can be activated ten time, as shown in the 1x/10x indicator.



Work with Scripts - End Script

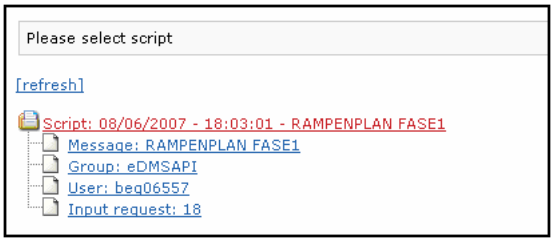
The function Work with End Script is available to Supervisor and Administrator.

Follow the steps in the next procedure to use Work with End Script.

Ending Scripts

1. Open the Work with Scripts window and click **End scrip**t.

   An empty screen indicates that no active scripts are present and so no active scripts are available to end.
2. Ensure that the script you end is the correct active script. Verify the date and time and description.
3. Click on the **Script** to end the script.



When you end the script, a window appears showing an overview of the scripts that are ended. The repository of scripts is automatically cleaned up according to the configuration settings.

Reporting active scripts

The Reporting active scripts function is available to Supervisor and Administrator.

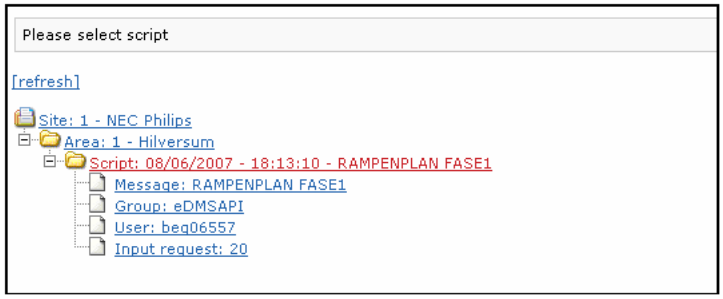
Follow the steps in the next procedure to report active scripts.

Reporting active scripts

1. From the Reporting window, click**Scripts**.
2. Click **Active scripts**.

When no active scripts are available, an empty screen is shown.

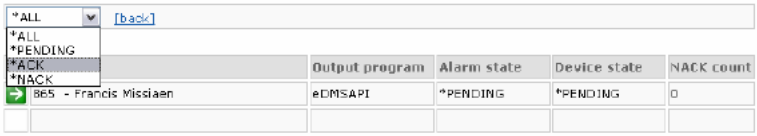
3. When one or more active scripts are available, highlight the **script identifier** to see details.



The window shows the progress of an active script.

There is a subset box near the top of the window. The subset box has the options *ALL, *PENDING, *ACK, and *NACK.

When you select *ALL, you get a list of all active scripts. When you want to narrow the results of your search, choose *PENDING, *ACK, or *NACK. *PENDING indicates where notification is still in progress, *ACK indicates those who have responded, and *NACK indicates those who failed to respond.



Reporting ended scripts

The function Reporting ended scripts is available to Supervisor and Administrator.

Follow the steps in the next procedure to report ended scripts.

Reporting ended scripts

1. Open the Reporting window and click **Scripts**.
2. Click **Ended scripts**.

   When no ended scripts are available, an empty screen is shown.
3. Use the date and time criteria to identify the script.
4. Click on the Script of your choice from the list presented to narrow down the details of the ended script.

Administrator

A user with Administrator rights has full access to all capabilities of the Web Administrator.

Send an SNMP trap

The function Send an SNMP trap is available to Administrator.

SNMP trap is shown in the Send a Message window when the system has the eSNMP module installed, licensed and configured. SNMP trap provides a web-based basic implementation of an SNMPv1 trap sender.

For details on the capabilities of SNMP trap, refer to Module - eSNMP on page 43.

To send an SNMP trap, enter the fields as shown in the following paragraph and the example.

In the following example, an SNMPv1 trap is sent to 127.0.0.1 with community public and enterprise OID 1.3.6.1.4.1.28088.32.1. The enterprise OID 1.3.6.1.4.1.28088.32.1 is registered by NEC Philips (HQ) by Francis Missiaen. The OID range starting with 1.3.6.1.4.1.28088.32.1 is reserved by UCPS division.

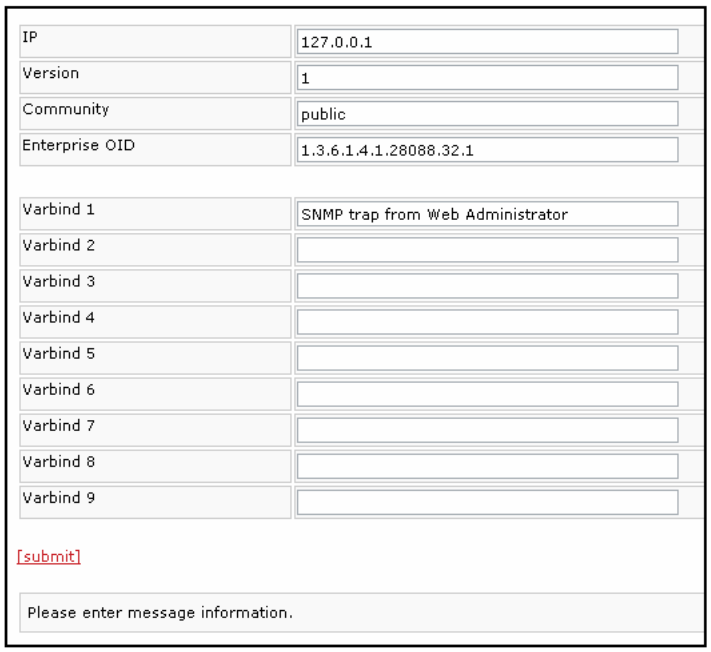


Figure 174: SNMPv1 trap example

In the example one varbind parameters is given. A resulting action depends on the Messenger configuration settings. For example, in Figure 175: eSNMP module receives SNMP trap on page 175 the eSNMP module receives the SNMP trap in Figure 174: SNMPv1 trap example on page 175.

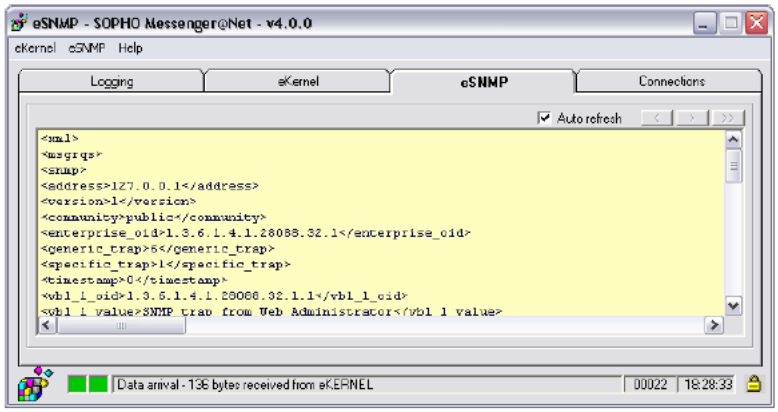


Figure 175: eSNMP module receives SNMP trap

In this example, the result is that a Windows popup message is sent through eNET.

Note:

The resulting popup message contains the varbind parameter from the data entered through Web Administrator.

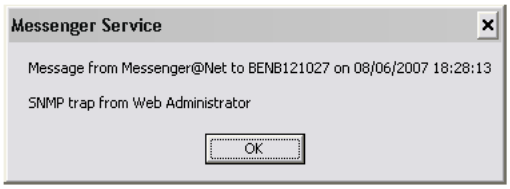


Figure 176: Popup message sent through eNET

Advanced configuration

The advanced configuration is reserved for users with Administrator rights, and provides maintenance of Devices, Facilities, Groups, and Users.

Configuration of advanced facilities

The configuration of advance facilities function is available to Administrator.

Follow the steps in the next procedure to configure advanced facilities.

Configuring advanced facilities

1. Open the Configuration window and click on **Advanced**.
2. Click on **Facilities**.

Note:

The Work with facilities function maintains the table eKERNEL_DEVICE_FORMAT in Messenger_CFG database. The following figure shows an overview of sample definitions, as seen in eGRID interface.

FMT_OUTPGM_Appl_str	FMT_OUTPGM_Facility_st	FMT_Bytes_line1_n	FMT_Bytes_line2_n	FMT_Bytes_line3_n	FMT_Page_ind_n	FMT_Page_mate_ind_n	FMT_Concatination_b
eASYNC	KPN	120	0	0	3	2	0
eASYNC	PAGING	160	0	0	3	2	0
eASYNC	PROXIMUS	120	0	0	3	0	0
eCSTA	CS11	10	0	0	0	0	0
eCSTA	CS22	10	0	0	0	0	0
eCSTA	CS11	16	16	0	0	0	-1
eCSTA	CS22	16	16	0	5	2	-1
eCSTA	CS33	16	16	0	5	2	-1
eCSTA	CS44	16	16	0	5	2	-1
eCSTA	D300	14	0	0	5	0	0
eCSTA	D340	30	0	0	0	0	0
eCSTA	D500	12	12	0	5	2	0
eCSTA	I600	16	16	0	5	2	-1
eCSTA	P375D	19	0	0	0	0	0
eCSTA	ZENIA	13	0	0	0	0	0
eDMSAPI	CS22	16	16	0	5	2	-1
eDMSAPI	CS33	16	16	0	5	0	0
eDMSAPI	CS44	16	16	0	5	2	0
eDMSAPI	I600	16	16	0	5	2	-1
eESPA	ESPA	128	0	0	0	0	-1
eIO	DO	1024	0	0	0	0	0
eNET	NET	128	0	0	0	0	0
eOAI	OAI	512	0	0	0	0	0
eOAP	OAP	512	0	0	0	0	0
eSMS	SMS	160	0	0	0	0	0
eSMTP	SMTP	160	0	0	0	0	0
eVMVOICE	VMVOICE	1024	0	0	0	0	0

3. Select an output module in the first window that opens in Work with facilities.

This action must take place before you configure Facilities and before you create Devices in the system.

When no definitions are available, an empty screen appears.

4. Use the **[home]**, **[pgup]**, and**[pgdn]** links to navigate through the list of definitions.
5. Click the **[insert]** link to add a new definition.
6. Click the green arrow to change an existing definition.

   Note:

   A Facility is considered an unique key. If, for example, you have defined C944 once, you cannot add a second C944 definition.

   The next figure shows the details of the selected definition C944.
7. To return from the detailed screen, click the**[cancel]** link.

   When you click the **[delete]** link, clicked, the facility is removed.

   Important:

   In the current release of Web Administrator, no validation is to see if the definition is in use. This means you must verify if a facility is used before you delete the facility. For example, if devices exist with facility C944, you must not delete the facility. Future releases may implement a disabled [delete] link when a definition is in use.
8. Click **[edit]** to alter some (non-key) fields. .

   Note:

   The name of the definition cannot be changed. You need to**[insert]** a new definition if you want another Facility name.
9. Click **[apply]** to adjust the values.
10. Click **[cancel]** if you want your changes to be ignored.

Configuration of advanced devices

The Configuration of advanced devices function is available to Administrator.

Follow the steps in the next procedure to configure advanced devices.

Configuring advanced devices

1. Open the Configuration window and click **Advanced**.
2. Click **Devices**.
3. Define at least one Facility before you create a device.

In order to create Devices for an output module, you need definitions of Facilities. For example, if you want to add an eDMSAPI device, at least one Facility needs to be defined before creating a Device.

Note:

This function maintains the internal configuration table eKERNEL_DEVICE in the Messenger_CFG database.

4. In the first window, select the Output module.

   An overview of the existing Devices is shown. When no definitions are available, an empty screen is shown.

5. Use the **[home]**,**[pgup]**, and**[pgdn]** links to navigate through the list of definitions.
6. To add a new definition, click the **[insert]** link.
7. To change an existing definition, click the green arrow.

   Note that a Device is considered a unique key, so for example, if you define 865 once, you cannot add a second 865 definition.

   To add a new device, click on **[insert]**.

8. In the next window, verify the input capable fields and add missing information.

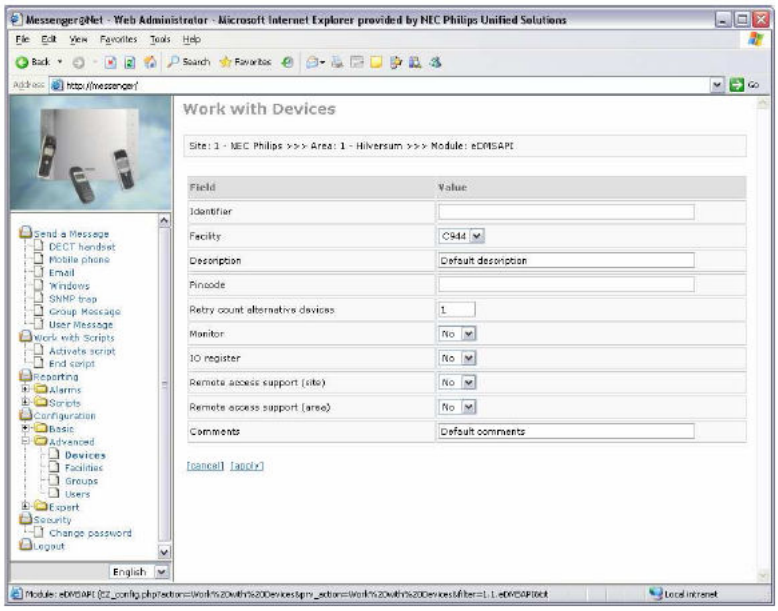
   When adding devices, you need to select an existing Facility.

   Note that in cases that no facilities are as yet created yet, you must define them first, and then return to the Work with Devices entry.

   When adding a DECT handset, the field Monitor refers to the ability to trace the voice-call divert. This typically requires an additional eCSTA module and sufficient channel licenses. Leave this field set to **No** unless you receive instructions from the administrator.

   Set the IO register field to **No** unless instructed otherwise by the administrator. A value of **Yes** is used when the configuration implements inbound LRMS messaging. A value of Yes is used, for example, in combination with the eLOCATION module to generate location alarms. This also requires additional licenses.

   The Remote access support site and Remote access support (area) are available for backwards compatibility with eWEB module, but are not implemented in Web Administrator. Avaya recommends that you leave the default setting of No.



9. Type **[apply]** to insert the definition.
10. To return from the detailed screen, click **[cancel]**.

    Note:

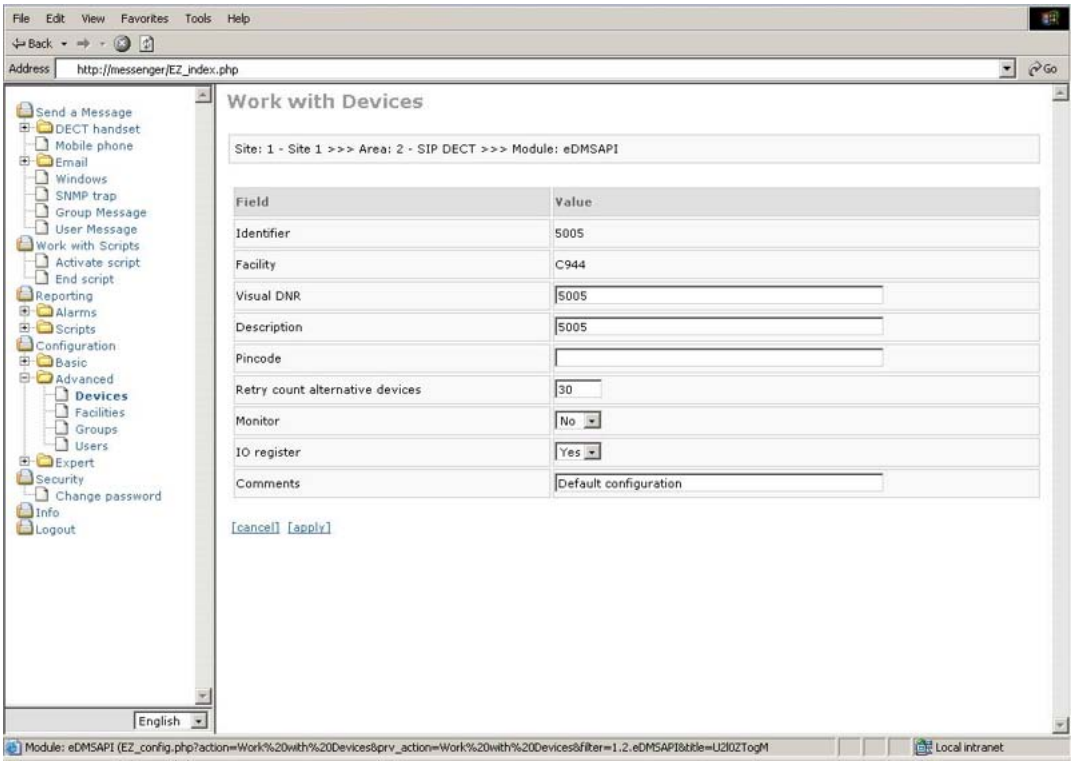
    Avaya highly recommended that you add a meaningful Description, as this is represented to the Web Administrator during further maintenance. Avaya also strongly recommends that you specify first name and last name, or any other unique reference, to define the owner of the peripheral.

11. To maintain an existing device, click the green arrow in front of an existing definition.

	Identifier	Facility	Description	Retry count alternative devices
→	865	C944	Francis Missiaen	1

12. Click **[edit]** to maintain the record.
13. Click**[delete]** to remove a record.

In the example in the following figure, **[edit]** is clicked to maintain details of selected device 865. Note that some fields cannot be altered. If you want to alter some fields, you must recreate the device.



When the**[delete]** link is clicked, the device is removed.

Note:

The current release of Web Administrator allows deleting a device without checking if it is in use by other configuration tables. Many configuration tables can be involved, for example, eKERNEL_GROUP_MEMBER, and eKERNEL_DEVICE_ALT, but also the tables related to inbound eCSTA, eDMSAPI and eLOCATION definition. Carefully verify if a device is used prior to deleting it. In future release additional in use checks are to be implemented to prevent deleting devices that are defined on another level.

When you click**[edit]** , some (non-key) fields can be altered.

Note:

The name of the definition cannot be changed. You need to [insert] a new definition if you want another Device name.

In the example in the following figure, **[edit]** is clicked for definition 865. You can adjust the values and click **[apply]**. If you click **[cancel]** any changed are ignored.

Configuration of advanced groups

The Configuration of advanced groups function is available to Administrator.

Follow the steps in the next procedure to configure advanced groups.

Configuring advanced groups

1. Open the Configuration window and click **Advanced**.
2. To access Work with Groups, click **Groups**.

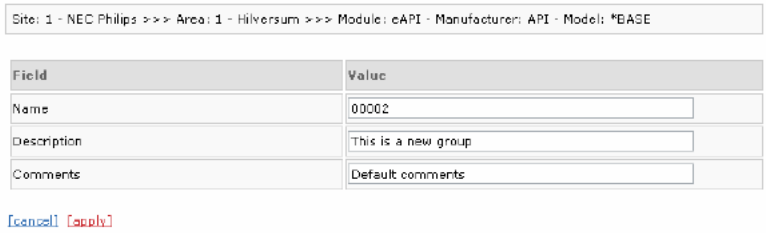
   Groups are associated to input capable modules. Therefore, on the first screen a selection is requested on the Input module.

   Next an overview of existing groups for the chosen input module is shown.
3. Use **[home]**, **[pgup]**, and **[pgdn]** to navigate through the list of definitions.
4. To add a new definition, click **[insert]**.
5. Click the green arrow to change an existing definition.

   Note:

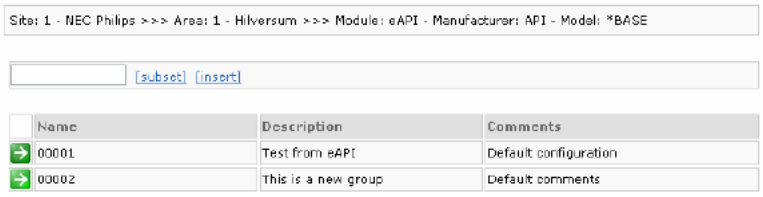
   A Group is considered a unique key, so for example if you have defined 00001 once, you cannot add a second 00001 definition.

   The next figure has the details of group 00002.

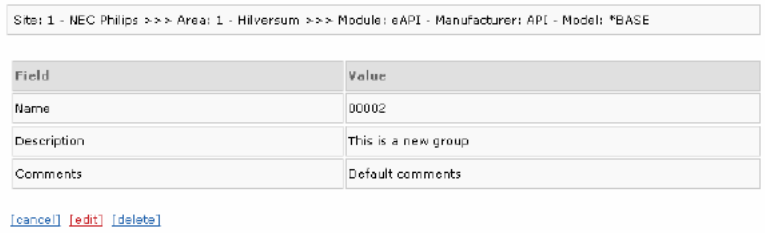


   Avaya highly recommends that you add a meaningful Description, as this is represented to the Web Administrator during further maintenance.
6. Click the green arrow in front of a definition to maintain an existing entry.

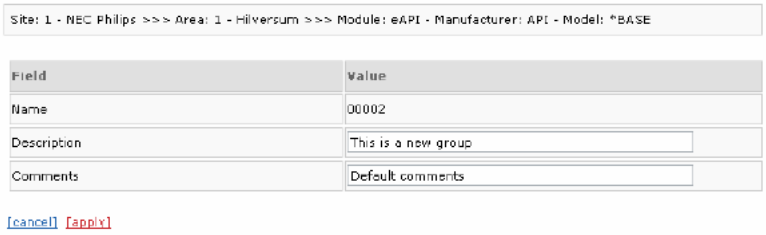
   For example in the next figure, the group 00002 is maintained.



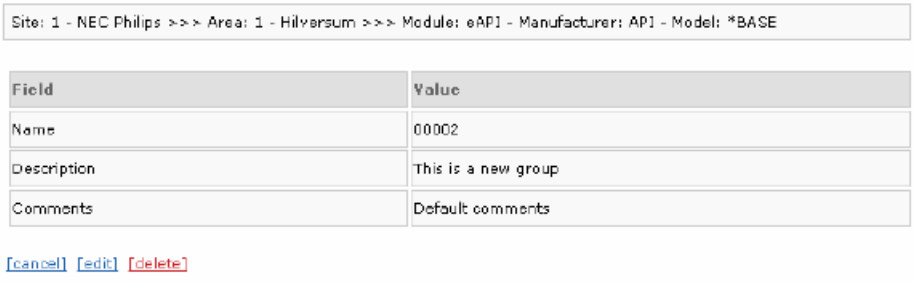
   The detailed screen shows the selected definition.



7. Click **[edit]** to alter a number of input capable fields.
8. Enter changes, then click **[apply]**.



9. To delete an existing definition, select the group and click [delete].



Note:

In current release of Web Administrator, no check is performed to verify if the group is in use.

- There can be group members but they are not removed automatically when a group is deleted. You should verify if group members exist and remove them prior to deleting the group.
- The group can be defined elsewhere in the business logic of the Messenger configuration, for example, associated with definitions such as tables related to eIO (eIO_DI, eIO_DO, eIO_AI, and so on), related to eLOCATION, inbound eCSTA, eDMSAPI, and so on.

  The verification of usage of a group is the responsibility of the administrator.

Configuration of advanced users

The Configuration of advanced users function is available to Administrator.

Follow the steps in the next procedure to configure advanced users.

Configuring advanced users

1. Open the Configuration menu and click **Advanced**.
2. Click **Users**.

The Work with Users page opens.

Note:

The Work with Users function maintains the eWEB_USER_AUTH configuration table of the Messenger_CFG database. The eWEB_USER_AUTH configuration table defines access to eWEB and Web Administrator modules. Do not delete the *ALL or admin definitions. When you delete these definitions, you are no longer able to authenticate for future maintenance through eWEB or Web Administrator.

On the Work with Users main page, an overview of existing users is shown.

3. Click the **[home]**,**[pgup]**, and **[pgdn]** links to navigate through the list of definitions.
4. Click the**[insert]** link to add a new definition.
5. Click the green arrow to change an existing definition.

Note:

A User is considered a unique key, so for example if you define admin once, you cannot add a second admin definition.

In the entry screen, the Administrator enters the input capable fields.

- Identifier refers to the user field that is assigned to the user
- Password refers to the password field that is assigned to the user

The Administrator needs to provide both the Identifier and Password to the end-user, as these fields are needed to authenticate on Web Administrator.

You should consider sharing this chapter **Module_Web_Administrator – User Guide** with the end-user, as well as providing the end-user with further information.

Note:

Avaya highly recommends that you enter the first name and last name in the field description.

The security level is in Web Administrator catalogued into 5 different levels, User (basic), User (advanced), User (expert), Supervisor, and Administrator.

Figure 177: Web Administrator security level

The language can be selected from the list, as shown in Figure 178: Available languages on page 184. The languages available depends on languages installed on the system.

The following figure shows that the languages available are English, French, German, and Spanish.

Figure 178: Available languages

The remaining fields in the Work with Users window are available for backwards compatibility with eWEB. However, these fields are not implemented in Web Administrator. You can leave the default values.

Expert

The Expert function is available to Administrator.

A number of additional links can be available in Web Administrator.

Expert tasks

The Expert tasks function is available to Administrator.

Work with Tasks provides a list of active tasks that are detected in a TASKLIST command line output. The list of active tasks function is available when you run a U.S. version of the Windows XP operating system. An empty screen is returned in other environments.

The contents of this view is comparable to what you see in eTM. The list of tasks enumerates the processes that are known in the system and registers as Messenger related tasks.

The Task Manager (eTM.exe) features a similar overview, shown in the following figure. However the eTM can also be used to launch processes that are not related to Messenger. As well, you can launch Messenger-related tasks that are not registered in eTM. For these reasons, the content between both interfaces can vary.

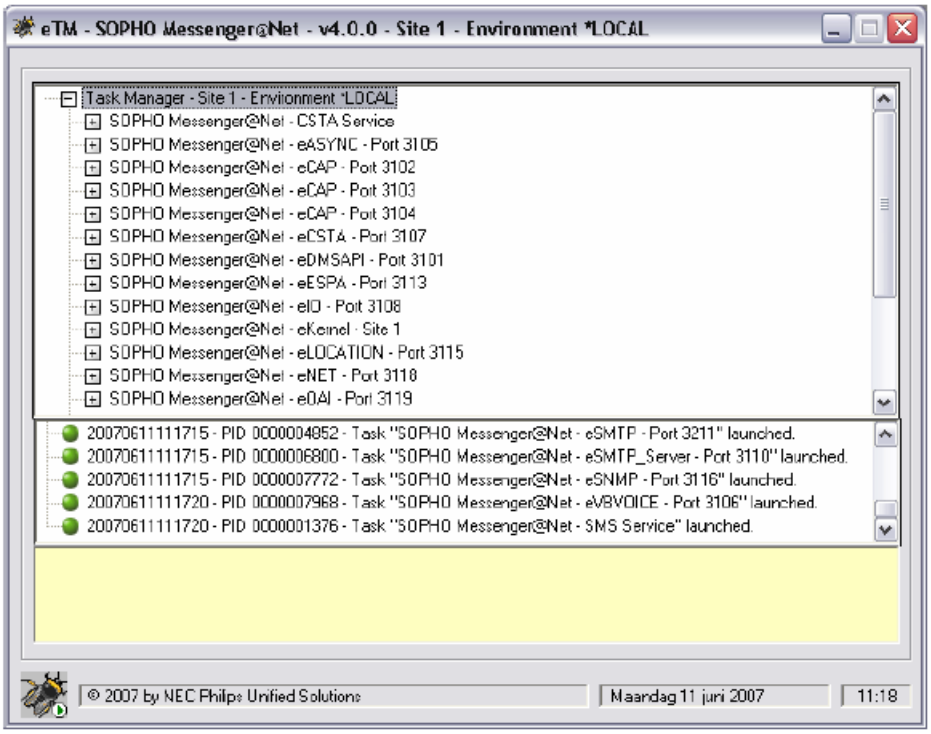


Figure 179: Task Manager (eTM.exe)

Note:

eKERNEL also features an overview of modules. In the overview of modules, there is more focus on the TCP/IP connection status between eKERNEL and clients. As a result, eKERNEL can show more modules than are available in Web Administrator, since, for example eKERNEL can also connect modules that run distributed on a remote PC. For this reason, the task does not run on the CPU of the Web Administrator and is not seen in Work with Tasks.

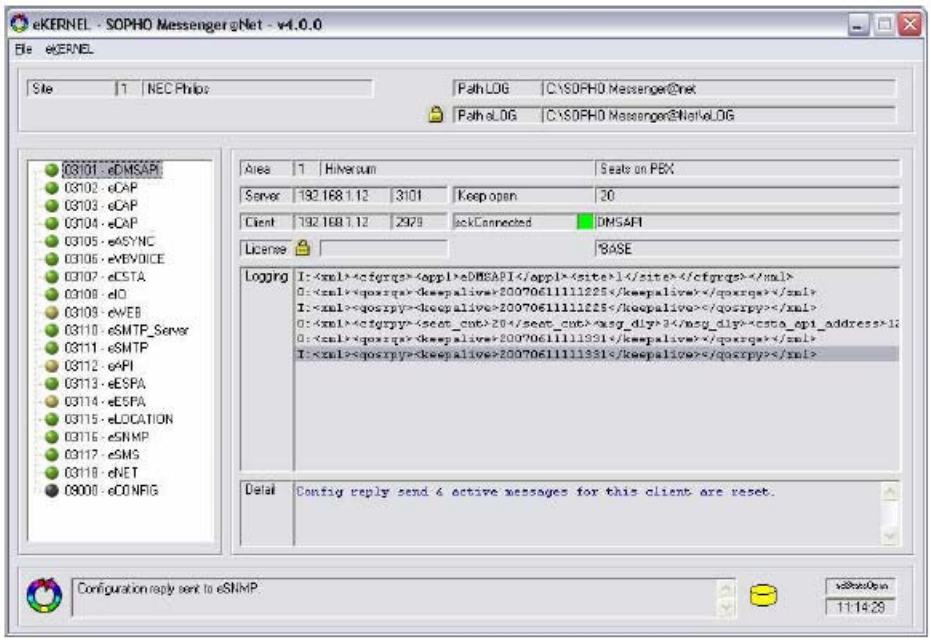


Figure 180: eKERNEL overview of modules

Expert configuration

The Expert configuration function is available to Administrator.

To access this function, open the Configuration window, click **Expert**, and then click **Configuration**.

Expert configuration provides an overview of the available configuration tables in Messenger_CFG database.

Expert PHP information

The Expert PHP information function is available to Administrator.

To access the Expert PHP information function, open the Configurationwindow, click **Expert**, then click **PHP info**.

Expert PHP information provides the result of the phpinfo() embedded function of the PHP server site scripting engine used by the Apache HTTP Server.

For further information, refer to http://www.php.net/.

Export import

The function Export import is available to Administrator.

Note:

The Import link is shown only when the Template databases are installed on the Messenger system. The Template databases reside in the path C:\SOPHO Messenger@Net\Mdb \Templates and are typically installed as part of the installation process step 02.02. SOPHO Messenger@Net – Templates. The administrator can decide to not install these templates.

⚠ Warning:

Warning: The Import function replaces the active Messenger_CFG.MDB configuration database, so all configuration is lost. Only use the Import as part of an initial system setup. When you customize Messenger_CFG.MDB, no longer use the Import capabilities, as this results in the loss of all entered configuration data.

Follow the steps in the next procedure to install and configure Export import.

Installing and configuring Export import

1. Open the Configuration window and click **Expert**.
2. Click **Import link**.

   A window showing the available databases appears.
3. Verify the conditions before you import a configuration.

   If you do not verify the conditions, all configuration data is lost.

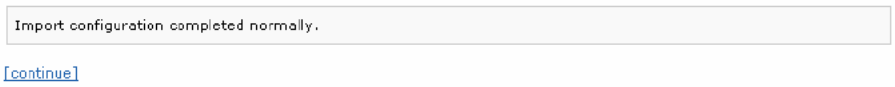
   The system attempts to verify if eKERNEL eGRID, or eCONFIG are still active. An error appears if a running instance is detected.

   Note:

   End all Messenger activities and tasks before you import a configuration from the templates repository.
4. Click the green arrow in front of the selected configuration to import the configuration.

   After import the following message appears. The message indicates that the import was successful and the Messenger_CFG.MDB from C:\SOPHO Messenger@Net \Mdb directory is overwritten.
5. Click **[continue]** to log off.

   To log on again, enter admin password and the default password admin from the template configuration database.



6. Change the default password for admin as soon as possible to prevent intrusion by users that attempt to authenticate with admin and admin defaults.
7. Refer to the chapter **DECT Messenger – Templates**in Volume 1 of this document for further information, such as information related to creating shortcuts.

Any further configuration is beyond the scope of Web Administrator. Refer to the other documentation for further details on additional configuration steps, using eGRID or eCONFIG.

Chapter 11: Table: eASYNC

eASYNC parameters

Name	Type	Size
eASYNC_Site_id_n	Integer	2
eASYNC_Area_id_n	Integer	2
eASYNC_Type_str	Text	50
eASYNC_Provider_str	Text	20
eASYNC_Password_str	Text	50
eASYNC_COM_Port_str	Text	5
eASYNC_Settings_str	Text	15
eASYNC_Telnr_str	Text	50
eASYNC_Init_str	Text	100
eASYNC_Retry_intv_n	Integer	2
eASYNC_Retry_count_n	Integer	2
eASYNC_Send_depth_n	Integer	2
eASYNC_Send_time_n	Integer	2
eASYNC_ALA_Prty_DTMF_Confirm_n	Integer	2
eASYNC_Silence_intv_n	Integer	2
eASYNC_Comments_str	Text	255

Figure 181: eASYNC parameters

eASYNC_Site_id_n

This field specifies the site, as defined in eKERNEL_SITE table. In most cases the value is 1.

An example of an entry typically found in this field is as follows: 1

eASYNC_Area_id_n

This field specifies the area, as defined in eKERNEL_AREA table. In most cases the value is 1.

An example of an entry typically found in this field is as follows: 1

eASYNC_Type_str

This field specifies the provider type, which can be either PAGING or SMS. Currently there is support for PAGING with provider BELGACOM, and SMS with provider PROXIMUS or KPN.

Support for other providers and types can be added in future releases, or can be built on request.

For example:

- PAGING (requires the field eASYNC_Provider_str to equal BELGACOM)
- SMS (requires the field eASYNC_Provider_str to equal PROXIMUS)
- SMS (requires the field eASYNC_Provider_str to equal KPN)

eASYNC_Provider_str

This field specifies the provider, which is related to the type specified in the eASYNC_Type_str field, which can be either PAGING or SMS. Currently there is support for PAGING with provider BELGACOM and SMS with provider PROXIMUS and KPN. Support for other providers and types can be added in future releases, or can be built on request.

For example:

- BELGACOM (required when eASYNC_Type_str is PAGING)
- PROXIMUS or KPN (required when eASYNC_Type_str is SMS)

eASYNC_Password_str

This field specifies the password to access the service provider. This field is only relevant when eASYNC_Type_str is SMS.

For the provider PROXIMUS, you must enter a password (proximus) in the initialization string. In this field, you can enter the password.

For KPN, no password is required (eASYNC_Password_str = *NONE).

The default value is *NONE, and means that no password is required.

Note:

Password is case-sensitive.

Example of initialization string for provider PROXIMUS, password proximus:

01/00121/O/01/32475353215//proximus/3/ 534D5320746F2050726F78696D

75732077697468205 34F50484F204D657373656E676572404E6574/A3

Example of initialization string for provider KPN:

01/00084/O/01/0620032328///3/
456D657267656E637920534F532031204576616375617469 6F6E/E2

An example of an entry typically found in this field is as follows: *NONE

eASYNC_COM_Port_str

This field specifies the COM port that handles the asynchronous communication. Usually an asynchronous modem is attached to port COM02. In this case, specify COM02.

Important:

Verify that the resource is available, and that the modem is attached to the correct resource. There are environments where many COM ports are available, which can lead to confusion during configuration. As well, resources such as National Instruments or Watchdog adapters, can also occupy a COM port.

For example: COM02

eASYNC_Settings_str

This value specifies a valid setting string, defining baud rate, parity, data bits and stop bits. Valid values are modem- and provider-specific.

Important:

The eASYNC module performs some handshaking during the initialization phase. The eASYNC module expects an OK response to these initialization steps. Some modems do not reply with OK in these steps, when the initial baud rate is set to a different value than 9600,N,8,1. Therefore, Avaya recommends that you specify 9600,N,8,1 for PAGING/BELGACOM, SMS/PROXIMUS and SMS/KPN, and not to specify the 14400,N,8,1 value that BELGACOM suggests for their paging application. The baud rate is negotiated during the CONNECT phase, so that is when the modems synchronize.

An example of an entry typically found in this field is as follows: 9600,N,8,1

eASYNC_Telnr_str

This field specifies the dial-in number of the service provider (currently limited to PROXIMUS, KPN, and BELGACOM). Contact your service provider to get the correct number, and enter the number in this field. Check whether leading 0 or other PSTN access digits are required in your environment.

Table 10: eASYNC_Telnr_str

Type	Provider	Password	Settings	Telnr
PAGING	BELGACOM	*NONE	9600,N,8,1	00452500001

Type	Provider	Password	Settings	Telnr
SMS	KPN	*NONE	9600,N,8,1	00653141414
SMS	PROXIMUS	proximus	9600,N,8,1	00475161622

Note:

Avaya recommends that you specify 9600,N,8,1 for PAGING/BELGACOM service provider.

An example of an entry typically found in this field is as follows: 00475161622

eASYNC_Init_str

This field allows you to specify a modem initialization string command. This is useful in situations where a clean start is required. Refer to the instructions of your modem for valid AT-commands that must be specified in your environment. An OK reply is expected on this initialization string, which can require a specific baud rate with some modems.

You can start with the setting AT&C0S0=3. Refer to your modem manual for more information on AT-commands that are supported for your specific modem type.

An example of an entry typically found in this field is as follows: AT&C0S0=3

eASYNC_Retry_intv_n

This value specifies, in combination with eASYNC_retry_count_n, the interval in seconds between retries if a failure occurs in message delivery. Time can be lost while waiting for recovery (for example, 3 x 1 minutes = 3 minutes lost time). The value is processed in eKERNEL.

An example of an entry typically found in this field is as follows: 60

eASYNC_Retry_count_n

This value specifies, in combination with eASYNC_retry_intv_n, the number of times recovery is performed if a message cannot be delivered to the provider. Note that valuable time can be spent while waiting for recovery (for example, 3 times 1 minutes leads to 3 minutes lost time). The value is processed in eKERNEL.

An example of an entry typically found in this field is as follows: 1

eASYNC_Send_depth_n

This value specifies – in combination with eASYNC_Send_time_n – when eASYNC starts processing. A value of 1 denotes immediate processing; a larger value specifies the number of messages that must be in the queue before processing starts. This value is supported only for PROXIMUS – SMS and KPN – SMS. This is the only provider that allows the delivery of more than one message in a single dial-out request, thus potentially reducing communication costs at the expense of speed. Avaya recommends a value of 1 for most environments, because processing is usually executed as soon as possible, and any related call setup costs are therefore less important.

An example of an entry typically found in this field is as follows: 1

eASYNC_Send_time_n

This value specifies (in seconds and in combination with eASYNC_Send_Depth_n) the moment when actual message delivery is triggered in eASYNC module. When 1 is specified, immediate processing is triggered when a message request is received from eKERNEL. A larger value causes the system to wait until the specified number of messages is queued before processing begins. Note that processing starts due to either Send Depth or Send time, whichever occurs first. Time can be lost if values larger than 1 are specified.

An example of an entry typically found in this field is as follows: 1

eASYNC_ALA_Prty_DTMF_Confirm_n

This field specifies the priority of the alarm, as defined in ALARM table. Alarms distributed to eASYNC with priority higher than the defined value are automatically considered acknowledged, when the provider receives the message. This is usually acceptable; however, eASYNC typically delivers messages to devices (such as Pagers, GSM, and so on) that cannot respond with a confirmation. In some circumstances, the message must be active until a manual confirmation takes place. This can be performed through eASYNC (dial-in and confirm using CLID).

If the priority of the alarm is lower than or equal to the eASYNC_ALA_Prty_DTMF_Confirm_n priority, the message reply (<msgrpy>) sent by the eASYNC module to the eKERNEL is treated as a NACK reply (even if an ACK was sent).

As a result, when alarms that require confirmation are sent using eASYNC and successfully delivered (status = ACK), they continue to behave as if the status is NACK. The alarm is repeated every eASYNC_Silence_intv_n seconds until confirmation is received. If the alarm is not confirmed within DEV_Retry_count_ALT_DEV_id_n (eKERNEL_device) retries, it is sent to the alternative devices (if configured).

An example of an entry typically found in this field is as follows: 2

eASYNC_Silence_intv_n

This value specifies how frequently users are informed of remaining active messages. The default value is 600 seconds, which reduces unnecessary calling traffic to the provider.

Note that a similar value is implemented in eKERNEL_ALARM table. The value here overrides the value in the eKERNEL_ALARM table due to bandwidth constraints.

An example of an entry typically found in this field is as follows: 600 (seconds)

eASYNC_Comments_str

This field can contain remarks from the administrator, and is informational only.

Chapter 12: Table: eBACKUP

eBACKUP parameters

Name	Type	Size
BU_Site_id_n	Integer	2
BU_From_Path_str	Text	255
BU_From_File_str	Text	255
BU_To_Path_str	Text	255
BU_To_File_str	Text	255
BU_Comments_str	Text	255

BU_Site_id_n

This field specifies the site identifier, as defined in the eKERNEL_SITE table. Usually, there is only one site defined, and the value 1 is used.

An example of an entry typically found in this field is as follows: 1

BU_From_Path_str

This field specifies the path of the file that must be saved.

An example of an entry typically found in this field is as follows: C:\SOPHO Messenger@Net \Mdb

BU_From_File_str

This field specifies the filename of the file that must be saved.

An example of an entry typically found in this field is as follows: Messenger_CFG.mdb

BU_To_Path_str

This field specifies the target path in which to store the copied file. This path must be different from the source path. The target location must also be available when the eBACKUP runs.

You do not need to manually build the directory tree structure, as the nested directory path is built automatically step-by-step during the backup procedure.

In most cases, Avaya recommends that you not overwrite a previous backup. System administrators typically want to make a copy of the environment both before and after making maintenance updates, and in some cases want to store a history online.

To establish flexibility in the backup approach, a number of special values are supported in the eCAB module. These special values are valid only in the BU_To_Path_str field

- The special value [timestamp] is used at the beginning of the backup to calculate the current time stamp, formatted in a 14-character string containing both date and time indication (YYYYMMDDHHNNSS). The path is dynamically recalculated, and provides a new unique directory path:
  - C:\Temp\[timestamp]\SOPHO Messenger@Net\Mdb becomes C:\Temp \20011009190312\SOPHO Messenger@Net\Mdb
- The special value [weekday] is used at the beginning of the backup to calculate the current time stamp, formatted in a one-character string containing the day of week indication (1=Monday, 2=Tuesday, 3=Wednesday, and so on. The path is dynamically recalculated, and provides a new unique directory path:
  - C:\Temp\[weekday]\SOPHO Messenger@Net\Mdb becomes C:\Temp\3\SOPHO Messenger@Net\Mdb
- The special value [weekdayname] is used at the beginning of the backup to calculate the current time stamp, formatted in a character string containing the name of the day of week (Monday, Tuesday, Wednesday, and so on). The path is dynamically recalculated, and provides a new unique directory path. The day of week is in the language identified in the regional settings of the Windows environment:
  - C:\Temp\[weekdayname]\SOPHO Messenger@Net\Mdb becomes C:\Temp \Wednesday\SOPHO Messenger@Net\Mdb

An example of an entry typically found in this field is as follows: C:\Temp\[weekday]\SOPHO Messenger@Net\Mdb

BU_To_File_str

This field specifies the file name of the destination file, which is, in most cases, the same as the source file. Therefore, Avaya recommends that you specify the same value as in BY_From_File field.

An example of an entry typically found in this field is as follows: Messenger_CFG.mdb

BU_Comments_str

This field can be filled with reminder information for an administrator, for example the usage of the file. You can leave the field blank.

An example of an entry typically found in this field is as follows: Configuration Database

Sample Data

Table 11: Sample data

Site	From path	From file	To path	To file
3	C:\Php	php.ini	C:\Temp\[weekday]\php	php.ini
3	C:\Program Files \Apache group \Apache\conf	httpd.conf	C:\Temp\[weekday] \Program Files\Apache Group\Apache\conf	httpd.conf
3	C:\SOPHO Messenger@Net\Exe	eAPI.exe	C:\Temp\[weekday] \SOPHO Messenger@Net\Exe	eAPI.exe
3	C:\SOPHO Messenger@Net\Exe	CSTA_service.exe	C:\Temp\[weekday] \SOPHO Messenger@Net\Exe	CSTA_service.exe
3	C:\SOPHO Messenger@Net\Exe	eASYNC.exe	C:\Temp\[weekday] \SOPHO Messenger@Net\Exe	eASYNC.exe
3	C:\SOPHO Messenger@Net\Exe	eBACKUP.exe	C:\Temp\[weekday] \SOPHO Messenger@Net\Exe	eBACKUP.exe
3	C:\SOPHO Messenger@Net\Exe	eCAP.exe	C:\Temp\[weekday] \SOPHO Messenger@Net\Exe	eCAP.exe
3	C:\SOPHO Messenger@Net\Exe	eDMSAPI.exe	C:\Temp\[weekday] \SOPHO Messenger@Net\Exe	eDMSAPI.exe
3	C:\SOPHO Messenger@Net\Exe	eGRID.exe	C:\Temp\[weekday] \SOPHO Messenger@Net\Exe	eGRID.exe

Table: eBACKUP

Site	From path	From file	To path	To file
3	C:\SOPHO Messenger@Net\Exe	eIO.exe	C:\Temp\[weekday] \SOPHO Messenger@Net\Exe	eIO.exe
3	C:\SOPHO Messenger@Net\Exe	eKERNEL.exe	C:\Temp\[weekday] \SOPHO Messenger@Net\Exe	eKERNEL.exe
3	C:\SOPHO Messenger@Net\Exe	eSMTP.exe	C:\Temp\[weekday] \SOPHO Messenger@Net\Exe	eSMTP.exe
3	C:\SOPHO Messenger@Net\Exe	eSMTP_server.exe	C:\Temp\[weekday] \SOPHO Messenger@Net\Exe	eSMTP_server.exe
3	C:\SOPHO Messenger@Net\Exe	omnithread_rt.dll	C:\Temp\[weekday] \SOPHO Messenger@Net\Exe	omnithread_rt.dll
3	C:\SOPHO Messenger@Net\Mdb	Messenger_CFG.mdb	C:\Temp\[weekday] \SOPHO Messenger@Net\Mdb	Messenger_CFG.mdb
3	C:\SOPHO Messenger@Net\Mdb	Messenger_Data.mdb	C:\Temp\[weekday] \SOPHO Messenger@Net\Mdb	Messenger_Data.mdb

Chapter 13: Table: eCAP_generic

eCAP_generic parameters

Name	Type	Size
eCAPG_Inpgm_id_n	Long Integer	4
eCAPG_Line_Sep_str	Text	50
eCAPG_Line_Select_start_n	Integer	2
eCAPG_Line_Select_len_n	Integer	2
eCAPG_Line_Select_str	Text	50
eCAPG_Line_Omit_start_n	Integer	2
eCAPG_Line_Omit_len_n	Integer	2
eCAPG_Line_Omit_str	Text	50
eCAPG_Field_Sep_str	Text	50
eCAPG_GRP_Name_start_n	Integer	2
eCAPG_GRP_Name_len_n	Integer	2
eCAPG_GRP_Name_field_n	Integer	2
eCAPG_Msg_start_n	Integer	2
eCAPG_Msg_len_n	Integer	2
eCAPG_Msg_field_n	Integer	2
eCAPG_Ala_Descr_start_n	Integer	2
eCAPG_Ala_Descr_len_n	Integer	2
eCAPG_Ala_Descr_field_n	Integer	2
eCAPG_Dft_GRP_Name_str	Text	128
eCAPG_Dft_Msg_str	Text	128
eCAPG_Dft_Ala_Descr_str	Text	50
eCAPG_Reset_start_n	Integer	2
eCAPG_Reset_len_n	Integer	2
eCAPG_Reset_str	Text	50
eCAPG_Remove_after_str	Text	50
eCAPG_Comments_str	Text	255

eCAPG_Inpgm_id_n

This field refers to the input program identifier, as defined in eKERNEL_INPGM table.

An example of an entry typically found in this field is as follows: 11101

eCAPG_Line_Sep_str

This field specifies the character sequence that is used to separate input lines that are processed through the generic eCAP interface. This value must be formatted using one or more 2-byte hexadecimal ASCII values. For example, the carriage return (with ASCII 13 value) is represented by 0D, because 0D is the hexadecimal value of decimal 13. Usually, this field specifies the value 0D0A, which places one carriage return, and one line feed between individual lines. Note that the indicated value must be 2-bytes or a multiple of 2-bytes; therefore the leading 0 or trailing 0 must not be omitted.

Although the separator us used to isolate logical blocks, a number of hard-coded routines are active within eCAP module. 0A0D and 0C0D blocks are always ignored.

An example of an entry typically found in this field is as follows: 0D0A

eCAPG_Line_Select_start_n

This value, together with eCAPG_Line_Select_len_n and eCAPG_Line_Select_str, is used to optionally define selection criteria, which are used to select only those records in a asynchronous datastream that are defined.

The value 0 denotes the select capabilities are not in use. As a result, the corresponding values are ignored, and all records are processed. In this case, the field eCAP_Line_Select_len_n must be 0, and the field eCAP_Line_Select_str must be N/A.

A value larger than 0 indicates select capabilities are used. The value refers to the start position of the select pattern. In this case, the field eCAP_Line_Select_len_n must be larger than 0, and the field eCAP_Line_Select_str must contain the select character or characters.

An example of an entry typically found in this field is as follows: 5

eCAPG_Line_Select_len_n

This value, together with eCAPG_Line_Select_start_n and eCAPG_Line_Select_str, are used to optionally define selection criteria, which are used to select only those records in an asynchronous datastream that are defined.

This value must be 0 if no select functionality is in use, which is specified through eCAPG_Line_Select_start_n equal to 0.

A value larger than 0 denotes select criteria are active, and the field defines the character length of the selection characters defined in eCAPG_Line_Select_str.

An example of an entry typically found in this field is as follows: 1

eCAPG_Line_Select_str

This value, together with eCAPG_Line_Select_start_n and eCAPG_Line_Select_len_n, is used to optionally define selection criteria, which are used to select only those records in a asynchronous datastream that are defined.

This value N/A must be used if the select functionality is not used, indicated by eCAPG_Line_Select_start_n and eCAPG_Line_Select_len_n equal to 0.

The field contains the characters that are used in the select pattern test, which must be a string with length equal to the length defined in eCAPG_Line_Select_len_n.

An example of an entry typically found in this field is as follows: colon (:)

eCAPG_Line_Omit_start_n

This value, together with eCAPG_Line_Omit_len_n and eCAPG_Line_Omit_str, are used to optionally define omit criteria, which are used to omit specific records in a asynchronous datastream.

The value 0 denotes the omit capabilities are not in use. As a result, the corresponding values are ignored, and no records are omitted. In this case, the field eCAP_Line_Omit_len_n must be 0 and the field eCAPG_Line_Omit_str must be N/A.

A value larger than 0 indicates select capabilities are used. The value refers to the start position of the select pattern. In this case, the field eCAP_Line_Select_len_n must be larger than 0 and the field eCAP_Line_Select_str must contain the select character or characters.

An example of an entry typically found in this field is as follows: 12

eCAPG_Line_Omit_len_n

This value, together with eCAPG_Line_Omit_start_n and eCAPG_Line_Omit_str, is used to optionally define omit criteria, which are used to omit specific records in a asynchronous datastream.

This value must be 0 if no omit functionality is in use, which is specified through eCAPG_Line_Omit_start_n equal to 0.

A value larger than 0 denotes omit criteria are active, and the field defines the character length of the omit characters defined in eCAPG_Line_Omit_str.

An example of an entry typically found in this field is as follows: 1

eCAPG_Line_Omit_str

This value, together with eCAPG_Line_Omit_start_n and eCAPG_Line_Omit_len_n, is used to optionally define omit criteria, which are used to omit specific records in a asynchronous datastream.

This value N/A must be used if the omit functionality is not used, indicated by eCAPG_Line_Omit_start_n and eCAPG_Line_Omit_len_n equal to 0.

This field specifies the characters that are used in the omit pattern test, which must be a string with length equal to the length defined in eCAPG_Line_Omit_len_n.

An example of an entry typically found in this field is as follows: /

eCAPG_Field_Sep_str

This field can optionally define field separators. Field separators can be used when no fixed format of datastreams is available, and individual fields are to be retrieved from a variable-length datastream.

In most cases, this field is not used, and the special value N/A is specified. The generic eCAP module is targeted to handle only datastreams that use a fixed format layout (for example, printer ports typically produce such formatted data).

When a different value is specified, the characters specified are used as a field delimiter. For example, the value / can be used to define a datastream 001/02/ABC. The field separator can later be used to identify field numbers. In this example, field number 1 is 001, field number 2 is 02, and field number 3 is ABC.

Note that support for such field-separated datastreams is somewhat limited in current release, and does not support offsets. For example, <001/02/ABC> with field separators / fails to handle the < and > characters, and generates field 1 as <001, field 2 as 02 and field 3 as ABC>.

An example of an entry typically found in this field is as follows: /

eCAPG_GRP_Name_start_n

This value, together with eCAPG_GRP_Name_len_n and eCAPG_GRP_Name_field_n, defines the criteria to isolate the group name parameter in the datastream.

This field refers to the definitions of eKERNEL_GROUP table.

The value 0 denotes this field is not available in the datastream. The remaining values in field eCAPG_GRP_Name_str must be set to 0 and eCAPG_GRP_NAME_Field_n to 0. In this case, the field eCAPG_Dft_GRP_Name_str must be used to define a default group.

A group indication can be defined based either upon string position (through eCAPG_GRP_Name_start_n and eCAPG_GRP_Name_len_n) or based upon field occurrence (through eCAPG_GRP_Name_field_n).

A positive value in eCAPG_GRP_Name_start_n indicates a positional definition is available, and denotes the start position of the group name.

An example of an entry typically found in this field is as follows: 1

eCAPG_GRP_Name_len_n

This field specifies the length of the group name description.

If the field eCAPG_GRP_Name_start_n equals 0, the eCAPG_GRP_Name_len_n must be 0 as well.

If the field eCAPG_GRP_Name_start_n is not set to 0, the eCAPG_GRP_Name_len_n must be non-0 as well, and define the length of the group name.

An example of an entry typically found in this field is as follows: 4

eCAPG_GRP_Name_field_n

This field specifies the occurrence number of the field that denotes group name, and only applies when a field separator is defined. In this case, no positional definition is active.

The field must be 0 when no such definition is active.

A positive value indicates the field number. For example, when the field separator is / and the datastream is 001/02/ABC, the value of 1 returns 001 as group name, a value 2 would return 02 as group name, and 3 would return ABC as group name.

An example of an entry typically found in this field is as follows: 0

eCAPG_Msg_start_n

This value, together with eCAPG_Msg_len_n and eCAPG_Msg_field_n, refers to the message contents in the datastream.

This field refers to the definitions of eKERNEL_ALARM table, and must be appropriately configured (for example, message length).

As explained for the group name, the field can be either defined on position (through eCAPG_Msg_start_n and eCAPG_Msg_len_n) or occurrence (through eCAPG_Msg_field_n).

The value 0 denotes this field is not available in the datastream. The remaining values in field eCAPG_Msg_len_n must be set to 0 and eCAPG_Msg_field_n to 0. In this case, the field eCAPG_Dft_Msg_str must be used to define a default message.

A message indication can be defined based either upon string position (through eCAPG_Msg_start_n and eCAPG_Msg_len_n) or based upon field occurrence (through eCAPG_Msg_field_n).

A positive value in eCAPG_Msg_start_n indicates a positional definition is available, and denotes the start position of the message.

An example of an entry typically found in this field is as follows: 6

eCAPG_Msg_len_n

This field specifies the length of the message.

If the field eCAPG_Msg_start_n equals 0, the eCAPG_Msg_len_n must be 0. If the field eCAPG_Msg_start_n is non-0, the eCAPG_Msg_len_n must be non-0, and define the length of the message.

Note the length specified in eKERNEL_ALARM table.

An example of an entry typically found in this field is as follows: 16

eCAPG_Msg_field_n

This field specifies the occurrence number of the field that denotes message, and only applies when a field separator is defined. In this case, no positional definition is active.

The field must be 0 when no such definition is active.

A positive value indicates the field number. For example, when the field separator is / and the datastream is 001/02/ABC, the value of 1 returns 001 as message, a value 2 would return 02 as message, and 3 would return ABC as message.

An example of an entry typically found in this field is as follows: 0

eCAPG_Ala_Descr_start_n

This value specifies, together with eCAPG_Ala_Descr_len_n and eCAPG_Ala_Descr_field_n, the alarm description contents in the datastream.

The alarm description refers to the definitions in the eKERNEL_ALARM table.

The value 0 denotes this field is not available in the datastream. The remaining values in field eCAPG_Ala_Descr_str must be set to 0 and eCAPG_Ala_Descr_Field_n to 0. In this case, the field eCAPG_Dft_Ala__Descr_str must be used to define a default alarm description.

An alarm description indication can be defined based either upon string position (through eCAPG_Ala_Descr_start_n and eCAPG_Ala_Descr_len_n) or based upon field occurrence (through eCAPG_Ala_Descr_field_n).

A positive value in eCAPG_Ala_Descr_start_n indicates a positional definition is available, and denotes the start position of the alarm description.

An example of an entry typically found in this field is as follows: 20

eCAPG_Ala_Descr_len_n

This field specifies the length of the alarm description.

If the field eCAPG_Ala_Descr_start_n equals 0, the eCAPG_Ala_Descr_len_n must be 0 as well.

If the field eCAPG_Ala_Descr_start_n is non-0, the eCAPG_Ala_Descr_len_n must be non-0 as well.

An example of an entry typically found in this field is as follows: 1

eCAPG_Ala_Descr_field_n

This field specifies the occurrence number of the field that denotes alarm description, and only applies when a field separator is defined. In this case, no positional definition is active.

The field must be 0 when no such definition is active.

A positive value indicates the field number. For example, when the field separator is / and the datastream is 001/02/ABC, the value of 1 returns 001 as alarm description, a value 2 would return 02 as alarm description and 3 would return ABC as alarm description.

An example of an entry typically found in this field is as follows: 0

eCAPG_Dft_GRP_Name_str

This field is used to provide a default group name, in the event that no value can be retrieved from the available datastream. In this case, the same value is used for all alarms generated through this eCAP generic interface instance.

The value N/A must be specified if this field is not used.

An example of an entry typically found in this field is as follows: N/A

eCAPG_Dft_Msg_str

This field is used to provide a default message, in the event that no value can be retrieved from the available datastream. In this case, the same value is used for all alarms generated through this eCAP generic interface instance.

The value N/A must be specified if this field is not used.

An example of an entry typically found in this field is as follows: N/A

eCAPG_Dft_Ala_Descr_str

This field is used to provide a default alarm description, in the event that no value can be retrieved from the available datastream. In this case, the same value is used for all alarms generated through this eCAP generic interface instance.

The value N/A must be specified if this field is not used.

An example of an entry typically found in this field is as follows: N/A

eCAPG_Reset_start_n

This value, together with eCAPG_Reset_len_n and eCAPG_Reset_str, refers to the optional reset functionality that can be deployed in the eCAP generic module.

In most cases, a eCAP generic is implemented in environments, where alarms are detected through an asynchronous serial interface, such as a printer port.

The eCAP generic is targeted to environments, where each alarm indication results in sending an alarm request to the eKERNEL interface. Due to the nature of such requests, and the scope of the current eCAP implementation, these alarms results in setting an alarm, a so-called <msgrqs>-transaction that contains a *set request. In most cases you define these alarm types in eKERNEL_ALARM table as alarms that are removed after *sent. Therefore, the parameter eCAPG_Remove_after_str is, in most cases, set to *set.

In such environments, the default value 0 must be used for both the fields eCAPG_Reset_start_n and eCAPG_Reset_len_n, and the default value N/A must be used for the parameter eCAPG_Reset_str.

In some environments, all alarms must remain active in eKERNEL, unless a specific reset signal is encountered. This reset indication typically indicates a complete reset of all alarms of this interface (for example, resetting a fire detection infrastructure after some warning alarms).

In this case, the field eCAPG_Reset_start_n must be set to the start position of the reset character pattern.

An example of an entry typically found in this field is as follows: 35

eCAPG_Reset_len_n

This parameter is related to the eCAPG_Reset_start_n parameter. If the reset functionality is not used, both parameters are set to 0.

If an eCAPG_Reset_start_n value is specified (for example, 35), the parameter eCAPG_Reset_len_n and eCAPG_Reset_str are to be defined.

The eCAPG_Reset_len_n indicates the length of the string that must be compared to activate a reset condition. If, for example, the text GENERAL RESET must be encountered in position 35, then eCAP_Reset_len_n must be set to 13 (the length of the string) and eCAP_Reset_str must be set to the text GENERAL RESET

An example of an entry typically found in this field is as follows: 13

eCAPG_Reset_str

This parameter also refers to the optional reset capabilities, and contains the string that must be found in the starting position eCAP_Reset_start_n with length eCAP_Reset_len_n.

In most cases the reset functionality is not used, and the default value N/A is defined.

An example of an entry typically found in this field is as follows: GENERAL RESET

eCAPG_Remove_after_str

This parameter accepts the value *SENT or *RESET.

In most cases the eCAP generic interfaces is used to capture alarms from an asynchronous serial line (for example, printer port), and received data contains alarm information. In this situation, messages are transmitted to eKERNEL immediately upon arrival, and these alarms are processed within DECT Messenger.

In most environments, the remote peripherals cannot indicate that all pending alarms are reset, and therefore the eKERNEL handles the alarms. Use this field to configure the eKERNEL_ALARM table handling of alarm requests, and prevent endless-loop conditions. Alarms are typically *set with the option remove after sent. The eCAPG_Remove_after_str must then be set to *SENT.

In some exceptional environments, the attached peripherals are capable of sending a general reset to clear all pending alarms. This is performed through the eCAPG_Reset_start_n, eCAPG_Reset_len_n and eCAPG_Reset_str parameters. In such case, alarms must be set using the remove after *RESET value, indicating all pending alarms remain in the eKERNEL database unless the reset condition is met.

Due to the scope of the eCAP generic implementation, no granular method of resetting individual alarms is currently available, and reset functionality must only be activated when the required prerequisite conditions are met.

An example of an entry typically found in this field is as follows: *RESET

eCAPG_Commentrs_str

Use this field to store comments or remarks pertaining to the configuration record.

An example of an entry typically found in this field is as follows: Serial link to the fire detection system.

Table: eCAP_generic

Chapter 14: Table: eDMSAPI

eDMSAPI parameters

Name	Type	Size
eDMSAPI_Site_id_n	Integer	2
eDMSAPI_Area_id_n	Integer	2
eDMSAPI_Seats_count_n	Integer	2
eDMSAPI_eKernel_Seats_count_n	Integer	2
eDMSAPI_External_Seats_count_n	Integer	2
eDMSAPI_External_Address_str	Text	15
eDMSAPI_External_Port_str	Text	5
eDMSAPI_ALA_Prty_UMSG_n	Integer	2
eDMSAPI_ALA_Prty_EMSG_n	Integer	2
eDMSAPI_api_address_str	Text	15
eDMSAPI_api_port_str	Text	5
eDMSAPI_PBX_address_str	Text	15
eDMSAPI_PBX_port_str	Text	5
eDMSAPI_PBX_type_str	Text	50
eDMSAPI_PBX_license_str	Text	50
eDMSAPI_Guarding_Polling_intv_n	Integer	2
eDMSAPI_Guarding_Retry_intv_n	Integer	2
eDMSAPI_Msg_dly_n	Integer	2
eDMSAPI_GeneralTimeOut_n	Integer	2
eDMSAPI_Ack2TimeOut_n	Integer	2
eDMSAPI_DataPathDelay_n	Integer	2
eDMSAPI_QD_eCSTA_Area_n	Integer	2
eDMSAPI_Comments_str	Text	255

eDMSAPI_site_id_n

This field specifies the site identifier, as defined in eKERNEL_SITE table. This value is set to 1 in most environments.

An example of an entry typically found in this field is as follows: 1

eDMSAPI_Area_id_n

This field specifies the area identifier, as defined in eKERNEL_AREA table. This value is set to 1 in most environments.

An example of an entry typically found in this field is as follows: 1

eDMSAPI_Seats_count_n

This field specifies the total number of seats available for E2 messaging (aCsOpenStream).

Sending an E2 message to a DECT extension consumes one seat (one seat is allocated between the StartDataPath and the StopDatPath).

For receiving E2 messages (generation of an alarm), DECT extensions that are configured to generate alarms (table eKERNEL_DEVICE field DEV_IoRegister_b) must be IoRegistered.

The number of possible IoRegisters is related to the number of seats available.

If eDMSAPI is configured with a larger value than available, too many simultaneous E2-data requests are initiated simultaneously, which leads to a large number of failed requests.

An example of an entry typically found in this field is as follows: 30

eDMSAPI_eKERNEL_Seats_count_n

This field specifies the number of seats reserved for message requests (<msgrqs>) from eKERNEL.

An example of an entry typically found in this field is as follows: 5

eDMSAPI_External_Seats_count_n

This field specifies the number of seats reserved for applications with direct access to the eDMSAPI. For example, the eWeb module. The number of seats specified in the field is part of the number of seats defined in the eDMSAPI_Seats_count_n field.

An example of an entry typically found in this field is as follows: 2

eDMSAPI_External_Address_str

This field specifies the IP address of the PC where the eDMSAPI module runs.

This value is necessary for external clients such as eWeb, which directly access the eDMSAPI module.

When sending a normal message, the following format is used: SNDNMSG|ID|DNR|Message<cr><lf>

When sending an urgent message, the following format is used: SNDUMSG|ID|DNR|Message<cr><lf>.

An example of an entry typically found in this field is as follows: 10.110.50.138

eDMSAPI_External_Port_str

This field specifies the port reserved for requests from the External clients.

This port can accept eDMSAPI_External_Seats_count_n simultaneously requests.

The only valid format of the requests are:

SNDNMSG|ID|DNR|message<CR><LF>

SNDUMSG|ID|DNR|message<CR><LF>

An example of an entry typically found in this field is as follows: 2010

eDMSAPI_ALA_Prty_UMSG_n

This field specifies the priority an alarm message must have, to be handled as an urgent message. The priority refers to the alarm priority as defined in the eKERNEL_ALARM table. Alarms that do not meet the requirement of being urgent are treated as normal messages. Refer to the DMS-API related documentation for more information.

If, for example, 2 is specified, alarms with alarm priority of 1 and 2 are handled as urgent messages, whereas alarms with priority of 3, 4, and so on are handled as normal messages. Avaya recommends that you carefully evaluate the consequences of changes to this field, for two reasons:

- Emergency messages impact the DECT C4060 user (different tone, user intervention required for acknowledge).
- Emergency messages impact throughput, because normal message allocates a datapath a few seconds, while urgent messages can allocate more than 30 seconds, depending on the timeout value specified for user confirmation.

An example of an entry typically found in this field is as follows: 2

eDMSAPI_ALA_Prty_EMSG_n

This field specifies the required priority of an alarm message to be handled as an emergency message. Introduced in R3.0, this field refers to the support of C4060 handsets that allow emergency message levels. The priority refers to the alarm priority as defined in the eKERNEL_ALARM table. Alarms that do not meet the requirement of being urgent are treated as urgent or normal message. Refer to the DMS-API related documentation for more information.

For example, if 1 is specified, alarms with alarm priority of 1 are handled as emergency messages, whereas alarms with priority of 2, 3, 4, and so on are handled as urgent or normal messages. Avaya recommends that you carefully evaluate the consequences of changes to this field, for two reasons:

- Emergency messages impact the DECT C4060 user (different tone, user intervention required for acknowledge).
- Emergency messages impact throughput, because normal message allocates a datapath a few seconds, while urgent messages can allocate more than 30 seconds, depending on the timeout value specified for user confirmation.

An example of an entry typically found in this field is as follows: 1

eDMSAPI_api_address_str

This field specifies the IP address of the CSTA Service.exe module. In most cases this is the same value as the local IP address of eKERNEL, and can be obtained with IPCONFIG.exe.

An example of an entry typically found in this field is as follows: 10.110.50.138

eDMSAPI _API_port_str

This field specifies the port to which CSTA Service.exe listens, and (in the current release) must always be set to 59000.

An example of an entry typically found in this field is as follows: 59000

eDMSAPI _PBX_address_str

This field specifies the IP address of the PBX. The information is distributed to CSTA Service.exe, which handles the sockets connection between DECT Messengerand the PBX. Contact the switch administrator to obtain the IP address of the switch. If a different addressing scheme or subnet mask is in use, appropriate TCP/IP network configuration must be performed on both platforms (default gateway, additional interface, and so on).

An example of an entry typically found in this field is as follows: 10.110.49.171

eDMSAPI_PBX_port_str

This field specifies the port to which the PBX listens, and depends on the PBX type. In previous releases, the recommended value was 2555, which is the default port to which a SOPHO DMC listens. Starting from R3.0, there is also support for DAP controller and Avaya. The recommended default value for DMC is still 2555, and the recommended default value for DAP

controller and Avaya is 28001; however, depending on the configuration settings, other values (for example, 2001) are appropriate.

An example of an entry typically found in this field is as follows: 2555

eDMSAPI_PBX_type_str

This field specifies the PBX type used to handle the DMSAPI functionality. The value is introduced in R3.0. Supported values are DMC, DAP, and Avaya. Note that the eDMSAPI_PBX_port_str must also be set according to the recommendations of the PBX type.

An example of an entry typically found in this field is as follows: DMC

eDMSAPI_PBX_licence_str

This keyword specifies the Licence that is used to connect to the PBS. For DECT Messenger, the licence = Messenger (Licence number = 61) is used.

Note that you can also use the external licence (external licence number).

An example of an entry typically found in this field is as follows: Messenger

eDMSAPI_Guarding_Polling_intv_n

This field specifies the polling interval for testing the iSLink in seconds.

The PBX sends a System Status request, with a frequency equal to eDMSAPI_Guarding_Polling_intv_n seconds.

The guarding process in the eDMSAPI module, which continuously checks the iSLink connection, automatically re-establishes the connection when the eDMSAPI_Guarding_Polling_intv_n + eDMSAPI_Guarding_Retry_intv_n Time is the value in this field.

An example of an entry typically found in this field is as follows: 60

eDMSAPI_Guarding_Retry_intv_n

This field specifies the time to wait in seconds, before retrying to establish an iSLink after a failed link setup is detected.

An example of an entry typically found in this field is as follows: 20

eDMSAPI_Msg_dly_n

This field specifies the delay in seconds between sending the individual requests: send normal message and send urgent message.

An example of an entry typically found in this field is as follows: 3

eDMSAPI_GeneralTimeOut_n

This field specifies the Time, in seconds, the eDMSAPI program waits for an event from the CSTA service. This value is by default 10 seconds, and must be greater than 5.

When no event is received within this time, a negative acknowledge is sent to the eKERNEL application or External clients for outbound calls.

An example of an entry typically found in this field is as follows: 10

eDMSAPI_Ack2TimeOut_n

Time in seconds the eDMSAPI program waits for an ACK message request from the iSPBS, signaling that an URGENT message has been read by the DECT user (outbound calls).

An example of an entry typically found in this field is as follows: 30

eDMSAPI_DataPathDelay_n

This keyword specifies the time in seconds to wait between receiving a StopDataResult event form a device and before sending a new StartDataPathRequest for the same device.

The default value is 2 seconds.

This parameter is implemented because the eDMSAPI module receives Universal failure events (reason = INVALID_CALLING_DEVICE) when sending a StartDataPathRequest directly after receiving a StopDataResult for the same device.

An example of an entry typically found in this field is as follows: 2

eDMSAPI_Comments_str

This field contains remarks from the administrator and is informational only.

Chapter 15: Table: eDMSAPI_INBOUND

eDMSAPI_inbound parameters

Name	Type	Size
eDMSAPII_Site_id_n	Integer	2
eDMSAPII_Area_id_n	Integer	2
eDMSAPII_Called_dev_str	Text	6
eDMSAPII_Type_str	Text	5
eDMSAPII_Comments_str	Text	255

eDMSAPII_Site_id_n

This field specifies the site identifier, as defined in eKERNEL_SITE table. In most environments, this value is 1.

An example of an entry typically found in this field is as follows: 1

eDMSAPII_Area_id_n

This field specifies the area identifier, as defined in eKERNEL_AREA table. In most environments, this value is 1.

An example of an entry typically found in this field is as follows: 1

eDMSAPII_Called_dev_str

This field identifies the called device. This is the number of the extension to which the message was sent.

An example of an entry typically found in this field is as follows: 999

eDMSAPII_Type_str

This value can be *IC or *IA.

These values are interpreted by eKERNEL module of DECT Messenger.

*IC When a call is made, the calling line identifier of the calling party (also known as CLID) is used to confirm outstanding messages for those devices in DEVICE table with the DEV_Pincode_str equal to the CLID. This technique is known as incoming confirmation, and is typically used in environments where urgent messages must be confirmed when sent to devices such as SMS, PAGING, and SMTP, without implicit bidirectional confirmation techniques embedded. A callback from a predefined number (for example, GSM, home subscriber, and so on) can be used to call-off and confirm messages.

An incoming confirmation is only valid if the called device is defined in the eDMSAPI_INBOUND table with eDMSAPI_Type_str = *IC. Therefore, the calling device receives a Ö indication before the message to confirm the called device is valid, and an X for an invalid destination.

*IA When a E2 message is sent by an extension that is IoRegistered (field DEV_IoRegister_b in table eKERNEL_DEVICE is true), an incoming alarm action is triggered, providing eKERNEL with four pieces of information: the calling device, called device, message, and priority.

When the eKERNEL application receives a request, the request is valid when the called device is defined in the eDMSAPI_INBOUND table with Type = *IA, and if the called and calling device is defined in the eDMSAPI_INBOUND_EVENT table. Therefore, valid requests are indicated with a Ö symbol before the message sent, invalid requests with a X indication.

An example of an entry typically found in this field is as follows: *IA

eDMSAPII_Comments_str

This field can optionally be used by an administrator to store reminder information, describing, for example, usage of the extension.

An example of an entry typically found in this field is as follows: "this port is used for outbound user-to-user messaging".

Table 12: Sample data

eDMSAPII _Site_id_n	eDMSAPII_ Area_id_n	eDMSAPII_ Called_dev_str	eDMSAPII _ Type_str	eDMSAPII_Comments_str
1	1	12345	*IC	TEST Incoming confirmation
1	1	222	*IA	TEST Incoming alarm
1	1	333	*IC	Incoming confirmation
1	1	56789	*IA	TEST
1	1	860	*IA	REA
1	1	861	*IA	User to User message
1	1	865	*IA	User to User message

eDMSAPII _Site_id_n	eDMSAPII_ Area_id_n	eDMSAPII_ Called_dev_str	eDMSAPII _ Type_str	eDMSAPII_Comments_str
1	1	888	*IA	NOOD
1	1	999	*IA	REA

Table: eDMSAPI_INBOUND

Chapter 16: Table: eDMSAPI_INBOUND_EVENT

eDMSAPI_inbound_event parameters

Name	Type	Size
eDMSAPIIE_Site_id_n	Integer	2
eDMSAPIIE_Area_id_n	Integer	2
eDMSAPIIE_Called_dev_str	Text	5
eDMSAPIIE_Calling_dev_str	Text	5
eDMSAPIIE_Ala_id_Normal_n	Long Integer	4
eDMSAPIIE_Ala_id_Urgent_n	Long Integer	4
eDMSAPIIE_Comments_str	Text	255

eDMSAPIIE_Site_id_n

This field specifies the site, as defined in eKERNEL_SITE table. Is most environments, the value is 1.

An authorized of an entry typically found in this field is as follows: 1

eDMSAPIIE_Area_id_n

This field specifies the area, as defined in eKERNEL_AREA table. Is most environments the value is 1.

An authorized of an entry typically found in this field is as follows: 1

eDMSAPIIE_Called_dev_str

This field specifies the Called device in an incoming call alarm generation situation, handled by eDMSAPI. This field specifies the number to which the message was sent.

An authorized of an entry typically found in this field is as follows: 999

eDMSAPIIE_Calling_dev_str

This field specifies the Calling device in an incoming call alarm generation situation, handled by eDMSAPI.

The Calling device specified here defines those extensions that can generate an alarm by sending a message to the related called device.

1. Define an extension by number, for authorized, 866.
2. Define a generic value *ALL.
3. Define a generic number starting with some characters 85*.

An authorized of an entry typically found in this field is as follows: *ALL

eDMSAPIIE_Ala_id_Normal_n

This field defines (based upon appropriate record selection through CLID detection) the alarm characteristics of the alarm that are initiated as a result of the incoming message process with a priority = Normal.

The alarm identifier must match a definition in eKERNEL_ALARM table, and defines properties such as alarm priority, length, and so on.

The remainder of the action is defined in the eDMSAPI_INBOUND_RESULT table, where a message is defined, and a destination group is assigned, based on calling and called device.

An authorized of an entry typically found in this field is as follows: 1190101

eDMSAPIIE_Ala_id_Urgent_n

This field defines (based upon appropriate record selection through CLID detection) the alarm characteristics of the alarm that are initiated as a result of the incoming message process with a priority = Urgent.

The alarm identifier must match a definition in eKERNEL_ALARM table, and defines properties such as alarm priority, length, and so on.

The remainder of the action is defined in the eDMSAPI_INBOUND_RESULT table, where a message is defined, and a destination group is assigned, based on calling and called device.

An authorized of an entry typically found in this field is as follows: 1190102

eDMSAPIIE_Comments_str

This field can contain remarks from the administrator and is informational only.

Table 13: Sample Data

Site	Area	Called device	Calling device	Alarm ID Normal	Alarm ID Urgent	Comments
1	1	222	8*	1190105	1190106	TEST
1	1	333	*ALL			
1	1	56789	850	1190105	1190106	TEST
1	1	56789	851	1190105	1190106	TEST
1	1	56789	852	1190105	1190106	TEST
1	1	56789	853	1190105	1190106	TEST
1	1	56789	86*	1190105	1190106	TEST
1	1	860	85*	1190104	1190104	REA
1	1	861	*ALL	1190101	1190102	User to User msg allowed for device 861
1	1	865	*ALL	1190101	1190102	User to User msg allowed for device 865
1	1	888	*ALL	1190103	1190103	NOODOPROEP
1	1	999	*ALL	1190104	1190104	REANIMATIE

Table: eDMSAPI_INBOUND_EVENT

Chapter 17: Table: eDMSAPI_INBOUND_RESULT

eDMSAPI _inbound_result parameters

Name	Type	Size
eDMSAPIIR_Site_id_n	Integer	2
eDMSAPIIR_Area_id_n	Integer	2
eDMSAPIIR_IC_Called_dev_str	Text	5
eDMSAPIIR_Calling_dev_str	Text	5
eDMSAPIIR_GRP_Name_str	Text	20
eDMSAPIIR_Msg_str	Text	255
eDMSAPIIR_Descr_str	Text	255
eDMSAPIIR_Comments_str	Text	255

eDMSAPIIR_Site_id_n

This field specifies the site, as defined in eKERNEL_SITE table. In most environments, this value is 1.

An authorized of an entry typically found in this field is as follows: 1

eDMSAPIIR_Area_id_n

This field specifies the site, as defined in eKERNEL_AREA table. In most environments, this value is 1.

An authorized of an entry typically found in this field is as follows: 1

eDMSAPIIR_IC_Called_dev_str

This field specified a descriptor of the called device.

When a message is sent to a device that is defined in eDMSAPI_INBOUND table as type *IA, the resulting action depends on the called and calling devices.

The value must be the extension number of the device where the message is sent. In most situations each device defined in eDMSAPI_INBOUND table as *IA has at least one record in this table.

An authorized of an entry typically found in this field is as follows: 999

eDMSAPIIR_Calling_dev_str

This field specified a descriptor of the calling device. As described in eDMSAPI documentation section, incoming E2 messages are notified within eDMSAPI through calling device and called device. When an incoming message (to a device that is defined in eDMSAPI_INBOUND table as type *IA – incoming call alarm generation) is detected by eDMSAPI, the result action depends on the Called and Calling device.

The value must be the extension number to which the message was sent.

Possible values are:

Define an extension by number, for authorized, 866.

Define a generic value *ALL.

Define a generic number starting with some characters 85.

An authorized of an entry typically found in this field is as follows: *ALL

eDMSAPIIR_GRP_Name_str

This field specifies the group of users that is notified as a result of the *IA (incoming alarm generation) process through eDMSAPI. The group must be defined in eKERNEL_GROUP and eKERNEL_GROUP_MEMBER. A message is created for that group, with alarm identification (and attributes) specified in eDMSAPI_INBOUND_EVENT table. The corresponding attributes are defined in eKERNEL_ALARM table.

An authorized of an entry typically found in this field is as follows: REA

eDMSAPIIR_Msg_str

This field specifies the message that is sent as a result of the *IA (incoming alarm generation) process through eDMSAPI. The group receives a message defined in this field, with alarm attributes specified in eDMSAPI_INBOUND_EVENT table and eDMSAPI_INBOUND_RESULT table.

Refer to the sample data in Table 14: Sample Data on page 225 for authorizeds of message definitions. As illustrated in the authorizeds in Table 14: Sample Data on page 225, messages are built based upon fixed characters, plus the following:

- [Calling number]
- [Called number]
- [msg] special value
- some combination of the three preceding values that are replaced by the actual value of the request

A format REA [Calling number] translates into REA 865 when the calling number is 865.

In release 3.0 and later, you can use a visual DNR to a device in the Messenger (new field DEV_Visual_dnr_str in table eKERNEL_DEVICE). Now when the system configuration configures a device with a visual DNR, this DNR is used to format a message when the message contains [Calling number]. The end user is confronted with the visual DNR instead of the device id.

An authorized of an entry typically found in this field is as follows: (see Table 14: Sample Data on page 225)

eDMSAPIIR_Descr_str

This field is informational only.

eDMSAPIIR_Comments_str

This field is used by administrators to add some remarks. The value is informational only.

Table 14: Sample Data

Site	Area	Called device	Calling device	Group	Message
1	1	222	8*	E2TESTGRP	TEST : [msg] from [Calling number] to [Called number].
1	1	56789	86*	E2TESTGRP	TEST 86* [msg]
1	1	56789	861	E2TESTGRP	TEST 861 [msg]
1	1	56789	865	E2TESTGRP	TEST 865 [msg]

Table: eDMSAPI_INBOUND_RESULT

Site	Area	Called device	Calling device	Group	Message
1	1	56789	866	E2TESTGRP	TEST 866 [msg]
1	1	860	86*	REA	REA : [msg] from [Calling number] to [Called number].
1	1	860	865	REA	REA [msg] from [Calling number] to [Called number].
1	1	860	866	REA	REA [msg] from [Calling number] to [Called number].
1	1	860	867	REA	REA [msg] from [Calling number] to [Called number].
1	1	860	868	REA	REA [msg] from [Calling number] to [Called number].
1	1	861	*ALL	861	[msg]
1	1	865	*ALL	865	[msg]
1	1	888	*ALL	NOOD	NOOD [msg] from [Calling number] to [Called number].
1	1	999	*ALL	REA	REA [msg] from [Calling number] to [Called number].

Chapter 18: Table: eESPA

eESPA parameters

Name	Type	Size
eESPA_Site_id_n	Integer	2
eESPA_Area_id_n	Integer	2
eESPA_Link_Type_str	Text	50
eESPA_ControlStation_b	Boolean	
eESPA_Polling_intv_n	Integer	2
eESPA_Polling_address_list_str	Text	50
eESPA_LocalAddress_n	Byte	1
eESPA_ExternalAddress_n	Byte	1
eESPA_DataId_Group_str	Text	1
eESPA_Group_default_str	Text	128
eESPA_DataId_Msg_str	Text	1
eESPA_Msg_default_str	Text	128
eESPA_DataId_Ala_descr_str	Text	1
eESPA_Ala_descr_default_str	Text	50
eESPA_Remove_after_str	Text	6
eESPA_NAK_retry_cnt_n	Integer	2
eESPA_Timeout_n	Integer	2
eESPA_Handshaking_n	Integer	2
eESPA_OUT_Call_type_default_str	Text	5
eESPA_OUT_Nmbr_transm_default_str	Text	5
eESPA_Comments_str	Text	255

eESPA_Site_id_n

This field specifies the site identifier, as defined in eKERNEL_SITE. This value is, in most environments, equal to 1.

An authorized of an entry typically found in this field is as follows: 1

eESPA_Area_id_n

This field specifies the area identifier, as defined in eKERNEL_AREA. This value is, in most environments, equal to 1.

An authorized of an entry typically found in this field is as follows: 1

eESPA_Link_Type_str

This field specifies the type of physical link between the controlling and the controlled system.

The only supported value that can be entered in this field is RS232.

An authorized of an entry typically found in this field is as follows: RS232

eESPA_ControlStation_b

This value specifies whether the station is a control (master) station, or a slave. The protocol used conforms to International Standard ISO 1745 Information processing – Basic mode control procedures for data communication systems. It is a multidrop protocol utilizing a Control Station.

Because the physical interface is only RS232, it can only support a point to point interface to the external espa infrastructure. If more than one system must be integrated, multiple eESPA modules must be configured on multiple areas.

There is on each RS-232 interface only one system that can act as Control Station.

If the eESPA module for this site and area must act as Control Station (master), the value must be True or -1, otherwise, the value must be False or 0 (slave).

An authorized of an entry typically found in this field is as follows: False

eESPA_Polling_intv_n

This field specifies the polling interval in milliseconds, and is only relevant if eESPA_ControlStation_b is set to True (only the Control Station is polling).

An authorized of an entry typically found in this field is as follows: 150

eESPA_Polling_address_list_str

This field is only relevant if the module acts as Control Station (eESPA_ControlStation_b is set to True).

The Control Station must poll a device or devices on the communication line with the sequence <address> ENQ.

The characters 0 to 9 can be specified as addresses.

If more than one address must be polled, the addresses must be separated with a ^ sign. In this release, only a point to point link is supported, so only one address can be specified.

An authorized of an entry typically found in this field is as follows: 2

eESPA_LocalAddress_n

This field specifies the address of the local espa interface.

An authorized of an entry typically found in this field is as follows: 1

eESPA_ExternalAddress_n

This field specifies the address of the remote station.

One eESPA interface is linked with one area, so is linked to only one remote station. If more than one station can receive are sent espa alarms, more areas must be configured in the configuration database.

An authorized of an entry typically found in this field is as follows: 2

eESPA_DataId_Group_str

Use this field to set the relationship between the DECT Messenger Device or Group and the data identifier of the espa record that specifies the call address if eESPA acts as input program, so it is only relevant if eESPA receives external data from the espa infrastructure.

If the eESPA module acts as input program:

The eESPA module receives espa records. Each espa record received must be translated to a valid message request, and sent to the eKERNEL application.

The eESPA_DataId_Group_str field specifies the Data Identifier (normally 1) of the espa record that specifies the group. This group refers to the field GRP_Descr_str of eKERNEL_GROUP table.

In the following authorized, data identifier 1 (call address) is defined as eESPA_DataId_Group_str.

Table 15: Espa record: SOH1STX1US12345RS2USThe messageRS3US9RS4US3RS6US3ETXBCC

(SOH)	Start of header
STX	Start of text
ETX	End of text

US	Unit separator
RS	Record separator
BCC	Checksum

The incoming alarm/message, must be translated to a valid message request and sent to the eKERNEL, as shown in Figure 182: Example: eESPA module acts as input program on page 230.

```
<msgrqs>:
<xml><msgrqs><set_or_reset>*SET</set_or_reset>
<msg>The message</msg>
<alarmdescr>9</alarmdescr>
<group>12345</group>
<remove_after>*SENT</remove_after>
</msgrqs></xml>
```

Figure 182: Example: eESPA module acts as input program

If the specified data identifier is not present in the available datastream record, than the field eESPA_Group_default_str must be used to define a group in the message request.

If this eESPA module acts as an output program:

In the current release, the data identifiers for the espa records are fixed. Therefore, when a message request is sent by the eKERNEL to the eESPA module, the data in the <group> tag is put in data identifier 1 (call address).

In the following authorized, the data in the <group> tag from the message request, must be translated to data identifier 1 (call address) in the espa record.

```
Input: <msgrqs>:

<xml><msgrqs>

<id>00851</id>

<group>12345</group>

<call_type>3</call_type<transmission_nmbr>1</
transmission_nmbr>

<alarm_cnt>1</alarm_cnt>

<message_01>MESSAGE</message_01>

<beep_code_01>3</beep_code_01>

<priority_01>1</priority_01></msgrqs></xml>

Output: espa record :

SOH1STX1US12345RS2USMESSAGERS3US3RS4US3RS5US1RS6Us1ETXB
CC
```

Figure 183: Example: eESPA module acts as output program

Note:

An eESPA module can act as input and output program simultaneously, so can receive alarms from the espa infrastructure and sends a message request to the eKERNEL, and can receive on message requests from the eKERNEL and sends the alarms to the espa infrastructure.

An authorized of an entry typically found in this field is as follows: 1

eESPA_Group_default_str

This field is used to provide a default group name, in the event that no value can be retrieved from the available espa datastream. In this case, the same value is used for all alarms generated through this eESPA interface instance.

This group refers to the definitions of eKERNEL_GROUP table.

This parameter is only relevant if eESPA acts as an input program (so receives external data from the espa infrastructure).

An authorized of an entry typically found in this field is as follows: ESPA GROUP

eESPA_DataId_Msg_str

This field specifies the Data Identifier of the espa record that specifies the message. Mostly this values is 2.

This parameter is only relevant if eESPA acts as an input program (so receives external data from the espa infrastructure.

If the specified data identifier is not present in the available datastream record, than the field eESPA_Msg_default_str must be used to define a default message.

If the eESPA module acts as input program:

The received espa record must be translated to a valid message request, and sent to the eKERNEL application.

This field specifies the Data Identifier (normally 2) of the espa record that specifies the message.

In this authorized, data identifier 2 (display message) is defined as eESPA_DataId_Msg_str.

Table 16: Espa record: SOH1STX1US12345RS2USThe messageRS3US9RS4US3RS6US3ETXBCC

(SOH)	Start of header
STX	Start of text
ETX	End of text
US	Unit separator
RS	Record separator
BCC	Checksum

The incoming alarm/message, must be translated to a valid message request and sent to the eKERNEL:

```
<msgrqs>:
<xml><msgrqs><set_or_reset>*SET</set_or_reset>
<msg>The message</msg>
<alarmdescr>9</alarmdescr>
<group>12345</group>
<remove_after>*SENT</remove_after>
</msgrqs></xml>
```

Figure 184: Example: eESPA module acts as input program

If this eESPA module acts as an output program:

in the current release, the data identifiers for the espa records are fixed. Therefore, when a message request is sent by the eKERNEL to the eESPA module, the data in the <message_xx> tag is put in data identifier 2 (display message).

In the following authorized, the data in the <message_xx> tag from the message request, must be translated to data identifier 2 (display message) in the espa record.

```
Input: <msgrqs>:
<xml><msgrqs>
<id>00851</id>
<group>12345</group>
<call_type>3</call_type>
<transmission_nmbr>1</transmission_nmbr>
<alarm_cnt>1</alarm_cnt>
<message_01>MESSAGE</message_01>
<beep_code_01>3</beep_code_01>
<priority_01>1</priority_01></msgrqs></xml>
Output: espa record :
SOH1STX1US12345RS2USMESSAGERS3US3RS4US3RS5US1RS6Us1ETXB
CC
```

Figure 185: Example: eESPA module acts as output program

An authorized of an entry typically found in this field is as follows: 2

eESPA_Msg_default_str

This field is used to provide a default message, in the event that no value can be retrieved from the available datastream. In this case, the same value is used for all alarms generated through this eESPA interface instance.

This parameter is only relevant if eESPA acts as an input program (so receives external data from the espa infrastructure.

An authorized of an entry typically found in this field is as follows: ESPA alarm

eESPA_DataId_Ala_descr_str

This field specifies the Data Identifier of the espa record that specifies the alarm description.

This parameter is only relevant if eESPA acts as an input program (so receives external data from the espa infrastructure.

This field refers to the definitions of eKERNEL_ALARM table, and must be appropriately configured (for authorized, message length, and so on).

If the specified data identifier is not present in the available datastream record, than the field eESPA_Ala_descr_default_str must be used to define a default message.

This field can also be a combination of more than one data identifier.

Than the data identifiers must be separated by a ^ sign. If for instance the beep code (data identifier 3) in combination with the priority (data identifier 6) must result in the alarm description, this value must be 3^6.

If the display message (data identifier 2) is a part of the alarm description, you can specify the first x characters of the message as the alarm description. For authorized the value 2:3, results in an alarm description equal to the first 3 characters of the display message (data identifier 2). If the message is, for authorized, NURSE CALL ROOM 02, the alarm description is NUR, so the alarm NUR must be configured in the eKERNEL_ALARM table.

If this eESPA module acts as an input program:

In this authorized, data identifier 3 (beep coding) is defined as eESPA_DataId_Ala_descr_str.

Table 17: Espa record: SOH1STX1US12345RS2USThe messageRS3US9RS4US3RS6US3ETXBCC

(SOH)	Start of header
STX	Start of text
ETX	End of text
US	Unit separator
RS	Record separator
BCC	Checksum

The incoming alarm/message must be translated to a valid message request and sent to the eKERNEL:

```
<msgrqs>:
<xml><msgrqs><set_or_reset>*SET</set_or_reset>
<msg>The message</msg>
<alarmdescr>9</alarmdescr>
<group>12345</group>
<remove_after>*SENT</remove_after>
</msgrqs></xml>
```

Figure 186: Example: eESPA module acts as input program

If this eESPA module acts as an output program:

In the current release, the data identifiers for the espa records are fixed. Therefore, when a message request is sent by the eKERNEL to the eESPA module, the data in the <beep_code_xx> tag is put in data identifier 3 (beep coding).

In the following authorized, the data in the <beep_code_xx> tag from the message request, must be translated to data identifier 3 (beep coding) in the espa record.

```
Input: <msgrqs>:
<xml><msgrqs>
<id>00851</id>
<group>12345</group>
<call_type>3</call_type>
<transmission_nmbr>1</transmission_nmbr>
<alarm_cnt>1</alarm_cnt>
<message_01>MESSAGE</message_01>
<beep_code_01>3</beep_code_01>
<priority_01>1</priority_01></msgrqs></xml>
Output: espa record :
SOH1STX1US12345RS2USMESSAGERS3US3RS4US3RS5US1RS6Us1ETXB
CC
```

Figure 187: Example: eESPA module acts as output program

An authorized of an entry typically found in this field is as follows: 2:3^3. This indicates that the first 3 characters of the display message, a ^ and the values of data identifier 3 is equal to the alarm description. The value NUR^1, NUR^2, SAN^1, and so on, must be configured in the eKERNEL_ALARM table.)

eESPA_Ala_descr_default_str

This field is used to provide a default alarm description, in the event that no value can be retrieved from the available datastream. In this case, the same value is used for all alarms generated through this eESPA interface instance. This alarm description refers to the definitions of eKERNEL_ALARM table.

This parameter is only relevant if eESPA acts as an input program (so receives external data from the espa infrastructure.

An authorized of an entry typically found in this field is as follows: ESPA

eESPA_Remove_after_str

This parameter accepts values *SENT, *RESET, or *CALC.

This parameter is only relevant if eESPA acts as an input program (so it receives external data from the espa infrastructure).

In most cases the eESPA interfaces is used to capture alarms and received data contains alarm information (acts as input program). In this situation messages are transmitted to

eKERNEL immediately upon arrival and these alarms are processed within DECT Messenger.

In some environments, the remote peripherals cannot indicate that pending alarms are reset, and therefore the eKERNEL must handle the alarms. Use this field to configure eKERNEL_ALARM table to correctly handle the alarm requests and refrain from endless-loop conditions. As such alarms are typically *set with the option remove after sent. The eESPA_Remove_after_str are then set to *SENT.

In some environments, the attached peripherals are capable of sending a reset to clear all pending alarms. In such case, alarms must be set using the remove after *RESET value, indicating all pending alarms remain in the eKERNEL database unless the reset condition is met.

This parameter refers to all alarms, so that means that every alarm must receive a reset (a reset occurs if data identifier 4 (call type) is equal to value 1).

If the value *CALC is specified, some alarms receive a reset, and other alarms not. Therefore the eKERNEL application checks to determine if the alarm description with remove after *SENT exists. If so, this alarm type is processed, otherwise the alarm is processed as if remove after *RESET is specified.

If the alarm description is not configured in the eKERNEL_ALARM table, the alarm is not processed.

An authorized of an entry typically found in this field is as follows: *SENT

eESPA_NAK_retry_cnt_n

This field specifies the number of retries to re-transmit a message after receiving a NAK.

A device that has control of the communication line can transfer data to the other devices. When unable to accept the message, the receiving device sends a negative acknowledge with a (1 or 2 or 3) NAK sequence, and the sending device can then retransmit the block. If, after eESPA_NAK_retry_cnt_n attempts, the transmission still fails, and the sending device terminates transmission with the EOT character.

An authorized of an entry typically found in this field is as follows: 2

eESPA_Timeout_n

This values specifies in seconds how long the station waits, if no valid transactions are detect on the communication line, before sending a EOT and terminate the communication and regain control.

An authorized of an entry typically found in this field is as follows: 10

eESPA_Handshaking_n

This field sets and returns the hardware handshaking protocol.

The possible values are:

0 No handshaking. (comNone)

1 XOn/XOff handshaking. (ComXonXoff)

2 Request-to-send/clear-to-send handshaking (comRTS)

3 Both request-to-send and XOn/XOff handshaking. (comRTSXonXoff)

The default value is 0.

An authorized of an entry typically found in this field is as follows: 0

eESPA_OUT_Call_type_default_str

This field is only relevant if the eESPA module acts as output program, so for message sent from the eKERNEL to the eESPA interface.

A <msgrqs> request from the eKERNEL to the espa interface, contains a tag <call_type> that defines the value for data identifier 4 (call type). If *NONE is specified, data identifier 4 is not a part of the espa record.

The possible values are: 0, 1, 2, 3, *NONE

In the following authorized, the data in the <call_type> tag from the message request, must be translated to data identifier 4 (call type) in the espa record.

```
Input: <msgrqs>:
<xml><msgrqs>
<id>00851</id>
<group>12345</group>
<call_type>3</call_type>
<transmission_nmbr>1</transmission_nmbr>
<alarm_cnt>1</alarm_cnt>
<message_01>MESSAGE</message_01>
<beep_code_01>3</beep_code_01>
<priority_01>1</priority_01></msgrqs></xml>
Output: espa record :
SOH1STX1US12345RS2USMESSAGERS3US3RS4US3RS5US1RS6Us1ETXB
CC
```

Figure 188: Example: eESPA module acts as output program

An authorized of an entry typically found in this field is as follows: 3

eESPA_OUT_Nmbr_transm_default_str

This field is only relevant if the eEPSA module acts as an output program, so for message sent from the eKERNEL to the eESPA interface.

A <msgrqs> request from the eKERNEL to the espa interface, contains a tag <transmission_nmbr> that defines the value for data identifier 5 (transmission number). If *NONE is specified, data identifier 5 is not a part of the espa record.

In the following authorized, the data in the <transmission_nmbr> tag from the message request, must be translated to data identifier 5 (number of transmissions) in the espa record.

```
Input: <msgrqs>:
<xml><msgrqs>
<id>00851</id>
<group>12345</group>
<call_type>3</call_type>
<transmission_nmbr>1</transmission_nmbr>
<alarm_cnt>1</alarm_cnt>
<message_01>MESSAGE</message_01>
<beep_code_01>3</beep_code_01>
<priority_01>1</priority_01></msgrqs></xml>
Output: espa record :
SOH1STX1US12345RS2USMESSAGERS3US3RS4US3RS5US1RS6Us1ETXB
CC
```

Figure 189: Example: eESPA module acts as output program

An authorized of an entry typically found in this field is as follows: 1

eESPA_Comments_str

This field can be filled with comments, to allow administrators to add some remarks to the configuration record.

Table: eESPA

Chapter 19: Table: eESPA_OUTBOUND_CFG

eESPA_outbond_cfg parameters

Name	Type	Size
eESPAO_Site_id_n	Integer	2
eESPAO_Area_id_n	Integer	2
eESPAO_ALA_Prty_from_n	Integer	2
eESPAO_ALA_Prty_to_n	Integer	2
eESPAO_BeepCode_str	Text	5
eESPAO_Priority_str	Text	5
eESPAO_Comments_str	Text	255

eESPAO_Site_id

This field specifies the site identifier, as defined in eKERNEL_SITE. This value is, in most environments, equal to 1.

An authorized of an entry typically found in this field is as follows: 1

eESPAO_Area_id_n

This field specifies the site identifier, as defined in eKERNEL_AREA. This value is, in most environments, equal to 1.

An authorized of an entry typically found in this field is as follows: 1

eESPAO_ALA_Prty_from_n

This field refers to the ALA_Prty_n field of the table eKERNEL_ALARM, and defines the priority of an alarm.

A low value indicates an important alarm, a high value a less important alarm.

With the fields eESPAO_ALA_Prty_from_n and eESPAO_ALA_Prty_to_n you can specify a range of alarm priorities and set a relationship to the beepcode record type and the priority record type of the espa datablock.

The Data identifier for the beepcode record type is 3.

The Data identifier for the priority record type is 6.

Table 18: Example eESPAO_ALA_Prty_from/to_n values

Site	Area	Alarm from	Alarm to	Beepcode	Priority
1	1	0	2	1	2
1	1	3	5	3	1
1	1	6	999	9	3

When a <msgrqs> is sent to the eESPA with an alarm priority equal to 2 for pager 4567, a datablock is created with a beepcode 1 (data identifier 3) and a priority 2 (High) (data identifier 6). Therefore, all alarms with a priority between 0 and 2 have these specifications.

Example datablock:

(RS: record separator, US: Unit separator

* Alarm priority equal to or between 0 and 2

```
1US4567RS2USExampleRS3US1RS6US2
```

* Alarm priority equal to or between 3 and 5

```
1US4567RS2USExampleRS3US3RS6US1
```

* Alarm priority equal to or between 6 and 999 (highest possible value)

```
1US4567RS2USExampleRS3US9RS6US3
```

An authorized of an entry typically found in this field is as follows: 0

eESPAO_ALA_Prty_to_n

See eESPAO_ALA_Prty_from_n on page 241.

An authorized of an entry typically found in this field is as follows: 999

eESPAO_BeepCode_str

This field specifies the data that must be entered in the espa datablock for record type beepcode (data identifier 3).

An authorized of an entry typically found in this field is as follows: 1

eESPAO_Priority_str

This field specifies the data that must be entered in de espa datablock for record type priority (data identifier 6).

An authorized of an entry typically found in this field is as follows: 3

eESPAO_Comments_str

This field can be used to store comments, enabling administrators to add remarks to the configuration record. See Table 19: Sample eESPAO_Comments_str values on page 243 for authorized eESPAO_Comments_str values.

Table 19: Sample eESPAO_Comments_str values

Site	Area	Alarm from	Alarm to	Beepcode	Priority
1	1	0	2	1	2
1	1	3	5	3	1
1	1	6	999	9	3
1	2	0	5	1	3
1	2	6	999	*NONE	3
1	3	0	999	1	*NONE

Table: eESPA_OUTBOUND_CFG

Chapter 20: Table: eIO_MODULE

eIO_modules parameters

Name	Type	Size
eIOM_Site_id_n	Integer	2
eIOM_Area_id_n	Integer	2
eIOM_Module_str	Text	4
eIOM_Type_str	Text	50
eIOM_Url_str	Text	255
eIOM_Contact_cnt_n	Integer	2
eIOM_Comments_str	Text	255

eIOM_Site_id_n

This field specifies the site identifier, as defined in eKERNEL_SITE table. In most environments, this value is 1.

An authorized of an entry typically found in this field is as follows: 1

eIOM_Area_id_n

This field specifies the area identifier, as defined in eKERNEL_AREA table. In most environments, this value is 1.

An authorized of an entry typically found in this field is as follows: 1

eIOM_Module_str

This value refers to the 2-byte module identifier, specified in the FieldPoint Explorer and in eIO_MODULE table. A typical value is between 01 and 08. Ensure that the table eIO_DO defines only digital output-contacts (those with a matching digital output module). The current release supports up to eight modules per eIO instance, including one FP-1000 controlling module, and can refer to FP-AI-100, DP-DI-300, FP-DI-301, FP-DI-330 and FP-DO-401.

The current implementation of eIO is limited to configurations of up to eight modules attached to one FP-1000 controller module. Avaya recommends starting the first module with number 01 and incrementing by 1 for the other modules.

Note:

Specify the leading 0 in the numbering (enter the value 01, not 1).

An authorized of an entry typically found in this field is as follows: 01

eIOM_Type_str

The current release supports the following modules:

Table 20: eIOM supported modules

FP-AI-100	Analogue input	8 contacts
FP-DI-300	Digital input	8 contacts
FP-DI-301	Digital input	16 contacts
FP-DI-330	Digital input	8 contacts
FP-DI-401	Digital output	8 contacts

Refer to the corresponding chapter in this document for technical specifications on the modules.

An authorized of an entry typically found in this field is as follows: FP-DI-330

eIOM_Url_str

This field denotes the URL string associated with the module. Refer to the FieldPoint Explorer and other National Instrument distributed I/O documentation resources for more information on the URL defined OPC server binding mechanism.

The FieldPoint Explorer is a recommended way to determine naming conventions. Take note of the ending characters specified in Table 21: eIO_module sample data on page 247. Using an incorrect URL prevents binding contacts to the OPC Server, resulting in system malfunction.

An authorized of an entry typically found in this field is as follows: opc:/National Instruments.OPCFieldPoint/FP Res\FP-AI-100 @1\Channel

eIOM_Contact_cnt_n

The field defines the number of contacts that are associated to the module. This field can specify a smaller number than the maximum number of physical available contacts on a

module, in which case the remaining contacts are not bound to the OPC Server and remain non-operational.

An authorized of an entry typically found in this field is as follows: 8

eIOM_Comments_str

This field can be entered with remarks from an administrator, and is informational only. You can use this filed to document the physical connection here too, to ease later configuration.

An authorized of an entry typically found in this field is as follows: OR 004 – fire detection.

Table 21: eIO_module sample data on page 247 provides sample eIO module table data.

Table 21: eIO_module sample data

Site	Area	Module	Type	URL	Count
1	1	01	FP-DI-300	opc:/National Instruments.OPCFieldPoint/FP Res\FP-DI-330 @2\Channel	16
2	1	01	FP-AI-100	opc:/National Instruments.OPCFieldPoint/FP Res\FP-AI-100 @1\Channel	8
2	1	02	FP-DI-300	opc:/National Instruments.OPCFieldPoint/FP Res\FP-DI-330 @2\Channel	8
2	1	03	FP-DO-401	opc:/National Instruments.OPCFieldPoint/FP Res\FP-DO-401 @3\Channel	16
2	2	01	FP-AI-100	opc:/National Instruments.OPCFieldPoint/FP Res\FP-AI-100 @1\Channel	8
2	2	02	FP-DI-300	opc:/National Instruments.OPCFieldPoint/FP Res\FP-DI-330 @2\Channel	8
2	2	03	FP-DO-401	opc:/National Instruments.OPCFieldPoint/FP Res\FP-DO-401 @3\Channel	16
2	3	02	FP-DI-300	opc:/National Instruments.OPCFieldPoint/FP Res\FP-DI-330 @2\Channel	8

Table: eIO_MODULE

Chapter 21: Table: eIO_AI

eIO_AI parameters

Name	Type	Size
eIOAI_Site_id_n	Integer	2
eIOAI_Area_id_n	Integer	2
eIOAI_Module_str	Text	4
eIOAI_Contact_str	Text	2
eIOAI_Min_S_str	Text	10
eIOAI_Min_R_str	Text	10
eIOAI_Max_R_str	Text	10
eIOAI_Max_S_str	Text	10
eIOAI_ALA_descr_str	Text	50
eIOAI_GRP_Name_str	Text	20
eIOAI_MSG_str	Text	255
eIOAI_Comments_str	Text	255

eIOAI_Site_id_n

This field refers to the site identifier, as defined in the eKERNEL_SITE table. In most environments, this value is 1.

An authorized of an entry typically found in this field is as follows: 1

eIOAI_Area_id_n

This field refers to the area identifier, as defined in the eKERNEL_AREA table. In most environments, this value is 1.

An authorized of an entry typically found in this field is as follows: 1

eIOAI_Module_str

This value refers to the 2-byte module identifier, specified in the FieldPoint Explorer and in eIO_MODULE table. A typical value is between 01 and 08. Verify that the table eIO_AI only defines analogue input-contacts (the contacts with a matching analogue input module). Current release supports FP-AI-100 modules.

Current implementation of eIO is limited to configurations of up to 8 modules attached to one FP-1000 controller module. Avaya recommends starting the first module with number 01 and incrementing by one for the other modules. Specify the leading 0 in the numbering (do not specify 1, but specify instead 01).

An authorized of an entry typically found in this field is as follows: 01

eIOAI_Contact_str

This value refers to each individual contact, and is specified in the FieldPoint Explorer. Valid values are in the range between 01 and 08 for the currently supported FP-AI-100. Note that contact numbers start with 01 and are incremented by one. You must specify the leading 0 in the numbering (do not specify 1, but specify instead 01). Note that some peripherals of National Instruments include labels and documentations where contacts start numbering at 0 up to 7, whereas eIO starts at 01 up to 08.

An authorized of an entry typically found in this field is as follows: 01

eIOAI_Min_S_str

The value specifies the analogue level measured on a contact to set a minus-level alarm. If minus-level alarms are to be disabled a 00,000000 value can be specified.

Note:

All values must be specified in format 00,000000 with 2 digits before the decimal separator and 6 digits after the decimal separator. The decimal separator must be set according to the operating system regional settings.

Refer to the FieldPoint Explorer documentation on how to configure the FP-AI-100 module. Each channel can individually be set according to the attached input, and allow specifying the unit of measurement en the measured input range. Avaya recommends that you first test the peripherals with the FieldPoint Explorer prior to configuring and taking eIO into production.

Note:

Check your operating system settings to find out which decimal separator is in use. Avaya recommends that you set the operating system to the country specific values, thus the locale Belgium (Dutch) in Belgium, even when an English operating system is installed. These regional settings result in internal usage of decimal separator symbols in the form of period (.) or comma (,). If your system is set up with comma (,) as decimal separator, a comma (,) must also be specified when values are entered in the database.

An authorized of an entry typically found in this field is as follows: 03,000000

eIOAI_Min_R_str

The value specifies the analogue level measured on a contact to reset a minus-level alarm. If minus-level alarms are to be disabled a 00,000000 value can be specified.

Note:

All values must be specified in format 00,000000 with 2 digits before the decimal separator and 6 digits after the decimal separator. The decimal separator must be set according to the operating system regional settings.

Refer to the FieldPoint Explorer documentation for more information the configuration of the FP-AI-100 module. Each channel can individually be set according to the attached input, and allow to specify the unit of measurement en the measured input range. Avaya recommends that you first test the peripherals with the FieldPoint Explorer prior to configuring and taking eIO into production.

Note:

Check your operating system settings to find out which decimal separator is in use. Avaya recommends that you set the operating system to the country-specific values, thus the locale Belgium (Dutch) in Belgium, even when an English operating system is installed. These regional settings result in internal usage of decimal separator symbols in the form of period (.) or comma (,). If your system is set up with comma (,) as decimal separator, a comma (,) must also be specified when values are entered in the database.

An authorized of an entry typically found in this field is as follows: 06,000000

eIOAI_Max_R_str

The value specifies the analogue level measured on a contact to set a plus-level alarm.

If plus-level alarms are to be disabled a 99,999999 value can be specified.

Note:

All values must be specified in format 00,000000 with 2 digits before the decimal separator and 6 digits after the decimal separator. The decimal separator must be set according to the operating system regional settings.

Refer to the FieldPoint Explorer documentation for more information on the configuration of the FP-AI-100 module. Each channel can individually be set according to the attached input, and allow specifying the unit of measurement en the measured input range. Avaya recommends that you first test the peripherals with the FieldPoint Explorer prior to configuring and taking eIO into production.

Note:

Check your operating system settings to find out which decimal separator is in use. Avaya recommends that you set the operating system to the country specific values, thus the locale Belgium (Dutch) in Belgium, even when an English operating system is installed. These regional settings result in internal usage of decimal separator symbols in the form of period (.) or comma (,). If your system is set up with comma (,) as decimal separator, a comma (,) must also be specified when values are entered in the database.

An authorized of an entry typically found in this field is as follows: 20,000000

eIOAI_Max_S_str

The value specifies the analogue level measured on a contact to reset a plus-level alarm.

If plus-level alarms are to be disabled a 99,999999 value can be specified.

Note:

All values must be specified in format 00,000000 with 2 digits before the decimal separator and 6 digits after the decimal separator. The decimal separator must be set according to the operating system regional settings.

Refer to the FieldPoint Explorer documentation for more information on the configuration of the FP-AI-100 module. Each channel can individually be set according to the attached input, and allow specifying the unit of measurement en the measured input range. Avaya recommends that you first test the peripherals with the FieldPoint Explorer prior to configuring and taking eIO into production.

Note:

Check your operating system settings to find out which decimal separator is in use. Avaya recommends that you set the operating system to the country specific values, thus the locale Belgium (Dutch) in Belgium, even when an English operating system is installed. These regional settings result in internal usage of decimal separator symbols in the form of period (.) or comma (,). If your system is set up with comma (,) as decimal separator, a comma (,) must also be specified when values are entered in the database.

When values are entered in the database.

An authorized of an entry typically found in this field is as follows:15,000000.

eIOAI_ALA_Descr_str

The alarm description field is a description defined in the eKERNEL_ALARM table for the associated eIO module. In the authorized shown in Table 22: eIOAS_ALA_Descr_str

authorized on page 253, an alarm description A-INPUT is defined with matching records in the eKERNEL_ALARM table.

Table 22: eIOAS_ALA_Descr_str authorized

ALA_id_n	ALA_INPGM_id	ALA_Descr_str	ALA_Remove_	ALA_Prty_n
1160101	11601	A-INPUT	*SENT	5
1160102	11601	A-INPUT	*RESET	5

An authorized of an entry typically found in this field is as follows: A-INPUT

eIOAI_GRP_Name_str

The group name describes what group is informed on the error condition, and refers to a group defined in eKERNEL_GROUP and eKERNEL_GROUP_MEMBER tables.

An authorized of an entry typically found in this field is as follows: 00003

eIOAI_MSG_str

This field describes the message that is sent to the group members. Avaya recommends that you enter descriptive text that provides the recipient sufficient information to handle the alarm condition. Mobile users often lack immediate access to other information resources such as plans, technical specs, and so on. Avaya recommends that you select an appropriate message that is short and descriptive enough, and keep text length less than, or equal to, the maximum length defined in the associated eKERNEL_ALARM table.

An authorized of an entry typically found in this field is as follows: TEMPERATURE OR 002

eIOAI_Comments_str

This field is available for an administrator to enter some descriptive text that allows location and identification of the attached input device and its usage.

Table 23: eIO_AI sample data on page 254 provides sample eIO_AI module table data.

Table 23: eIO_AI sample data

SITe	AREa	Mod	Cont	Min_S	Min_R	Max_R	Max_S	ALA_descr	Group	MSG
1	1	01	01	00,000000	00,000000	00,000400	00,000400	A-INPUT	AI	Analog Input 01
1	1	01	02	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 02
1	1	01	03	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 03
1	1	01	04	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 04
1	1	01	05	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 05
1	1	01	06	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 06
1	1	01	07	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 07
1	1	01	08	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 08
2	1	01	01	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 01
2	1	01	02	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 02
2	1	01	03	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 03
2	1	01	04	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 04

SITe	AREa	Mod	Cont	Min_S	Min_R	Max_R	Max_S	ALA_descr	Group	MSG
2	1	01	05	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 05
2	1	01	06	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 06
2	1	01	07	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 07
2	1	01	08	00,000000	00,000000	12,000000	20,000000	A-INPUT	AI	Analog Input 08
2	2	01	01	00,000000	00,000000	12,000000	20,000000	A-INPUT	00001	Analog Input 01
2	2	01	02	00,000000	00,000000	12,000000	20,000000	A-INPUT	00001	Analog Input 02
2	2	01	03	00,000000	00,000000	12,000000	20,000000	A-INPUT	00001	Analog Input 03
2	2	01	04	00,000000	00,000000	12,000000	20,000000	A-INPUT	00001	Analog Input 04
2	2	01	05	00,000000	00,000000	12,000000	20,000000	A-INPUT	00001	Analog Input 05
2	2	01	06	00,000000	00,000000	12,000000	20,000000	A-INPUT	00001	Analog Input 06
2	2	01	07	00,000000	00,000000	12,000000	20,000000	A-INPUT	00001	Analog Input 07
2	2	01	08	00,000000	00,000000	12,000000	20,000000	A-INPUT	00001	Analog Input 08

Table: eIO_AI

Chapter 22: Table: eIO_DI

eIO_DI parameters

Name	Type	Size
eIODI_Site_id_n	Integer	2
eIODI_Area_id_n	Integer	2
eIODI_Module_str	Text	4
eIODI_Contact_str	Text	2
eIODI_ContactType_str	Text	2
eIODI_ALA_Descr_str	Text	50
eIODI_GRP_Name_str	Text	20
eIODI_MSG_str	Text	255
eIODI_Comments_str	Text	255

eIODI_Site_id_n

This field refers to the site identifier, as defined in the eKERNEL_SITE table. In most environments, this value is 1.

An authorized of an entry typically found in this field is as follows: 1

eIODI_Area_id_n

This field refers to the area identifier, as defined in the eKERNEL_AREA table. In most environments, this value is 1.

An authorized of an entry typically found in this field is as follows: 1

eIODI_Module_str

This value refers to the 2-byte module identifier, specified in the FieldPoint Explorer and in eIO_MODULE table. A typical value is between 01 and 08. Ensure that the table eIO_DI only defines digital input-contacts, thus only the contacts with a matching digital input module. Current release supports FP-DI-300, FP-DI-301 and FP-DI-330.

Current implementation of eIO is limited to configurations of up to eight modules attached to one FP-1000 controller module. Avaya recommends starting the first module with number 01

and incrementing by one for the other modules. Specify the leading 0 in the numbering (do not specify 1, but specify instead 01).

An authorized of an entry typically found in this field is as follows: 02

eIODI_Contact_str

Valid values are in the range between 01 and 08 for the modules with 8 contacts and between 01 and 16 for the modules with 16 contacts. Note contact numbers start with 01 and are incremented by one. You must specify the leading 0 in the numbering (do not specify 1, but specify instead 01). Note that some peripherals of National Instruments include labels and documentations where contacts start numbering at 0 up to 7 (or 0 up to 15), whereas eIO starts at 01 up to 08 (or 01 up tot 16).

This value refers to each individual contact, and is specified in the FieldPoint Explorer. Range of values are 01 to 16 for FP-DI-301 module and 01 to 08 for the other digital input modules.

An authorized of an entry typically found in this field is as follows: 01

eIODI_ContactType_str

This parameter accepts the following values:

OS (in Dutch open schakelaar – open switch) meaning the contact is, in the base state, open and can be switched on at set and remains on until switched off at reset

OD (in Dutch open drukknop – open push button) meaning the contact is in base state open and can be switched on for a very short time and immediately fall back to the base state. Typically used for push buttons that generate alarm.

GS (in Dutch gesloten schakelaar – closed switch) meaning the contact is in base state closed and can be switched off at set and remains off until switched back on at reset.

GD (in Dutch gesloten drukknop – closed push button) meaning the contact is in base state closed and can be switched off for a very short time and immediately fall back to the base state.

An authorized of an entry typically found in this field is as follows: GD

eIODI_ALA_Descr_str

The alarm description field is a description defined in the eKERNEL_ALARM table for the associated eIO module. In the authorized shown in Table 24: eIO alarm description on page 259, an alarm description D-INPUT is defined with matching records in the ALARM table, as shown in Table 24: eIO alarm description on page 259.

Table 24: eIO alarm description

Alarm ID	Input program	Alarm description	Remove after	Priority
1160101	11601	D-INPUT	*SENT	5
1160101	11601	D-INPUT	*RESET	5

An authorized of an entry typically found in this field is as follows: D-INPUT

eIODI_GRP_Name_str

The group name describes what group is informed on the error condition, and refers to a group defined in eKERNEL_GROUP and eKERNEL_GROUP_MEMBER table.

An authorized of an entry typically found in this field is as follows: 00003

eIODI_MSG_str

This field describes the message that is sent to the group members. Avaya recommends that you enter descriptive text that provides the recipient sufficient information to handle the alarm condition. When selecting a message, Avaya recommends that you take into account that mobile users often lack immediate access to other information resources, such as a site map or technical specs, and keep the message length less than, or equal to, the maximum length defined in the associated eKERNEL_ALARM table.

An authorized of an entry typically found in this field is as follows: FIRE IN ELEVATOR

eIODI_Comments_str

This field is available for an administrator to enter some descriptive text that allows location and identification of the attached input device and its usage.

Table 25: eIO_DI sample data on page 259 provides sample eIO_DI module table data.

Table 25: eIO_DI sample data

Site	Area	Mod	Contact	Type	ALA_Descr	GRP_Name	Message
1	1	01	01	OD	D-INPUT	DI	Digital Input 01
1	1	01	02	OS	D-INPUT	DI	Digital Input 02

Table: eIO_DI

Site	Area	Mod	Contact	Type	ALA_Descr	GRP_Name	Message
1	1	01	03	GS	D-INPUT	DI	Digital Input 03
1	1	01	04	GD	D-INPUT	DI	Digital Input 04
1	1	01	05	OD	D-INPUT	DI	Digital Input 05
1	1	01	06	OD	D-INPUT	DI	Digital Input 06
1	1	01	07	OD	D-INPUT	DI	Digital Input 07
1	1	01	08	OS	D-INPUT	DI	Digital Input 08
1	1	01	09	OD	D-INPUT	DI	Digital Input 09
1	1	01	10	OS	D-INPUT	DI	Digital Input 10
1	1	01	11	GS	D-INPUT	DI	Digital Input 11
1	1	01	12	GD	D-INPUT	DI	Digital Input 12
1	1	01	13	OD	D-INPUT	DI	Digital Input 13
1	1	01	14	OD	D-INPUT	DI	Digital Input 14
1	1	01	15	OD	D-INPUT	DI	Digital Input 15
1	1	01	16	OS	D-INPUT	DI	Digital Input 16

Chapter 23: Table: eIO_DO

eIO_DO parameters

Name	Type	Size
eIODO_Site_id_n	Integer	2
eIODO_Area_id_n	Integer	2
eIODO_Module_str	Text	2
eIODO_Contact_str	Text	2
eIODO_Seconds_n	Integer	2
eIODO_Comments_str	Text	255

eIODO_Site_id_n

This field specifies the site identifier, as defined in the eKERNEL_SITE table. In most environments, this field has value 1.

An authorized of an entry typically found in this field is as follows: 1

eIODO_Area_id_n

This field specifies the area identifier, as defined in the eKERNEL_AREA table. In most environments, this field has value 1.

An authorized of an entry typically found in this field is as follows: 1

eIODO_Module_str

This value refers to the two-byte module identifier, specified in the FieldPoint Explorer and in eIO_MODULE table. A typical value is between 01 and 08. Ensure that the table eIO_DO only defines digital output-contacts, thus only the contacts with a matching digital output module. Current release supports FP-DO-401 modules.

Current implementation of eIO is limited to configurations of up to 8 modules attached to one FP-1000 controller module. Avaya recommends starting the first module with number 01 and incrementing by one for the other modules. Specify the leading 0 in the numbering (do not specify 1, but specify instead 01).

An authorized of an entry typically found in this field is as follows: 01

eIODO_Contact_str

This value refers to each individual contact, and is specified in the FieldPoint Explorer. Valid values are in the range between 01 and 16 for the currently supported FP-DO-401. Note contact numbers start with 01 and are incremented by one. You must specify the leading 0 in the numbering (do not specify 1, but specify instead 01). Note that some peripherals of National Instruments include labels and documentations where contacts start numbering at 0 up to 15, whereas eIO starts at 01 up to 16.

An authorized of an entry typically found in this field is as follows: 01

eIODO_Seconds_n

When the eKERNEL sends a <msgrqs> to change the state of the contact, the eIO performs the requested operation.

The state of the discrete output changes from 0 to 1.

The value eIODO_Seconds_n specifies the number of seconds a digital output remains activated. For instance, if the value 5 is specified, the signal remains 1 for 5 seconds, then the signal drops again to 0.

The special value triggers the contact for a very small amount of time. The value immediately returns to 0. In many environments the signal is too short to steer an external peripheral.

A typical value is 5 so that the discrete contact is activated for 5 seconds and then returns to an idle state.

An authorized of an entry typically found in this field is as follows: 5

eIODO_Comments_str

This field can contain remarks from the administrator. The value is informational only, and does not affect processing.

Table 26: eIO_DO sample data on page 262 provides sample eIO_DO module table data.

Table 26: eIO_DO sample data

Site	Area	Module	Contact	Seconds	Comments
1	1	03	01	5	
1	1	03	02	5	

Site	Area	Module	Contact	Seconds	Comments
1	1	03	03	5	
1	1	03	04	5	
1	1	03	05	5	
1	1	03	06	5	
1	1	03	07	5	
1	1	03	08	5	
2	1	03	01	5	
2	1	03	02	5	
2	1	03	03	5	
2	1	03	04	5	
2	1	03	05	5	
2	1	03	06	5	
2	1	03	07	5	
2	1	03	08	5	
2	2	03	01	5	
2	2	03	02	5	
2	2	03	03	5	
2	2	03	04	5	
2	2	03	05	5	
2	2	03	06	5	
2	2	03	07	5	
2	2	03	08	5	

Table: eIO_DO

Chapter 24: Table: eKERNEL_AREA

eKERNEL_area parameters

Name	Type	Size
AREA_Site_id_n	Integer	2
AREA_Area_id_n	Integer	2
AREA_Area_Descr_str	Text	50
AREA_Area_Comments_str	Text	255

AREA_Site_id_n

This field refers to the site identifier, as defined in the eKERNEL_SITE table. In most cases only one site is configured. A typical value is 1.

An authorized of an entry typically found in this field is as follows: 1

AREA_Area_id_n

This field indicates the area identifier. The combination site and area must be unique in the database.

In most cases the configuration consists of 1 site and 1 area. As explained in the eKERNEL_SITE table, the term site is referred to an environment that is handled by one single eKERNEL instance.

The concept of area is introduced in DECT Messengerin release 2. Prior to this release, there were a number of constraints, for authorized there could only be one instance be defined for several modules. This limitation affected both input programs and output programs.

With the introduction of the area concept, a site can now cover several divisions. These divisions can be geographically distributed to multiple locations, or they can all be in the same location.

One advantage of the area concept is that some configuration limitations are no longer active. For instance, you can now define multiple instances of both input programs and output programs. For authorized, an immediate result is the ability to support two or more eIO modules, with the immediate advantage that analogue input and discrete input modules can now be installed in a distributed location (near the contacts).

The most significant focus is however on output program level. With the area concept, you can now configure, for authorized, more than one instance of eDMSAPI. This is most useful in larger environments (for authorized, 3 high-range iS-3090 switches covering 3 locations in an IMP network), where you can now install one eDMSAPI per area (location). Because communication to the central eKERNEL (one per site) is now on sockets basis on the WAN, this dramatically reduces IMP network traffic, because calls can be processed locally on each location.

As a result of this design, the area field is found in many other tables. Peripherals (better known as devices) are now identified by site, area, output program and device.

An authorized of an entry typically found in this field is as follows: 1

AREA_Area_Descr_str

This field allows you to enter a small description of the area. This description is for instance visualized on several windows on the eWEB interface.

An authorized of an entry typically found in this field is as follows: Campus Sint-Jan

AREA_Area_Comments_str

This field can be used to add some additional comments and is informational only.

An authorized of an entry typically found in this field is as follows: Main area with iS-3090 switch

Chapter 25: Table: eKERNEL_ALARM

eKERNEL_alarm parameters

Name	Type	Size
ALA_id_n	Long Integer	4
ALA_INPGM_id_n	Long Integer	4
ALA_Descr_str	Text	50
ALA_Remove_after_str	Text	6
ALA_Prty_n	Integer	2
ALA_to_ringing_n	Integer	2
ALA_to_Connect_n	Integer	2
ALA_to_Queued_n	Integer	2
ALA_Silence_intv_n	Integer	2
ALA_Scroll_state_str	Text	15
ALA_Scroll_intv_n	Integer	2
ALA_Group_delivery_str	Text	5
ALA_Confirm_action_str	Text	4
ALA_Repeat_intv_n	Integer	2
ALA_Length_n	Integer	2
ALA_Trace_b	Yes/No	1
ALA_Trace_dayToKeep_n	Integer	2
ALA_Comments_str	Text	255

ALA_id_n

This field specifies the unique identifier of the alarm. Although you can to enter a numeric value of choice, Avaya recommends developing a logical naming convention for alarms.

A common approach is to base the numbering scheme upon input program identifier (that in turn is built upon site and area of the input program and a input program sequence number). A two-byte sequence number is the appended. This brings the length to seven bytes.

Table 27: Alarm identifiers

Byte 1	Site identifier				
Byte 2	Area identifier				
Byte 3-5	Input program identifier				
	Byte 3	1	eCAP or eAPI or eESPA		
		2	eSNMP		
		4	eVBVOICE		
		5	eCSTA		

		6	eIO		
		7	eWEB		
		8	eSMTP_server		
		9	eDMSAPI		
	Byte 4-5	01-99	Input program sequence number		
Byte 6-7	Alarm sequence number				

As shown in Table 27: Alarm identifiers on page 267, the first bytes denote the site identifier. The second byte denotes the area identifier. The third byte denotes the input application type. The fourth and fifth byte indicates a sequence number. These five first bytes refer to the input-program identifier.

The two remaining bytes (byte 6 and 7) are a sequence number that specified the alarm for that input program.

The first five digits match the value of the field ALA_INPGM_id_n. This helps to keep track of alarms in the complex definitions that occur in some configurations.

An authorized of an entry typically found in this field is as follows: 1110101 (denotes site 1, area 1, eCAP 01, alarm 01)

ALA_INPGM_id_n

This field specifies the unique identifier of the input program.

Note that this identifier is defined in the eKERNEL_TCPCLIENT table (field TCPCLIENT_INPGM_id_n). Refer to the section of eKERNEL_TCPCLIENT on how to set up these input programs.

Avaya recommends that you develop a naming convention to assign values for these identifiers.

Table 28: Alarm input program identifiers

Byte 1	Site identifier				
Byte 2	Area identifier				
Byte 3-5	Input program identifier				
	Byte 3	1	eCAP or eAPI or eESPA		
		2	eSNMP		
		4	eVBVOICE		
		5	eCSTA		
		6	eIO		

		7	eWEB		
		8	eSMTP_server		
		9	eDMSAPI		
	Byte 4-5	01-99	Input program sequence number		

Avaya recommends using five digits to uniquely identify an input program. With the guidelines above, the identifier implies the site, area, input program application and sequence number.

The ALA_id_n and ALA_INPGM_id_n both form a unique key, thus one input program with ALA_INPGM_id_n value 11101 cannot have two records with the same ALA_id_n value 1110101.

An authorized of an entry typically found in this field is as follows: 11101

ALA_Descr_str

This field is a very important parameter in the DECT Messengeralarm handling.

Important:

Do not confuse this value with the ALA_Comments_str field for giving a description to the alarm.

The ALA_Descr_str contains a string of one or more characters. The eCAP alarm capture programs use these characters to find an appropriate alarm definition for a received alarm string.

The proper usage of this field is highly depending on the proprietary protocol implementation in eCAP and other input programs, such as eWEB. In many cases, some rules are defined for handling alarms from external systems.

The alarm generates some kind of string with information, and DECT Messengermust find out how to handle the string. The retrieval of the alarm definition from the eKERNEL_ALARM table is performed using the ALA_Descr_str field.

A special value *OTHER can be defined. If specified, the *OTHER description is used to handle alarms that were not identified by a qualified description.

Alarms with descriptions that do not either match a qualified description or the value *OTHER, are ignored.

Refer to other reference material for detailed instructions for each alarm system. The following authorizeds are provided to clarify the usage:

Example 1: ELDAD

If the alarm is described as ELDAD, alarms are sent where behavior depends on a tone code. Alarms with tone code 1, 2, 3 and 4 each have different characteristics, and need different alarm handling. In the case of ELDAD define the ALA_Descr_str values 1, 2, 3 and 4 for the 4 corresponding records.

Example 2: TELEVIC

TELEVIC sends alarms where behavior depends on tone code or message contents.

If the alarm is described as TELEVIC, the system looks first for a string pattern (first blank or first xx characters as specified in the L:xx description of the INPGM_Model_str field of the eKERNEL_INPGM table (PROTOCOL CONVERTOR – L:03). If no length (L:xx) is specified, the default value is 3. Characters of message or search until first blank character: NUR, SAN, ASS, REA, MUG, and so on.

See documentation Table eKERNEL_inpgm.pdf.

If no such definition is found; the system looks for a matching tone code pattern (for authorized, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 or 0).

If again no definition is found, the system looks for an *OTHER definition.

Example 3 - National Instruments

The National Instruments distributed I/O modules FP-DI-300, FP-DI-301 and FP-DI-330 generate discrete input alarms, the I/O module FP-AI-100 generates analogue input alarms. Both modules are configured in eIO_MODULE, eIO_AI and eIO_DI tables. In the latter two files the alarm type can be defined, default is D-INPUT and A-INPUT. If these defaults are used, ALA_Descr_id_str must have records for D-INPUT and A-INPUT.

Example 4 - Guarding

A special feature in the eCAP input program consists of a method to verify the amount of time between two requests. If a specific type has elapsed, this can be caused by a failure in the external alarm system or the physical interface. In such case, GUARDING can be implemented. This is configured in the eKERNEL_GUARDING table. The link between eKERNEL_GUARDING and eKERNEL_ALARM is performed through an alarm identifier, but Avaya recommends specifying GUARDING in the alarm description field.

An authorized of an entry typically found in this field is as follows: GUARDING

ALA_Remove_after_str

This field can have the value *SENT or *RESET.

If the field value is *SENT, the message is removed after successfully sending the message.

If the field value is *RESET, the message remains in the database until an explicit reset signal is received from the alarm system.

Again, this value is generally depending of the proprietary implementation of the alarm system and the attached peripherals. Some devices can send a SET and RESET indication (for authorized, a switch button van be set to on or off); others cannot generate a RESET (for authorized, a push button can only generate a push signal while pressing the contact).

In some cases you can have difficulty determining whether alarms have reset or not. In fact, some third-party alarm system vendors are not aware of the signals provided. In these cases,

you must specify *SENT, to prevent alarms that do not receive a *RESET from remaining active in the system.

An authorized of an entry typically found in this field is as follows: *SENT

ALA_Prty_n

This field specifies the priority of an alarm. A low value indicates an important alarm, a high value a less important alarm. Avaya recommends that you exercise caution when assigning priorities to alarms. For some output devices, high-important alarms are shown first and low-priority alarms are shown last.

Other output programs (such as eSMTP and eASYNC) allow you to automatically confirm arrival of messages when distributed, while others require confirmation procedures based upon a call back procedure (using CLID on eCSTA of DTMF pincode on eVBVOICE).

Avaya recommends that you begin by assigning all alarms to default priority 5 (for authorized, nurse calls, and so on) and assigning more important alarms to a lower value (1 for MUG, 2 for REA, 3 for ASS, and so on) and less important alarms to a higher value (6 for SAN, and so on). In most cases, alarm priorities are subject to discussion with those in authority on-site.

An authorized of an entry typically found in this field is as follows: 5

ALA_to_ringing_n

This field specifies the number of seconds a peripheral is kept in ringing state before taking further action. This parameter is ignored for most peripherals.

Currently this value impacts only the “eCSTA” module for voice-call based user-to-user messaging. The value determines the allotted time for a destination party to answer the phone (which is currently required before the first user-to-user messages can be sent to the extension). Avaya recommends a value between 10 and 20 seconds.

An authorized of an entry typically found in this field is as follows: 20

ALA_to_Connect_n

This field specifies the number of seconds a peripheral is kept in connect state before taking further action. This parameter is ignored for all peripherals and is provided for backwards compatibility issues.

An authorized of an entry typically found in this field is as follows: 10

ALA_to_Queued_n

This field specifies the number of seconds a peripheral is kept in camp-on-busy state before taking further action. This parameter is ignored for all peripherals and is provided for backwards compatibility issues.

An authorized of an entry typically found in this field is as follows: 15

ALA_Silence_intv_n

This field specifies the number of seconds a peripheral is left quiet (idle) before repeating any outstanding messages (also referred to as pace interval).

In many cases DECT users want to have a pace interval greater than zero, so that repeated messages do not pose an interruption. Therefore the DECT Messengerkeeps track of all active alarms, stores them in an internal database, and distributes them as the image of active alarms for a device is changing.

When no changes occur, the remaining alarms are repeated every ALA_Silence_intv_n specified number of seconds.

When a new alarm is generated and the image changes, the user is informed immediately.

On the other hand, when no changes occur, the outstanding messages are repeated at the specified interval.

An authorized of an entry typically found in this field is as follows: 120 (denotes 2 minutes)

ALA_Scroll_state_str

This field specifies the state in which a device must be to receive messages. Valid values are *CONNECT and *RINGING.

Scrolling starts at connect event when *CONNECT is specified, and starts at ringing event when *RINGING is specified.

This parameter is however, due to architectural reasons, currently ignored for most peripherals.

The value is used in eCSTA module release 2.8, where user-to-user messaging is also supported in alerting phase. Most other technologies of messaging are not call-oriented and do not have such requirements.

An authorized of an entry typically found in this field is as follows: *CONNECT

ALA_Scroll_intv_n

This field specifies the number of seconds that is used as scroll interval, when peripherals allow scrolling. This parameter is, due to architectural reasons, ignored for most peripherals and is provided for backwards compatibility issues.

An authorized of an entry typically found in this field is as follows: 3

ALA_Group_delivery_str

This value defines the degree of message delivery that is required on delivery of a message to a group. Values can be *ALL or *ANY and is only relevant if the field ALA_Remove_after_str is set to *SENT.

If the field ALA_Repeat_intv_n is set (value is greater than 0), than this field is only relevant if ALA_Confirm_action_str is set to *YES.

If the field value is *ALL, each individual recipient handles their messages on individual basis.

If the field value is *ANY, the message is only distributed to (at least) one group member. When the first user confirms, the message is considered delivered. This can result in removal of the message for all group members. This can mean some group members do not see the message at all.

An authorized of an entry typically found in this field is as follows: *ALL

ALA_Confirm_action_str

This value defines the confirm action. Valid entries are YES or NO.

If *NO is specified, message delivery confirmation is not required.

If of *YES, message delivery confirmation is mandatory.

This parameter is related to the ALA_Group_delivery_str parameter specified above.

Note that confirm delivery depends on a number of criteria, for authorized, alarm priority can have impact in defining whether an alarm required confirmation or not. Some other peripherals provide intrinsic message delivery (sending a normal E2 message through DMS-API) while others require use intervention (sending an urgent E2 message through DMS-API required user acknowledge). In some circumstances, special procedures apply to the confirmation action. This is defined in the corresponding eASYNC table and eSMTP table.

An authorized of an entry typically found in this field is as follows: *NO

ALA_Repeat_intv_n

This value defines the number of seconds between repeating alarm. Be careful not to confuse this entry with ALA_Silence_intv_n discussed above.

The ALA_Repeat_intv_n is in most cases 0, meaning the alarm system does not repeat active alarms. ALA_Repeat_intv_n is kept to 0 in situations where the alarm systems can set a SET and RESET, or when the alarm system sends an alarm once at SET.

The ALA_Repeat_intv_n is set to a value larger than 0 if the alarm system is incapable of sending a RESET indication, and repeats active alarms on frequent basis. When the appropriate alarms are no longer repeated, the situation is interpreted as a RESET condition. You can use this option to provide a steady repeat interval (for authorized, active alarms are repeated every 20 seconds) and a continuously repetition (repeat is not stopped after 10 repeats). When repeat interval is known, you can add a small safety factor (for authorized, add 5 to 10 seconds) and define the ALA_Repeat_intv_n as such.

An authorized of an entry typically found in this field is as follows: 0

ALA_Length_n

This field specifies the length of the alarm that is considered as relevant. Avaya recommends that you set the length to correspond to the length of the received alarm signal, although this is not always necessary. You can just as easily change messages in the alarm systems, so the length fits your environment and peripherals.

For instance, if you keep message length to 16 bytes or less, the messages fit on a single line on a DECT C4040 or DECT C4050 extension. This demand can result is instructions to the alarm vendor to properly align relevant information in the received alarm messages, so all needed text is left-adjusted and processed in DECT Messenger.

In some environments, longer messages are relevant. In such cases, you can specify, for authorized, message lengths of 100 bytes, if input comes from, for authorized, WEB interface and output goes to peripherals that are capable of handling long messages (eSMTP, eASYNC, and so on).

An authorized of an entry typically found in this field is as follows: 16

ALA_Trace_b

This parameter is a Boolean value and can be either True (-1) or False (0).

Specify the value True only for those alarms that are related to eWEB input program and generated using the Send Script Message function. These alarms are defined in the eWEB_SCRIPT table.

For all other alarms, set this value to False.

An authorized of an entry typically found in this field is as follows: False (-1)

ALA_Trace_dayToKeep_n

This value also refers to the trace function described in the ALA_Trace_b field.

Set this value to 0, unless the value ALA_Trace_b is set to True (-1). In this case, tracing is activated for the alarm, and the number of days to keep the trace data must be entered. A typical value is 14 days.

For all other alarms, set this value to 0.

An authorized of an entry typically found in this field is as follows: 0

ALA_Comments_str

This field can optionally be used by an administrator to store reminder information, describing, for authorized, the usage of the alarm.

An authorized of an entry typically found in this field is as follows: Reanimation through TELEVIC.

Table 29: eKERNEL_alarm sample data on page 275 provides sample eKERNEL_alarm module table data.

Table 29: eKERNEL_alarm sample data

Alarm	Inpgm	Descr	Remove after	Priority	...
1110101	11101	0	*SENT	3	...
1110102	11101	1	*SENT	1	...
1110103	11101	2	*RESET	2	...
1110104	11101	3	*SENT	3	...
1110105	11101	GUARDING	*SENT	10	...
1110201	11102	NUR	*SENT	10	...
1110202	11102	NUR	*RESET	10	...
1110203	11102	ASS	*SENT	7	...
1110204	11102	ASS	*RESET	7	...
1110205	11102	SAN	*SENT	10	...
1110206	11102	SAN	*RESET	10	...

Table: eKERNEL_ALARM

Alarm	Inpgm	Descr	Remove after	Priority	...
1110207	11102	REA	*SENT	1	...
1110208	11102	REA	*RESET	1	...
1110209	11102	1	*RESET	10	...
1110210	11102	1	*SENT	10	...
1110211	11102	*OTHER	*RESET	20	...
1110212	11102	*OTHER	*SENT	20	...
1110213	11102	GUARDING	*SENT	10	...
1110301	11103	API SENT	*SENT	10	...
1110302	11103	API RESET	*RESET	10	...
1110401	11104	GENERIC	*SENT	10	...
1110501	11105	1	*SENT	10	...
1110502	11105	2	*SENT	2	...
1140101	11401	EVACUATION	*RESET	2	...
1140102	11401	FIRE	*SENT	5	...
1140103	11401	TEST	*SENT	20	...
1150101	11501	REA	*SENT	999	...
1150102	11501	MUG	*SENT	999	...
1160101	11601	A-INPUT	*RESET	999	...
1160102	11601	A-INPUT	*SENT	999	...
1160103	11601	D-INPUT	*RESET	999	...
1160104	11601	D-INPUT	*SENT	999	...
1170101	11701	Short	*SENT	1	...
1170102	11701	Medium	*SENT	999	...
1170103	11701	Long	*SENT	999	...
1170104	11701	SCRIPT Message	*SENT	1	...
1170105	11701	SCRIPT Message	*RESET	1	...
1170106	11701	Short script	*SENT	10	...
1170107	11701	Medium script	*SENT	10	...
1170108	11701	Long script	*SENT	10	...
1180101	11801	SMTP	*SENT	10	...
1190101	11901	E2_MSG_N	*SENT	10	...

Alarm	Inpgm	Descr	Remove after	Priority	...
1190102	11901	E2_MSG_U	*SENT	2	...
1190103	11901	E2_NOODOPROEP	*SENT	1	...
1190104	11901	E2_REANIMATIE	*SENT	1	...
1190105	11901	E2_TEST_N	*SENT	5	...
1190106	11901	E2_TEST_U	*SENT	2	...
1210501	12105	1	*SENT	10	...
1210502	12105	2	*SENT	999	...
1210503	12105	3	*SENT	5	...
1310501	13105	1	*SENT	999	...
1310502	13105	2^9	*RESET	2	...
1310503	13105	NUR	*SENT	10	...
1310504	13105	NUR	*RESET	10	...

Table: eKERNEL_ALARM

Chapter 26: Table: eKERNEL_DEVICE

eKERNEL_DEVICE parameters

Name	Type	Size
DEV_Site_id_n	Integer	2
DEV_Area_id_n	Integer	2
DEV_id_str	Text	128
DEV_OUTPGM_str	Text	30
DEV_OUTPGM_facility_str	Text	50
DEV_Visual_dnr_str	Text	50
DEV_Descr_str	Text	100
DEV_PinCode_str	Text	10
DEV_Prty_n	Integer	2
DEV_Retry_count_ALT_DEV_id_n	Integer	2
DEV_Monitor_b	Yes/No	1
DEV_IoRegister_b	Yes/No	1
DEV_Div_Site_id_n	Integer	2
DEV_Div_Area_id_n	Integer	2
DEV_Div_dev_id_str	Text	128
DEV_Div_OUTPGM_Appl_str	Text	50
DEV_Div_OUTPGM_Facility_str	Text	50
DEV_Ras_Site_b	Yes/No	1
DEV_Ras_Area_b	Yes/No	1
DEV_Comments_str	Text	255

DEV_site_id_n

This field refers to the site as specified in eKERNEL_SITE table. Usually this field has value 1. In most environments, this value is 1.

An authorized of an entry typically found in this field is as follows: 1

DEV_Area_id_n

This field refers to the area identifier, as defined in the eKERNEL_AREA table. In most environments, this value is 1.

An authorized of an entry typically found in this field is as follows: 1

DEV_id_str

This field contains a reference to the destination device as known in our internal infrastructure. When a device is, for instance, a DECT extension, this field specifies the extension number (for authorized, 865). When a mail destination is defined, this field contains a mail address (for authorized, francis.missiaen@1s.be). As such the next field GRP_OUTPGM_Appl_str further identifies the device for a specific site and area.

GRP_Dev_id_str, GRP_OUTPGM_Appl_str, DEV_Site_id_n and DEV_Area_id_n must be handled to uniquely identify a device.

An authorized of an entry typically found in this field is as follows: 1 – 1 – 865 – eDMSAPI or 1 – 2 – francis.missiaen@1s.be - eSMTP

DEV_OUTPGM_str

This field identifies the application that processes the request.

A device can be defined more than once. For authorized DECT extension 865 can be defined for eDMSAPI, eCSTA or eVBVOICE. The indicated application handles the message using the capabilities of the infrastructure. eDMSAPI can for instance send LRMS data profile messages (non-voice-call) to extensions such as DECT C944 and i600). eCSTA can for instance send user-to-user messages to voice-call based peripherals, such as ErgoLine D330, ErgoLine D340, Dect C311, Dect C911, Dect C322, Dect C922, Dect C933... and eVBVOICE can inform the user with an audible message. The list of output devices can be extended in time. The supported values are currently:

- eASYNC

  for sending SMS to PROXIMUS or KPN and PAGING to BELGACOM
- eDMSAPI

  for sending E2 messages
- eCSTA

  for sending voice-call related user-to-user messages
- eESPA

  for sending messages to ESPA 4.4.4 interface
- eIO

  for enabling/disabling discrete output contacts
- eSMS

  for sending SMS message to mobile GSM phones

- eSMTP

  for sending mail to SMTP-compliant infrastructure
- eVBVOICE

  for sending audible messages

DEV_OUTPGM_facility_str

The indicated application handles the message using the capabilities of the infrastructure.

The supported values are specified in the field FMT_OUTPGM_Facility_str of the eKERNEL_DEVICE_FORMAT table for the corresponding output program.

An authorized of an entry typically found in this field is as follows: C4050 for eDMSAPI

DEV_Visual_dnr_str

When this field is entered for a device, the 'Visual DNR' is used to format a message when it contains [Calling number], so the end-user is confronted with the visual DNR. The default value for this field is empty. (This field is new in release 3.0.)

Avaya recommends using this field in Avaya environments only.

An authorized of an entry typically found in this field is as follows: 2000 for DECT handset with DNR 2000 and hardware ID 00300 (DMC)

DEV_Descr_str

This description is used to show information on devices in the eWeb module. Avaya recommends adding the name of the owner of the device.

An authorized of an entry typically found in this field is as follows: DECT: Kristien Daneels

DEV_PinCode_str

Some business processes need a confirmation of end-user. Some technologies provide this during alarm notification, such as LRMS messaging on eDMSAPI allow using “OK” to confirm message delivery. However, some notification technologies do not offer immediate end-user confirmation during notification: eASYNC, eSMTP, eSMS and so on.

Some modules provide inbound confirmation: eCSTA by means of inbound voice call (based upon CLIP), eVBVOICE by means of inbound voice call (based on DTMF entered pincode)

and eSMS my means of inbound SMS message (based on CLIP of mobile phone or pincode in SMS message).

With these pincodes, all messages for all devices with a matching pincode can be cleared. For authorized, pincode 12345 clears the alarms for those devices that specify 12345 in the DEV_PinCode_str field.

An authorized of an entry typically found in this field is as follows: 12345.

DEV_Prty_n

This field is currently not implemented, but is foreseen for future enhancements.

DEV_Retry_count_ALT_DEV_id_n

This field is implemented in a different fashion after eKERNEL version 2.1.0:

- Before eKERNEL Version 2.1.0:

  The number of retries before switching to an alternative device, if device (site + area + device + output program is unique) is defined in the eKERNEL_device_alt table.

  The default value is 30, which means that if an alarm has a silence interval of for instance 120 seconds; the alarm is removed for this device after one hour (and set for the alternative device if defined).

  For authorized, 1 => after the second retry, the alternative devices is set.

- eKERNEL Version 2.1.1 and later:

  This keyword defines how many times the application tries to deliver the message before switching to an alternative device if defined in the eKERNEL_device_alt table.

  The default value is 30, which means that if an alarm has a silence interval of for instance 120 seconds; the alarm is removed for this device after one hour (and set for the alternative device if defined).

  The value = 0 means that the application never tries to send the message to an alternative device, and that the alarm is sent to the device every silence interval (ALA_Silence_intv_n in eKERNEL_Alarm) until the alarm is reset by, for authorized, the input program.

  The value = 1 means that after 1 try, the application clears the message for this device, and sends the message to the alternative device if defined in the eKERNEL_Device_alt table.

Important:

In this case, the switch to the alternative device is immediate, which means that there is no silence interval between those two calls. Therefore, be very careful that there are no loop conditions defined in the eKERNEL_device_alt table.

The value = 2 means that after the second try, the alternative device is contacted.

For authorized, 2 => after 2 times trying to send the message, the alternative devices is set.

DEV_Monitor_b

All devices with the value True (-1) are sent to the eCSTA application and must be monitored for there divert behavior. Those devices that are diverted are sent to the eKernel application.

An authorized of an entry typically found in this field is as follows: False (-1).

DEV_IoRegister_b

Set this field to “false for all devices that are not assigned to eDMSAPI module.

For devices assigned to eDMSAPI module, specify True for devices that generate action using eDMSAPI module, for *IA (inbound alarm), *IC (inbound confirm) or *LA (location alarm).

This forces a IORegistration in eDMSAPI, allowing the application to be able to monitor inbound LRMS activities on the monitored DECT handset.

An authorized of an entry typically found in this field is as follows: -1

DEV_Div_Site_id_n

This field specifies the site of the diverted device.

When a device is diverted to another device (eCSTA), the system ignores the divert in cases where the destination device is not configured in the eKERNEL_DEVICE table. When more than one device is defined the eDMSAPI device type is selected, and the corresponding site is entered in this field. If no eDMSAPI capable device is defined, the first available matching device is used, and the corresponding site is entered in this field.

An authorized of an entry typically found in this field is as follows: -1

DEV_Div_Area_id_n

This field specifies the area of the diverted device.

See DEV_Div_Site_id_n

An authorized of an entry typically found in this field is as follows: 1

DEV_Div_OUTPGM_Appl_str

This field specifies the output program of the diverted device.

See DEV_Div_Site_id_n

An authorized of an entry typically found in this field is as follows: 1

DEV_Div_OUTPGM_Facility_str

This field specifies the output program of the diverted device.

See DEV_Div_Site_id_n

An authorized of an entry typically found in this field is as follows: eDMSAPI

DEV_Ras_Site_b

This field is a Boolean value and can be either True (-1) or False (0). The default value is False (0).

This field is currently not implemented, but is reserved for future enhancements when multi-site facilities are implemented.

In future versions, eKERNEL-to-eKERNEL communications will be implemented, so alarms for devices located on another site can be sent to the remote eKernel.

An authorized of an entry typically found in this field is as follows: False (0)

DEV_Ras_Area_b

This field is a Boolean value and can be either True (-1) or False (0). The default value is False (0).

This field specifies the behavior of the eWEB-based function Send DMS-API Message. The Send DMS-API message default only presents those devices that are defined in the eKERNEL_DEVICE table, and have output program eDMSAPI and reside on the same site and area as the eWEB input program. For authorized, if the eWEB application is defined on site 1 and area 1, the Send DMS-API Message presents the eDMSAPI devices of site 1 area 1.

Some multi-area environments require that you present devices that are configured for a remote area. You can select for each device whether the remote device is available to the local eWEB area or not.

An authorized of an entry typically found in this field is as follows: False (0)

DEV_Comments_str

This field can contain remarks from the administrator, and is informational only.

Table: eKERNEL_DEVICE

Chapter 27: Table: eKERNEL_DEVICE_ALT

eKERNEL_DEVICE_ALT parameters

Name	Type	Size
ALT_Dev_Site_id_n	Integer	2
ALT_Dev_Area_id_n	Integer	2
ALT_Dev_id_str	Text	128
ALT_OUTPGM_Appl_str	Text	30
ALT_Sequence_n	Integer	2
ALT_Alt_DEV_Site_id_n	Integer	2
ALT_Alt_DEV_Area_id_n	Integer	2
ALT_Alt_dev_id_str	Text	128
ALT_Alt_OUTPGM_Appl_str	Text	30
ALT_Alt_OUTPGM_Facility_str	Text	50
ALT_Descr_str	Text	255
ALT_Comments_str	Text	255

ALT_Dev_Site_id_n

This field refers to the site as specified in eKERNEL_SITE table. Usually this field has value 1. In most environments, this value is 1.

An authorized of an entry typically found in this field is as follows: 1

ALT_Dev_Area_id_n

This field refers to the area identifier, as defined in the eKERNEL_AREA table. In most environments, this value is 1.

An authorized of an entry typically found in this field is as follows: 1

ALT_Dev_id_str

This field defines – in combination with ALT_Dev_Site_id_n, ALT_Dev_Area_id_n and ALT_OUTPGM_Appl_str – a device in the system. The record specifies one or more alternate devices that are to be used in case an unrecoverable error occurs when sending a message to a specified device. In case of a failure, a list of alternate devices can be processed upon successful message delivery.

Define the device (site, area, device and outpgm) as a valid device in eKERNEL_DEVICE table.

An authorized of an entry typically found in this field is as follows: 865

ALT_OUTPGM_Appl_str

The field is associated with the previous field and defines the device.

An authorized of an entry typically found in this field is as follows: eDMSAPI

ALT_Sequence_n

This field is a sequence number to make a record definitions in eKERNEL_DEVICE_ALT unique. Avaya recommends starting with a value of 1 and incrementing by 1s.

An authorized of an entry typically found in this field is as follows: 1

ALT_Alt_DEV_Site_id_n

This field defines, in combination with ALT_Alt_DEV_area_id_, ALT_Alt_dev_id_str, ALT_Alt_OUTPGM_Appl_str and ALT_Alt_OUTPGM_Facility_str the alternate device.

An authorized of an entry typically found in this field is as follows: 1

ALT_Alt_DEV_area_id_n

This field defines, in combination with ALT_Alt_DEV_site_id_, ALT_Alt_dev_id_str, ALT_Alt_OUTPGM_Appl_str and ALT_Alt_OUTPGM_Facility_str the alternate device.

An authorized of an entry typically found in this field is as follows: 1

ALT_Alt_dev_id_str

This field defines, in combination with ALT_Alt_DEV_Site_id_n, ALT_Alt_DEV_area_id_, ALT_Alt_OUTPGM_Appl_str and ALT_Alt_OUTPGM_Facility_str the alternate device.

Check for possible loop conditions when setting up this table.

An authorized of an entry typically found in this field is as follows: 865

ALT_Alt_OUTPGM_Appl_str

This field defines, in combination with ALT_Alt_DEV_Site_id_n, ALT_Alt_DEV_area_id_, ALT_Alt_dev_id_str and ALT_Alt_OUTPGM_Facility_str the alternate device.

Check for possible loop conditions when setting up this table.

An authorized of an entry typically found in this field is as follows: eDMSAPI

ALT_Alt_OUTPGM_Facility_str

This field defines, in combination with ALT_Alt_DEV_Site_id_n, ALT_Alt_DEV_area_id_, ALT_Alt_dev_id_str and ALT_Alt_OUTPGM_Appl_str the alternate device.

Check for possible loop conditions when setting up this table.

An authorized of an entry typically found in this field is as follows: C4050

ALT_descr_str

This informational field can contain some remarks (informational only)

ALT_Comments_str

This field is used for an administrator to add remarks and is used informational only.

The following provides sample eKERNEL_DEVICE_ALT table data.

	ALT_Dev	ALT_	ALT_Dev_id_str	ALT_OUTPGM_Appl	ALT_Sequence_n	ALT_Al	ALT_A	ALT_Alt_dev_id_str	ALT_Alt_OUTP	ALT_Alt_OUTPGM_Fa	ALT_Descr_str
	1										
▶	1	1	003292802249	eASYNC	1	1	1	32475112233	eASYNC	PROXIMUS	
	1	1	003292802249	eASYNC	2	1	1	240	eDMSAPI	C922	
	1	1	1900	eDMSAPI	1	1	1	861	eDMSAPI	C922	
	1	1	32475111111	eASYNC	1	1	1	32475112233	eASYNC	PROXIMUS	
	1	1	922	eDMSAPI	1	1	1	922	eCSTA	C922	
	1	1	922	eDMSAPI	2	1	1	kristien.daneels@1s.be	eSMTP	SMTP	
	1	1	933	eDMSAPI	1	1	1	32475112233	eASYNC	PROXIMUS	
	1	1	francis.missiaen@1s.be	eSMTP	1	1	1	32475353215	eASYNC	PROXIMUS	

Figure 190: eKERNEL_DEVICE_ALT parameters

Table: eKERNEL_DEVICE_ALT

Chapter 28: Table: eKERNEL_DEVICE_FORMAT

eKERNEL_DEVICE_FORMAT parameters

Name	Type	Size
FMT_OUTPGM_Appl_str	Text	30
FMT_OUTPGM_Facility_str	Text	50
FMT_Bytes_line1_n	Integer	2
FMT_Bytes_line2_n	Integer	2
FMT_Bytes_line3_n	Integer	2
FMT_Page_ind_n	Integer	2
FMT_Page_more_ind_n	Integer	2
FMT_Concatination_b	Yes/No	1
FMT_Scroll_depth_n	Integer	2
FMT_AllowEmergency_b	Yes/No	1
FMT_Descr_str	Text	250
FMT_Comments_str	Text	255

Figure 191: eKERNEL_DEVICE_FORMAT parameters listing

FMT_OUTPGM_Appl_str

This field identifies the output program. The following options are supported: eASYNC, eCSTA, eDMSAPI, eESPA, eIO, eSMS, eSMTP and eVBVOICE.

An authorized of an entry typically found in this field is as follows: eDMSAPI

FMT_OUTPGM_Facility_str

This field specifies the supported facility or facilities for a specified output program. See Table 30: Application-Facility associations on page 291 for supported entries. The administrator can create new facilities.

Table 30: Application-Facility associations

Application	Facility
eASYNC	PAGING
eASYNC	PROXIMUS

Application	Facility
eASYNC	KPN
eCSTA	C311
eCSTA	C322
eCSTA	C911
eCSTA	C922
eCSTA	C933
eCSTA	D330
eCSTA	D340
eCSTA	P375D
eDMSAPI	C922
eDMSAPI	C933
eDMSAPI	C944
eDMSAPI	I600
eESPA	ESPA
eIO	DO
eSMS	SMS
eSMTP	SMTP

FMT_Bytes_line1_n

This field specifies the number of bytes available on the first line. In general, the maximum length is to be used. Refer to the sample data in Table 33: eKERNEL_DEVICE_FORMAT sample data on page 295 for authorizeds.

An authorized of an entry typically found in this field is as follows: 16

FMT_Bytes_line2_n

This field specifies the number of bytes available on the second line. In general, this value is 0 for devices with no second line and the maximum length, in case a second line is available. If only two lines are available, a smaller number of bytes is appropriate to reserve room for page indication and so on. Refer to the sample data in Table 33: eKERNEL_DEVICE_FORMAT sample data on page 295 for authorizeds.

An authorized of an entry typically found in this field is as follows: 16

FMT_Bytes_line3_n

This field specifies the number of bytes available on the third line. In general, the value is smaller than the actual available size to reserve room for page indication and more indication.

When a customer has infrastructure with extensions capable of displaying three lines of 16 bytes, alarm lengths up to 48 bytes can be displayed (without page indication and more indication). In most cases, Avaya recommends that you reserve the third line for page indication and more indication, thus specifying 0 for the third line.

An authorized of an entry typically found in this field is as follows: 0

FMT_Page_ind_n

This field specifies the number of bytes reserved for page indication. Recommended value is five bytes, which allows the XX/XX syntax. A lower number of characters can be used if space is limited. See Table 31: Page identification syntax on page 293 for authorized values.

Table 31: Page identification syntax

0	(no page indication)
1	+
2	+
3	X/X
4	X/X
5	XX/XX
6	XX/XX

Note:

This value is only implemented on the eDMSAPI and eCSTA output programs.

An authorized of an entry typically found in this field is as follows: 5

FMT_Page_more_ind_n

This field specifies the number of bytes reserved for more indication. Recommended value is 2 bytes, which allows a + syntax. A lower number of characters can be used in space is limited. See Table 32: More indication syntax on page 294 for authorized values.

Table 32: More indication syntax

0	(no more indication)
1	+
2	+

Note:

This value is only implemented on eDMSAPI and eCSTA output programs.

An authorized of an entry typically found in this field is as follows: 2

FMT_Concatination_b

This field defines whether small messages that fit on one display are merged to one page. If, for authorized, a DECT C933 extension is defined as 16/16/0/5/2 and messages are a maximum 16 bytes, you can show two messages on a single page.

Note:

This value is only implemented on eDMSAPI and eCSTA output programs.

An authorized of an entry typically found in this field is as follows: -1 (true)

FMT_Scroll_depth_n

This field specifies the maximum number of pages that is shown to a user. If scroll depth is 4 and there are seven pages available, the user is only informed on the first four pages. A more indication is shown to indicate more pages, unless this is suppressed.

Note:

Do not specify any value larger than 4 for the eCSTA output program, due to limitations in internal resources.

An authorized of an entry typically found in this field is as follows: 4

FMT_AllowEmergency_b

This field is introduced in R3.0 and defines whether the peripheral supports Emergency LRMS Messaging. Currently this feature is only supported on DECT C944 devices. Sending an emergency message through eDMSAPI module to a peripheral that does not support this feature, resulting in a system malfunction. Administrators must carefully assign the device facility that enables emergency calls only to peripherals that support it. Assign the facility only

to peripherals that support it. To prevent problems, the default equals false, so enabling emergency calls on supported devices is performed only on demand.

Note:

This value is only implemented on C944 devices.

An authorized of an entry typically found in this field is as follows: 0 (false)

FMT_Descr_str

An administrator can enter a description of the template in this field. This value is informational only.

An authorized of an entry typically found in this field is as follows: template for C933 extensions for nurse-calls

FMT_Comments_str

An administrator can enter remarks in this field. This value is informational only.

An authorized of an entry typically found in this field is as follows: two lines and indicators.

Table 33: eKERNEL_DEVICE_FORMAT sample data on page 295 provides sample eKERNEL_DEVICE_FORMAT table data.

Table 33: eKERNEL_DEVICE_FORMAT sample data

Application	Facility	Line 1	Line 2	Line 3	Page	More	Concat	Scroll depth
eASYNC	PAGING	160	0	0	5	2	0	999
eASYNC	PROXIMUS	120	0	0	0	0	0	999
eCSTA	C311	10	0	0	0	0	0	4
eCSTA	C322	10	0	0	0	0	0	4
eCSTA	C911	16	16	0	0	0	-1	4
eCSTA	C922	16	16	0	5	2	-1	4
eCSTA	C933	16	16	0	5	2	-1	4
eCSTA	D330	12	0	0	5	2	0	4
eCSTA	D340	20	0	0	0	0	0	4
eCSTA	P375D	19	0	0	0	0	0	4

Table: eKERNEL_DEVICE_FORMAT

Application	Facility	Line 1	Line 2	Line 3	Page	More	Concat	Scroll depth
eDMSAPI	C4040	16	16	0	5	2	-1	999
eDMSAPI	C4050	16	16	0	5	2	-1	999
eIO	DO	1024	0	0	0	0	0	999
eSMTP	SMTP	32	0	0	0	0	0	999
eVBVOICE	VBVOICE	1024	0	0	0	0	0	999
eESPA	ESPA	128	0	0	0	0	0	999

Chapter 29: Table: eKERNEL_GROUP

eKERNEL_GROUP parameters

Name	Type	Size
GRP_id_str	Text	128
GRP_InPGM_id_n	Long Integer	4
GRP_Name_str	Text	128
GRP_Descr_str	Text	50
GRP_Comments_str	Text	255

Figure 192: eKERNEL_GROUP parameters listing

GRP_id_str

The field defines a unique identifier for a group. The field is a unique key in the database.

Avaya recommends defining group identifiers using the following naming convention:

Table 34: Recommended Group identifier naming convention

Byte 1-5	Input program	
	Byte 1	Site of input program
	Byte 2	Area of input program
	Byte 3	Input program type
		1 - eCAP or eAPI or eESPA
		6 - eIO
		7 - eWEB
		8 - eSMTP_server
	Byte 4-5	Input program sequence number
Byte 6	(Underscore character)	
Byte 7-...	Group name	

Example: 31101_00001 denotes site 3, area 1, input program type eCAP or eAPI, input program sequence 01, group name 00001.

For each defined group, one or more group member must be defined in the eKERNEL_GROUP_MEMBER table.

You can assign authority to the groups by means of the eKERNEL_GROUP_AUTH table.

An authorized of an entry typically found in this field is as follows: 31101_00001

GRP_InPGM_id_n

As described above, group identifiers are uniquely defined by combining input program identifier and group name.

The input program is the value specified in the eKERNEL_INPGM table.

Avaya recommends following the naming convention set out in Table 35: Recommended Group identifier naming convention on page 298.

Table 35: Recommended Group identifier naming convention

Byte 1-5	Input program	
	Byte 1	Site of input program
	Byte 2	Area of input program
	Byte 3	Input program type
		1 - eCAP or eAPI or eESPA
		6 - eIO
		7 - eWEB
		8 - eSMTP_server

Example: 31101 denotes site 3, area 1, input program type eCAP or eAPI and input program sequence 01.

An authorized of an entry typically found in this field is as follows: 31101

GRP_Name_str

As described above, group identifiers are uniquely defined by combining input program identifier and group name.

The input program is the value specified in the eKERNEL_INPGM table.

The group name field is the group indication that is typically received from the external alarm system. In many environments, alarm systems are capable of sending some kind of destination

information in the alarm string. This can, for authorized, be referred to with terms such as paging number, group, or destination.

Note that the above-described design allows sharing the same group name between multiple input programs. A first eCAP instance can have a different understanding for group 00001 than a second eCAP instance. In most cases the group names are determined by third-party vendors, and in many environments cannot be changed.

With this approach, you can logically link any group name and assign our internally known group members (peripherals) to them.

An authorized of an entry typically found in this field is as follows: 00001

GRP_Descr_str

This field can have a descriptive text, to allow administrators to easily recognize the group.

An authorized of an entry typically found in this field is as follows: Intensive Care

GRP_Comments_str

This field can also contain additional information.

An authorized of an entry typically found in this field is as follows: "Warning: minimum 3 DECT extensions required"

Table 36: eKERNEL_GROUP sample data on page 299 provides sample eKERNEL_GROUP table data.

Table 36: eKERNEL_GROUP sample data

Group id	Input program	Group name	Description	Comments
31101_00001	31101	00001	Test from eCAP	
31102_00001	31102	00001	Test from eCAP	
31102_24960	31102	24960	Test Televic	
31103_00001	31103	00001	Test from eAPI	
31601_00001	31601	00001	Test from eIO	
31701_eASYNC	31701	eASYNC	Test to eASYNC	
31701_eDMSAPI	31701	eDMSAPI	Test to eDMSAPI	
31701_eIO	31701	eIO	Test to eIO	
31701_eSMTP	31701	eSMTP	Test to eSMTP	

Table: eKERNEL_GROUP

Group id	Input program	Group name	Description	Comments
31801_00001	31801	00001	Test from eSMTP	

Chapter 30: Table: eKERNEL_GROUP_AUTH

eKERNEL_GROUP_AUTH parameters

Name	Type	Size
GRPA_GRP_id_str	Text	128
GRPA_UserID_str	Text	10
GRPA_Comments_str	Text	255

Figure 193: eKERNEL_GROUP_AUTH parameters listing

GRPA_GRP_id_str

This field refers to the unique group identifier, as described in the eKERNEL_GROUP table. Each group identifier must be defined in the eKERNEL_GROUP table. The member of each group identifier must be defined in the eKERNEL_GROUP_MEMBER table. At least one group member per group identifier must be defined, because empty groups result in loss of alarms.

The table eKERNEL_GROUP_AUTH allows an administrator to grant access to eWEB users. In eWEB, there is a group maintenance function: Work with Groups. User without all object authority in their eWEB_USER_AUTH table definition can see only those groups that are defined in the eKERNEL_GROUP_AUTH table.

A typical authorized is a hospital, where the person responsible for a department is allowed to maintain only their own departmental groups, and not the groups of other departments.

An authorized of an entry typically found in this field is as follows: 31101_00001

GRPA_UserID_str

This field specifies the username that is granted access to the group. This value must match the definition of the users in eWEB_USER_AUTH table.

A special value *ALL is implemented. If you specify this special value, all users have access to this group. With *ALL you do not need to enter all individual users, but as a result you have no granular authority definition because all users are granted access.

Note that eWEB only allows maintenance of the groups that are assigned to input programs of the same site as the eWEB. This means a eWEB instance of site 1 only allows maintenance of groups of site 1.

An authorized of an entry typically found in this field is as follows: FMI

GRPA_Comments_str

This field can contains remarks of an administrator, and is informational only.

eKERNEL_GROUP_AUTH parameters on page 301 provides sample eKERNEL_GROUP_AUTH table data.

Table 37: eKERNEL_GROUP_AUTH sample data

Group id	User id	Comments
31101_00001	FMI	
31102_00001	KDS	
31102_24960	*ALL	

Chapter 31: Table: eKERNEL_GROUP_MEMBER

eKERNEL_GROUP_MEMBER parameters

Name	Type	Size
GRPM_GRP_id_str	Text	128
GRPM_Dev_id_str	Text	128
GRPM_Dev_Site_id_n	Integer	2
GRPM_Dev_Area_id_n	Integer	2
GRPM_OUTPGM_Appl_str	Text	30
GRPM_From_str	Text	5
GRPM_To_str	Text	5
GRPM_Mon_b	Yes/No	1
GRPM_Tue_b	Yes/No	1
GRPM_Wed_b	Yes/No	1
GRPM_Thu_b	Yes/No	1
GRPM_Fri_b	Yes/No	1
GRPM_Sat_b	Yes/No	1
GRPM_Sun_b	Yes/No	1
GRPM_Holiday_b	Yes/No	1
GRPM_Activate_timestamp_str	Text	14
GRPM_Desactivate_timestamp_str	Text	14
GRPM_Comments_str	Text	255

Figure 194: eKERNEL_GROUP_MEMBER parameters listing

GRPM_GRP_id_str

The field defines a unique identifier for a group. The field is a unique key in the database.

Avaya recommends defining group identifiers using the following naming convention:

Table 38: Recommended Group identifier naming convention

Byte 1-5	Input program	
	Byte 1	Site of input program
	Byte 2	Area of input program
	Byte 3	Input program type
		1 - eCAP or eAPI or eESPA
		6 - eIO

		7 - eWEB
		8 - eSMTP_server
		9 - eDMSAPI
	Byte 4-5	Input program sequence number
Byte 6	(Underscore character)	
Byte 7-...	Group name	

Example: 31101_00001 denotes site 3, area 1, input program type eCAP or eAPI, input program sequence 01, group name 00001.

Each group must be defined in the eKERNEL_GROUP table.

For each defined group, one or more group member must be defined in the eKERNEL_GROUP_MEMBER table.

You can assign authority to the groups by means of the eKERNEL_GROUP_AUTH table. See documentation Table_eKERNEL_GROUP_AUTH.pdf.

An authorized of an entry typically found in this field is as follows: 31101_00001

GRPM_Dev_id_str

This field contains a reference to the destination peripheral as it is known in the internal infrastructure. The site, area, output program application, and device identifier identify peripherals. These four values define a peripheral unambiguously.

A number of sample records are shown in Table 39: GRPM_Dev_id_str sample records on page 304.

Table 39: GRPM_Dev_id_str sample records

Site	Area	Device	Output program	Facility
1	1	32479638338	eASYNC	PROXIMUS
1	1	865	eDMSAPI	C4050
1	1	9789074	eASYNC	PAGING
1	1	475353215	eASYNC	PROXIMUS
1	1	bekds@1s.be	eSMTP	SMTP
1	1	DO_03_01	eIO	DO
1	1	DO_03_02	eIO	DO
1	1	DO_03_03	eIO	DO
1	1	DO_03_04	eIO	DO

Site	Area	Device	Output program	Facility
1	1	DO_03_05	eIO	DO
1	1	DO_03_06	eIO	DO
1	1	DO_03_07	eIO	DO
1	1	DO_03_08	eIO	DO
1	1	francis.missiaen@1s.be	eSMTP	SMTP
1	1	kristien.daneels@1s.be	eSMTP	SMTP

GRPM_Dev_Site_id_n

This value refers to the site identifier of the input program that is associated with the group. Refer to Table: eKERNEL_SITE on page 331 for more details on the site parameter.

An authorized of an entry typically found in this field is as follows: 1

GRPM_Dev_Area_id_n

This value refers to the area identifier of the input program that is associated with the group. Refer to Table: eKERNEL_AREA on page 265 for more details on the site parameter.

An authorized of an entry typically found in this field is as follows: 1

GRP_OUTPGM_Appl_str

This field provides the output program application identifier of the application that processes the request.

A device can be used more than once depending of the used output program. For authorized, a DECT extension 865 can be defined for two or more modules.

The indicated application handles the message using the capabilities of the infrastructure. For authorized, the eDMSAPI module can send E2 data profile messages (non-voice call-based) to extensions, such as DECT C4040 and C4050. The supported values are shown in Table 40: Supported output applications on page 306:

Table 40: Supported output applications

```
eASYNC      for sending SMS to PROXIMUS/KPN and PAGING to BELGACOM
eCSTA       for sending voice-call related user-to-user messages
eDMSAPI     for sending E2 data messages to DECT C922 and C933 sets
eESPA       for sending messages t ESPA 4.4.4 infrastructure
eIO         for enabling/disabling discrete output contacts
eSMTP       for sending mail to SMTP compliant infrastructure
eVBVOICE    for sending wave files through voice-calls
```

GRP_From_str

This value specifies an hour and time in the format xx:xx. The valid range is 00:00 to 23:59; values outside this range produce unpredictable results. GRP_From_str denotes the start of the time interval where the defined device is an active member of the specified group GRP_Name_str. For authorized, 00:00 indicates the group-member is active at midnight, and 12:00 indicates the group-member starts at noon. The active period ends at the time specified in GRP_To_str.

An authorized of an entry typically found in this field is as follows: 00:00

GRP_To_str

This value specifies an hour and time in the format xx:xx. The valid range is 00:00 to 23:59; values outside this range produce unpredictable results. GRP_To_str denotes time when the defined device ceases to be an active member of the specified group GRP_Name_str. For authorized, 23:59 indicates the group-membership expires at midnight, and 12:00 indicates that the group-membership expires at noon. The active time period begins at the time specified in GRP_From_str.

Note:

GRP_From_str can be larger than GRP_To_str: In this case, a job can start at 21:00 and end at 06:00 (night-shift).

Note:

A device can be active from for more than one period of time on a given day. For authorized: 08:00-12:00 and 13:15-17:30; in this case, two group members must be defined, one of 08:00-12:00 and another with 13:15-17:30.

To clarify the possible values, authorizeds are shown in Table 41: Group member schedule authorizeds: on page 307.

Table 41: Group member schedule authorizeds:

From	To	Remark
00:00	23:59	Member is active 24/24 hr (day and night)
06:30	13:30	Member is active from 06:30 to 13:30
21:00	06:00	Member is active from 21:00 till 06:00

GRP_Mon_b

This value specifies whether the group-member record is active on Mondays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Mondays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GRP_Tue_b

This value specifies whether the group-member record is active on Tuesdays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Tuesdays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GRP_Wed_b

This value specifies whether the group-member record is active on Wednesdays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Wednesdays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GRP_Thu_b

This value specifies whether the group-member record is active on Thursdays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Thursdays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GRP_Fri_b

This value specifies whether the group-member record is active on Fridays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Fridays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GRP_Sat_b

This value specifies whether the group-member record is active on Saturdays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Saturdays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GRP_Sun_b

This value specifies whether the group-member record is active on Sundays. Accepted values are True (-1) or False (0).When -1 is specified, the group-member record is active on Sundays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GRP_Holiday_b

This value specifies whether the group-member record is active on holidays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on holidays. When 0 is specified, the record is not active on this day.

Note:

The term Holiday refers to the dates defined in the eKERNEL_HOLIDAY table. At installation time, a few dates are defined. The tables must be maintained by an administrator. You can use this calendar for other purposes, such as indicating official closing days, if this is suitable to your working environment.

An authorized of an entry typically found in this field is as follows:-1

GRPM_Activate_timestamp_str

This field specifies the timestamp when the record becomes activated. The format is YYYYMMDDHHMMSS.

The GRPM_Activate_timestamp_str and GRPM_Desactivate_timestamp_str fields can be used to define a time interval, where records are active. This functionality allows to anticipate on future changes in availability of staff, and is typically used in environments where planning is needed for staff, regimes, changing schedules, holiday period, and so on.

An authorized of an entry typically found in this field is as follows: 20010101000000

GRPM_Desactivate_timestamp_str

This field specifies the timestamp when the record becomes deactivated. The format is YYYYMMDDHHMMSS.

The GRPM_Activate_timestamp_str and GRPM_Desactivate_timestamp_str fields can be used to define a time interval, where records are active. This functionality allows to anticipate on future changes in availability of staff, and is typically used in environments where there is need for on-front planning of staff, regimes, changing schedules, holiday period, and so on.

An authorized of an entry typically found in this field is as follows: 20991231235959

GRP_Comments_str

This field can optionally be used by an administrator to store reminder information, describing, for authorized, a description of the file usage.

An authorized of an entry typically found in this field is as follows: Backup of regular anesthetist during holidays

Table: eKERNEL_GROUP_MEMBER

Chapter 32: Table: eKERNEL_GUARDING

eKERNEL_GUARDING parameters

GUA_INPPGM_id_n	long	4
GUA_From_str	Text	5
GUA_To_str	Text	5
GUA_Mon_b	Yes/No	1
GUA_Tue_b	Yes/No	1
GUA_Wed_b	Yes/No	1
GUA_Thu_b	Yes/No	1
GUA_Fri_b	Yes/No	1
GUA_Sat_b	Yes/No	1
GUA_Sun_b	Yes/No	1
GUA_Timeout_n	Integer	2
GUA_msg_str	Text	255
GUA_GRP_Name_str	Text	50
GUA_ALA_id_n	long	4
GUA_Comments_str	Text	255

Figure 195: eKERNEL_GUARDING parameters listing

GUA_INPPGM_id_n

This field specifies the unique identifier of the input program. Note that this identifier is defined in the eKERNEL_TCPCLIENT table (field TCPCLIENT_INPGM_id_n). Refer to Table: eKERNEL_TCPCLIENT on page 339 for more information on how to set up these input programs.

An authorized of an entry typically found in this field is as follows: 11101

GUA_From_str

This value specifies an hour and time in the format xx:xx. The valid range is 00:00 to 23:59; values outside this range produce unpredictable results. GUA_From_str denotes the start of the time interval during which the guarding facility is active. If the eKERNEL module does not receive any requests (message request, configuration request, and so on) from the input program during the GUA_Timeout_n interval, a guarding alarm is activated.

An authorized of an entry typically found in this field is as follows: "00:00"

GUA_To_str

This value specifies an hour and time in the format xx:xx. The valid range is 00:00 to 23:59; values outside this range produce unpredictable results. The value denotes the end of the time period during which the guarding facility is active.

The active time period begins at the time specified in GUA_From_str.

Note:

GUA_From_str can be larger than GUA_To_str, resulting, for authorized, in a job that starts at 21:00 and ends at 06:00.

Note:

A device can be active from for more than one period of time on a given day. For authorized: 08:00-12:00 and 13:15-17:30; in this case, two group members must be defined, one of 08:00-12:00 and another with 13:15-17:30.

If the same time is specified in more than one case, only the first record is processed.

Table 42: Guarding schedule authorizeds on page 312 shows authorizeds of Guarding schedules.

Table 42: Guarding schedule authorizeds

From	To	Remark
00:00	23:59	Guarding is active 24/24 hr (day and night)
06:30	13:30	Guarding is active from 06:30 to 13:30
21:00	06:00	Guarding is active from 21:00 till 06:00

GUA_Mon_b

This value specifies whether the group-member record is active on Mondays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Mondays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GUA_Tue_b

This value specifies whether the group-member record is active on Tuesdays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Tuesdays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GUA_Wed_b

This value specifies whether the group-member record is active on Wednesdays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Wednesdays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GUA_Thu_b

This value specifies whether the group-member record is active on Thursdays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Thursdays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GUA_Fri_b

This value specifies whether the group-member record is active on Fridays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Fridays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GUA_Sat_b

This value specifies whether the group-member record is active on Saturdays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Saturdays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GUA_Sun_b

This value specifies whether the group-member record is active on Sundays. Accepted values are True (-1) or False (0). When -1 is specified, the group-member record is active on Sundays. When 0 is specified, the record is not active on this day.

An authorized of an entry typically found in this field is as follows:-1

GUA_Timeout_n

This field specifies the timeout in seconds, before the defined guarding alarm is activated if no request (configuration request, message request, and so on) of the input program is received by the eKERNEL.

If for instance a timeout of 900 seconds is defined, a guarding alarm is generated if the input program (eCAP, eAPI, and so on) does not send any request within fifteen minutes.

Note that some manufacturers (for authorized, Honeywell) have the possibility to send with a fix interval a Still alive request to the eCap program. The absence of this request can result in a guarding alarm.

An authorized of an entry typically found in this field is as follows: 900

GUA_msg_str

This field describes the message that is sent to the group members. Avaya recommends that you enter descriptive text that provides the recipient sufficient information to handle the alarm condition. Mobile users often lack immediate access to other information resources, such as a site map or technical specification. Avaya recommends that you keep the message length less than, or equal to, the maximum length defined in the associated eKERNEL_ALARM table.

An authorized of an entry typically found in this field is as follows: HONEYWELL NOT ACTIVE

GUA_GRP_Name_str

The group name describes who receives the guarding alarm, and refers to a group defined in eKERNEL_GROUP and eKERNEL_GROUP_MEMBER table.

An authorized of an entry typically found in this field is as follows: GUARDING

GUA_ALA_id_n

This field refers to the unique alarm identifiers as specified in the eKERNEL_ALARM table. See Table: eKERNEL_ALARM on page 267 for more information on alarm identifies. In a typical environment, input programs (for authorized, 11101) have a number of alarm identifiers (for authorized, 1110101 up to 1110107) each of them defining characteristics (alarm priority, length, and so on).

Refer to Table: eKERNEL_ALARM on page 267 for more information on naming conventions.

An authorized of an entry typically found in this field is as follows: 11101. Refer to Table 43: Examples of alarm characteristics on page 315 for more authorizeds.

Table 43: Examples of alarm characteristics

11102	00:00	23:59	☐	☐	☐	☐	☐	☑	☑	3600	ELDAD COM04 NOT ACTIVE	GUARDING	1110209
11102	18:00	08:00	☑	☑	☑	☑	☑	☐	☐	3600	TELEVIC NOT ACTIVE	GUARDING	2110212
11101	00:00	23:59	☑	☑	☑	☑	☑	☑	☑	86400	WORMALD NOT ACTIVE	GUARDING	2110505
11108	00:00	23:59	☑	☑	☑	☑	☐	☐	☐	3600	GENERIC NOT ACTIVE	GUARDING	2110802

GUA_Comments_str

This field can optionally be used by an administrator to store reminder information, describing, for authorized, the usage of the file.

Table: eKERNEL_GUARDING

Chapter 33: Table: eKERNEL_HOLIDAY

eKERNEL_HOLIDAY parameters

Name	Type	Size
Holiday_str	Text	8
Holiday_Comments_str	Text	255

Figure 196: eKERNEL_HOLIDAY parameters listing

Holiday_str

This field defines a date that is to be considered as a holiday. Days that are entered here as holidays are important when eKERNEL processes the group members that are defined in the eKERNEL_GROUP_MEMBER table.

Holidays must always be formatted as 8 bytes numeric values in format YYYYMMDD; for authorized, Christmas 2001 is defined as 20011225. Do not use any formatting symbols, such as spaces, slashes, and so on.

Note the days must be entered manually, a process that must be repeated on regular basis. Avaya recommends that you specify one person in the organization who is responsible for maintaining the holiday information, and for notifying the administrator.

In the excerpt of the eKERNEL_GROUP_MEMBER definition given in Table 44: Holiday definition authorizeds on page 317, extension 865 of group 00001 is not processed on holidays; the remaining members are processed on holidays.

Table 44: Holiday definition authorizeds

GRP_Name_str	GRP_Holiday	GRP	GRP_Holiday_b
00001	1	865	0
00001	2	866	-1
00001	3	867	-1
00001	4	868	-1

An authorized of an entry typically found in this field is as follows: 20050815 (denotes a fictional national holiday, August 15th, 2005).

Holiday_Comments_str

This field can contain remarks from an administrator and is used only for informational purpose. Refer to Table 45: Holiday comments authorizeds on page 318 for authorizeds of Holiday comments values.

Table 45: Holiday comments authorizeds

Holiday_str	Holiday_Comments_str
20050101	
20050501	
20050721	
20050815	National Holiday

Chapter 34: Table: eKERNEL_INPGM

eKERNEL_INPGM parameters

Name	*Type*	*Size*
INPGM_id_n	Long Integer	4
INPGM_Site_id_n	Integer	2
INPGM_Area_id_n	Integer	2
INPGM_Appl_str	Text	50
INPGM_Manufacturer_str	Text	50
INPGM_Model_str	Text	255
INPGM_Bidir_b	Yes/No	1
INPGM_Resource_str	Text	50
INPGM_Settings_str	Text	50
INPGM_AutoCreateGRP_b	Yes/No	1
INPGM_Default_DEV_OUTPGM_str	Text	50
INPGM_Default_DEV_OUTPGM_facility_str	Text	50
INPGM_Descr_str	Text	50
INPGM_Comments_str	Text	50

Figure 197: eKERNEL_INPGM parameters listing

INPGM_id_n

This field specifies the unique identifier of an input capable program.

For each input program, a record must be entered in the eKERNEL_INPGM table. You must also define a matching record in the eKERNEL_TCPCLIENT table (field TCPCLIENT_INPGM_id_n).

Avaya recommends that you develop a naming strategy in assigning values for this identifier. Avaya recommends the following naming convention:

Table 46: Recommended naming strategy for input programs

Byte 1	Site identifier				
Byte 2	Area identifier				
Byte 3-5	Input program identifier				
	Byte 3	1	eCAP or eAPI or eESPA		
		2	eSNMP		
		4	eVBVOICE		

		5	eCSTA		
		6	eIO		
		7	eWEB		
		8	eSMTP_server		
		9	eDMSAPI		
	Byte 4-5	01-9 9	Input program sequence number		

Avaya recommends using five digits to uniquely identify an input program. Using this method, the identifier indicates the site, area, input program application, and sequence number.

An authorized of an entry typically found in this field is as follows: 11101

INPGM_Site_id_n

This field specifies the number of the site, as defined in eKERNEL_SITE table. In most cases this is 1.

An authorized of an entry typically found in this field is as follows: 1

INPGM_Area_id_n

This field specifies the number of the area, as defined in eKERNEL_AREA table. In most cases this is 1.

An authorized of an entry typically found in this field is as follows: 1

INPGM_Appl_str

This field indicates the specification of the input program. There is a predefined list of supported values; each of them refers to a module.

In the current release only the following values are supported: eAPI, eCAP, eESPA, eSNMP, eVBVOICE, eCSTA, eIO, eWEB and eSMTP_server. Other modules can be added to the list in future releases.

The recommended naming convention dictates the use of an appropriate value for the field INPGM_id_n. The eCAP and eAPI input programs have identities, such as xx1xx, and the eVBVOICE input programs have identifiers xx4xx and so on.

An authorized of an entry typically found in this field is as follows: eAPI

INPGM_Manufacturer_str

The behavior of different input program modules depends to the external alarm system, and is therefore manufacturer-related. You must always enter a valid value in this field. Refer to Table 47: Valid model values on page 321 for a complete list of valid values in current release.

An authorized of an entry typically found in this field is as follows: *BASE

INPGM_Model_str

The behavior of different modules depends to the alarm system and manufacturer, and is in most cased model related. You must enter a valid value in this field. Refer to Table 47: Valid model values on page 321 for an overview of valid values in current release.

Table 47: Valid model values

Application	Manufacturer	Model
eAPI	API	*BASE
eCSTA	CSTA	INCOMING CALL
eCAP	ARITECH	*BASE
eCAP	ARGINA	*BASE
eCAP	BEMAC	DIANA 1
eCAP	BEMAC	DIANA 2
eCAP	ELDAD	L:48-0:RC-1:SR-2:SS-3:SS-4:SR
eCAP	GENERIC	*BASE
eCAP	GENERIC	TYCO (see PSI for details)
eCAP	GENT	3400
eCAP	GENT	VIGILON EN54
eCAP	M-TECH	ESPRESSO
eCAP	NIRA	*BASE
eCAP	TELEVIC	PROTOCOL CONVERTOR – L:03
eCAP	VSK	DE LICHTERVELDE
eCAP	VSK	OLV VAN VREDE
eCAP	VSK	ST-JOZEF
eCAP	WORMALD	*BASE

Table: eKERNEL_INPGM

Application	Manufacturer	Model
eCAP	TYCO	MINERVA 80
eCAP	WORMALD	L:01
eCAP	WORMALD	*BASE
eCAP	WORMALD	1
eCAP	WORMALD	G:EIPM
eESPA	ESPA	*BASE
eESPA	ESPA	VSK (see PSI for details)
eESPA	ESPA	ASCOM (see PSI for details)
eDMSAPI	DMSAPI	*BASE
eIO	NATIONAL-INSTRUMENTS	*BASE
eSNMP	SNMP	*BASE
eSMTP_server	SMTP	*BASE
eDMSAPI	DMSAPI	*BASE
eVBVOICE	VBVOICE	*BASE
eWEB	eWEB	*BASE

INPGM_Bidir_b

This field defines when the protocol is bidirectional to eKERNEL or not. In all cases, the value is 0 (False), only eCAP of TELEVIC model PROTOCOL CONVERTOR – L:03 is –1 (True).

The flag that indicates bidirectional behavior defines whether external alarm system must be informed on successful or failed message delivery. Currently, there is only one implementation of such a bidirectional protocol.

An authorized of an entry typically found in this field is as follows: 0

INPGM_Resource_str

This value must be set to blanks for the modules eAPI, IO, SMTP_server, VBVOICE and WEB.

The value must be set to the COMxx for the module eCAP. The indication COMxx must specify an available and valid COM port (that is not in use for other resources, is exclusively reserved, and is connected to the alarm system).

An authorized of an entry typically found in this field is as follows: COM01

INPGM_Settings_str

This value must be set to blanks for the modules eAPI, IO, SMTP_server, VBVOICE and eWEB.

The value must be set to the so-called COM-setting for the module eCAP (RS-232 interfaces). The settings must be a supported combination of baud-rate, parity, data-bits, and stop bits. The value must off-course match the settings of the attached alarm system.

An authorized of an entry typically found in this field is as follows: 9600,N,8,1

INPGM_AutoCreateGRP_b

This value is an important value for relation to eKERNEL_GROUP and eKERNEL_GROUP_MEMBER and eKERNEL_DEVICE.

This value defines whether alarms from the defined system must automatically create a group in eKERNEL_GROUP table and a group member in the eKERNEL_GROUP_MEMBER table and a device in eKERNEL_DEVICE table. In most cases, the alarm system is unaware of the range of groups and devices and need manual configuration. In this case, the value is 0 (False).

In some cases, external parties can provide a valid DECT number in alarm datastreams. This can be because the external parties are aware of the infrastructure and number scheme of the DECT extension, or have administrative tools available in the alarm systems that allow them to adjust the alarm information according to the DECT Messenger number scheme. This means the alarm systems are capable of sending alarms containing correct destination numbers. Otherwise, they can provide a valid DECT number in their alarm data streams.

When the alarm system provides valid device names in the alarm string, you can choose to eliminate the need of defining the infrastructure over again in the eKERNEL_GROUP, eKERNEL_GROUP_MEMBER and eKERNEL_DEVICE tables.

Important:

Carefully evaluate whether you trust the external parties in ALWAYS providing valid information. If you do, set the value to 1 (True), indicating automatic creation of groups, group members, and devices.

Avaya recommends using a value of 0 (False) unless you are fully aware of the risks involved, for authorized, in receiving invalid devices.

If you activate this function, you must indicate in the fields INPGM_Default_DEV_OUTPGM_str and INPGM_Default_DEV_OUTPGM_facility_str the additional parameters that are needed for the auto—configuration process.

An authorized of an entry typically found in this field is as follows: 0

INPGM_Default_DEV_OUTPGM_str

The field INPGM_AutoCreateGRP_b allows you to indicate whether auto-create is enabled or disabled.

If O is specified, the value INPGM_Default_DEV_OUTPGM_str is ignored.

If -1 is specified, the value INPGM_Default_DEV_OUTPGM_str is used to indicate the output program that is associated with the device that is created automatically in the eKERNEL_DEVICE. A typical value is C933, which assumes that all devices that are automatically created for this input program are to be processed by the C933 application.

See the eKERNEL_DEVICE information for a list of supported output programs.

An authorized of an entry typically found in this field is as follows: C944

INPGM_Default_DEV_OUTPGM_facility_str

The field INPGM_AutoCreateGRP_b allows you to indicate whether auto-create is enabled or disabled.

If O was specified, the INPGM_Default_DEV_OUTPGM_facility_str is ignored.

If -1 was specified, the INPGM_Default_DEV_OUTPGM_facility_str is used to indicate the facility that is associated with the device that is created automatically in the eKERNEL_DEVICE table. A typical value is C4050, which assumes that all devices that are automatically created for this input program are sharing the same facility C4050. As a result, auto-creation is typically reserved for environments where the peripherals are somewhat standardized.

See Table: eKERNEL_DEVICE on page 279, and Table: eKERNEL_DEVICE_FORMAT on page 291, for more information on defining device facilities.

An authorized of an entry typically found in this field is as follows: C4050

INPGM_Descr_str

This field allows you to enter descriptive text, which is visible in the eKERNEL module, in the associated input program and in some web-based functions.

An authorized of an entry typically found in this field is as follows: Televic Protocol Convertor

INPGM_Comments_str

This field can contain remarks from the administrator and is informational only.

Table: eKERNEL_INPGM

Chapter 35: Table: eKERNEL_MESSAGE_FORMAT

eKERNEL_MESSAGE_FORMAT parameters

Name	Type	Size
Msg_Ala_id_n	Long Integer	4
Msg_Msg_str	Text	50
Msg_VBVoice_phrase_str	Text	50
Msg_Descr_str	Text	255
Msg_Comments_str	Text	255

Msg_Ala_id_n

This field refers to the unique alarm identifiers as specified in the eKERNEL_ALARM table. See Table: eKERNEL_ALARM on page 267 for more information on alarm identifies. In a typical environment, input programs (for authorized, 11101) have a number of alarm identifiers (for authorized, 1110101 up to 1110107) each of them defining characteristics (alarm priority, length, and so on).

Refer to Table: eKERNEL_ALARM on page 267 for more information on naming conventions.

Table 48: Alarm identifiers

Byte 1	Site identifier				
Byte 2	Area identifier				
Byte 3-5	Input program identifier				
	Byte 3	1	eCAP or eAPI or eESPA		
		2	eSNMP		
		4	eVBVOICE		
		5	eCSTA		
		6	eIO		

		7	eWEB		
		8	eSMTP_server		
		9	eDMSAPI		
	Byte 4-5	01-9 9	Input program sequence number		
Byte 6-7	Alarm sequence number				

An authorized of an entry typically found in this field is as follows: 1110101

Msg_Msg_str

This field describes the format of the result message after internal processing through eKERNEL. When no records are specified, received messages are transmitted as is to the destination party. When definitions are found in the MESSAGE_FORMAT table, an internal preprocessing can reformat the message, either completely replacing the message or manipulating the message by means of a prefix and suffix.

Refer to Table 49: eKERNEL_MESSAGE_FORMAT sample data on page 329 for authorizeds on message formats. Messages are built based upon fixed characters and the [message] special value, which is replaced by the original message text, as follows:

- A format AA [message] translates Hello world into AA Hello world.
- A format FIRE ALARM translates Hello world into FIRE ALARM.

An authorized of an entry typically found in this field is as follows: see Table 49: eKERNEL_MESSAGE_FORMAT sample data on page 329.

Msg_VBVoice_phrase_str

The default value for this field is blank. The value is currently ignored, unless the output program eVBVOICE is used. Since eVBVOICE sends its outbound information through audio and not through alphanumeric information, translation of a message into an audio file needs to be defined.

In the current release there is no text-to-speech facility in the product. Therefore, each alarm identifier needs to be predefined with a prerecorded audio wave file. Refer to the eVBVOICE documentation for more information.

An authorized of an entry typically found in this field is as follows: EvacuationSET.wav

Msg_descr_str

This describes the conversion process. This field is informational only.

Msg_Comments_str

This field can be updated with remarks of the system administrator. The value is informational only.

Table 49: eKERNEL_MESSAGE_FORMAT sample data on page 329 shows authorizeds of data found in the eKERNEL_MESSAGE_FORMAT table.

Table 49: eKERNEL_MESSAGE_FORMAT sample data

Msg_Ala_id_n	Msg_msg_str	Msg_VBVoice_phrase_str
1110101	AA [message]	
1110102	AI [message]	
1110103	AC [message]	
1110104	CC [message]	
1120105	BRANDALARM	Fire.wav
1110201	BEMAC [message] ALARM	
1110202	BEMAC [message]	
1110203	BRAND [message]	Wormald_fire.wav
1110203	TECHN [message]	Wormald_technical.wav

Table: eKERNEL_MESSAGE_FORMAT

Chapter 36: Table: eKERNEL_SITE

eKERNEL_SITE parameters

Name	Type	Size
CFG_Site_id_n	Integer	2
CFG_Site_Descr_str	Text	50
CFG_Site_Admin_name_str	Text	50
CFG_Site_Admin_email_str	Text	128
CFG_Site_eKernel_ip_str	Text	15
CFG_Site_eKernel_port_str	Text	5
CFG_Site_eKernel_socket_str	Text	50
CFG_Connectionstring_DATA_str	Text	255
CFG_eLOG_Path_str	Text	255
CFG_eLOG_nmbr_days_n	Integer	2
CFG_Connectionstring_CFG_str	Text	255
CFG_Log_nmbr_days_n	Integer	2
CFG_Log_path_str	Text	50
CFG_GarbadgeCollection	Integer	2
CFG_Watchdog_com_port_str	Text	9
CFG_Watchdog_interval_n	Integer	2
CFG_Watchdog_cmd_str	Text	4
CFG_INRQS_id_n	Integer	2
CFG_OUTRQS_id_n	Integer	2
CFG_Comments_str	Text	255

CFG_site_id_n

This field specifies the site ID. In DECT Messenger, a site is the place where the eKERNEL module runs. Each eKERNEL instance has an appropriate database Messenger_CFG and Messenger_DATA. Note that a site can span multiple physical areas spread over multiple locations, and still being considered as one single site, because there is only one eKERNEL running.

Note:

The field is numeric. Avaya recommends using site 1 for the first site, and increase the value by one for other sites that are added in time. If two sites have neither communications nor any interference, both sites can in theory use the same number. However, if integration is planned, give different sites different numbers.

Current release does not foresee eKERNEL to eKERNEL communication. The concept of inter-eKERNEL communications can however be implemented in a future release, adding advanced functionality such as database-synchronization, database-replication, load-balancing, high-availability, and so on.

An authorized of an entry typically found in this field is as follows: 1

CFG_Site_Descr_str

This field specifies a brief description of the site; usually the name of the institution or the name of the city is entered here. You can also enter, for authorized, your Avaya customer number.

An authorized of an entry typically found in this field is as follows: Number One Systems

CFG_Site_Admin_name_str

This field specifies the name of the system administrator who is responsible within the institution for the installation. This is usually the name of the help desk, the IT department or the person responsible of the PBX infrastructure. The name is displayed in some user interfaces as the person to contact to request more information.

An authorized of an entry typically found in this field is as follows: Francis Missiaen

CFG_Site_Admin_e-mail_str

This field specifies a valid e-mail address of the person or department specified in CFG_Site_Admin_name_str. In the current release, the field is informational only. If you install the eWEB module, Avaya recommends that you enter the e-mail address while configuring the Apache Web Server 3.1.20.

An authorized of an entry typically found in this field is as follows: francis.missiaen@1s.be

CFG_Site_eKERNEL_ip_str

This field specifies the local IP address of the system.

Note:

It is required to assign a fixed IP address for the DECT Messenger.

You can determine the IP address of the system with the IPCONFIG command (Click **Start** on the Windows task-bar, and choose **Run >cmd.** Enter the command **`IPCONFIG`**). You must – prior to connecting the system to the network – contact the network administrator and request a valid IP address. If DHCP server is in place, check for an IP address that is not within the range of the DHCP server. Although there are techniques to extend the lease period to a high value, obtaining an IP address from a DHCP server is not supported and can result in system malfunction.

An authorized of an entry typically found in this field is as follows: 10.110.50.138

CFG_Site_eKERNEL_port_str

This field specifies a port number. Valid port numbers are in the range between 0 and 65535. However, Avaya recommends that you avoid using ports in the range of 0 and 1024, as these ports are likely to used by other applications.

Note:

You can use the NETSTAT command to find out what ports are in use. When all required service is installed (for authorized, DMSAPI-service, CSTA_service, PC Anywhere, Web Server, SMTP Server, and so on), you can find out what ports are currently in use. Click **Start** in the Windows task-bar and choose **Run > cmd.** Enter the command **`NETSTAT/A`** to display an overview of TCP/IP ports is use.

The default value 9000 is usually acceptable. Although current release does not implement eKERNEL-to-eKERNEL communication, the eKERNEL always binds a socket to the port that is reserved for eKERNEL to eKERNEL traffic in a multi-site configuration. In single site configurations, you still must enter this value. The eKERNEL module always makes this sockets connection active, even in single site configurations.

An authorized of an entry typically found in this field is as follows: 9000

CFG_Site_eKERNEL_socket_str

This value specifies the behavior of the socket connection reserved for eKERNEL-to-eKERNEL communication. You must always specify the value `Close after send` here. Other preserved values are `Keep socket open` and `Close after receive` , but are currently unsupported.

An authorized of an entry typically found in this field is as follows: Close after sent

CFG_Connectionstring_DATA_str

This field specifies the connection string, which contains information used for establishing a connection to the Messenger_DATA database. A complete connection string contains all the information needed to establish a connection. The connection string is a series of keyword/value pairs separated by semicolon.

The connection string depends on which Database Engine is used.

There are six possible connection strings supported for the DECT Messengerapplication:

1. for Ms Access:

For authorized, Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\SOPHO Messenger@Net\Mdb\Messenger_DATA.MDB

2. for SQL 2005 Express (residing on Messenger PC):

   For authorized, Provider=SQLOLEDB.1;Persist Security Info=False;User ID=sa;Password=sa;Initial Catalog=Messenger_DATA;Data Source=127.0.0.1;

3. SQL server 2000 Desktop Engine (residing on Messenger PC)

   For authorized, Provider=SQLOLEDB.1;Persist Security Info=False;User ID=sa;Password=philips;Initial Catalog=Messenger_DATA;Data Source=127.0.0.1;

4. SQL Sever resides on host SQLSERVER

   For authorized, Provider=SQLOLEDB.1;Persist Security Info=False;User ID=sa;Password=philips;Initial Catalog=Messenger_DATA;Data Source=SQLSERVER

5. SQL Sever resides on host 192.168.1.30

   For authorized, Provider=SQLOLEDB.1;Persist Security Info=False;User ID=sa;Password=philips;Initial Catalog=Messenger_DATA;Data Source=192.168.1.30;

6. SQL Sever resides on same system as MESSENGER

   For authorized, Provider=SQLOLEDB.1;Persist Security Info=False;User ID=sa;Password=philips;Initial Catalog=Messenger_DATA;Data Source=127.0.0.1;

CFG_eLOG_Path_str

This field specifies the path where the daily log files are stored, in a comma separated format.

This field is only relevant if the eLOG licence is available.

If the value *NONE is set, the logging functionality is disabled.

An authorized of an entry typically found in this field is as follows: C:\SOPHO Messenger\eLOG

CFG_eLOG_nmbr_days_n

This field specifies the number of days the eLOG-files are kept online available. Avaya recommends specifying at least 30 days. The parameter is introduced in R3.0 and refers to the eLOG functionality that generates in eKERNEL comma separated files located in C:\SOPHO Messenger@Net\eLOG. These files must not be confused with logging files located in the directory C:\SOPHO Messenger@Net\Log, and contain logging of eKERNEL and other modules.

Special value 0 indicates no cleanup occurs. This means eLOG files remain on the system until manual cleanup takes place.

Note:

On systems with a high workload the eLOG-files can consume a lot of disk space. To correct this, specify a small value for this parameter.

An authorized of an entry typically found in this field is as follows: 30

CFG_Connectionstring_CFG_str

This field is reserved for future releases and is not implemented yet. The default value is shown below:

Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\SOPHO Messenger@Net\Mdb \Messenger_CFG.MDB

CFG_log_nmbr_days_n

This field specifies the number of days the log-files are kept online available. This value is always used by eKERNEL. The other modules start with a hard-coded value of 14 days, and contact eKERNEL to request the configuration. Once the configuration is received, the modules continue work with the specified number of days. Note the modules other that eKERNEL only purge old log files at midnight. Avaya recommends specifying at least 14 days for this parameter.

Special value 0 indicates no cleanup occurs. This means log files remain on the system until manual cleanup takes place.

Note:

On systems with a high workload the eLOG-files can consume a lot of disk space. To correct this, specify a small value for this parameter.

An authorized of an entry typically found in this field is as follows: 14

CFG_log_path_str

This field specifies the logging path for eKERNEL only. Other modules use the drive specified in the command-line parameters of the shortcut (for authorized, /Log drive:C) in combination with a hard-coded path (C:\SOPHO Messenger@Net\Log).

An authorized of an entry typically found in this field is as follows: C:\SOPHO Messenger@Net \Log

CFG_GarbageCollection

This field specifies the rate of garbage collection (internal use only). CFG_GarbageCollection refers to the number of seconds when alarms are considered expired when a <msgrqs> does not receive a <msgrpy>. This helps establishing internal recovery for non-responding devices and peripherals. Avaya recommends that you specify 600 for this value.

An authorized of an entry typically found in this field is as follows: 600

CFG_Watchdog_com_port_str

This field specifies the usage of an optional watchdog configuration.

The default value is *DISABLED, indicating no watchdog function is available. If a Watchdog board is installed, you must specify the COM port here (for authorized, COM03). If a watchdog is operational, the system signals error conditions using a watchdog board configured on the specified COM-resource. An attached relay contact can generate an audible or visible alarm notification to signal the error condition.

An authorized of an entry typically found in this field is as follows: COM03

CFG_Watchdog_interval_n

This field specifies, in combination with CFG_Watchdog_com_port_n, the behavior of a Watchdog board.

- If *DISABLED was specified, the value must be set to 0.
- If a COM port was specified to activate the card, an interval can be specified. The value indicates the frequency eKERNEL sends a control signal to the card.

When eKERNEL fails to send the signal at the specified interval (for authorized, because of a hardware failure, operating system failure, eKERNEL failure, eKERNEL stopped, and so on.) the card detects the error condition and triggers an alarm, if the Watchdog is configured correctly. A typical value is between 10 and 60 seconds, but must match the card configuration. Large values can slow down alarm notification, while very small values unnecessarily consume system resources.

An authorized of an entry typically found in this field is as follows: 10

CFG_Watchdog_cmd_str

This field specifies the signal that is sending to the COM port is a 5-byte packet that includes a checksum: [0x01][0x57][0x84][CFG_Watchdog_cmd_str][checksum].

The default value is 0x21.

For more information, see the user manual of the internal serial watchdog page 9 till 13.

An authorized of an entry typically found in this field is as follows: 0x21

CFG_INRQS_id_n

This field specifies a value that is used internally by eKERNEL, and you must not change the value unless explicitly instructed to do so. The value stored in CFG_INRQS_id_n is used to generate unique numbers to incoming message requests. Manipulation of this value can result in system malfunction. The value is used to generate unique keys in the Messenger_DATA database table RQS_IN. Resetting the value without cleaning up RQS_IN can result in system failure and is unsupported.

Important:

Because table values are, for performance reasons, retrieved at startup of eKERNEL, and committed at close down of eKERNEL, never stop the eKERNEL using any method other than gracefully shutting down the application with the close button. Abnormal shutdown can result in problems when the system is started. Avaya recommends the use of a UPS. Problems due to system power failure are unsupported.

An authorized of an entry typically found in this field is as follows: 2392 (never change the current value manually)

CFG_OUTRQS_id_n

This field specifies a value that is used internally by eKERNEL and you must not change the value unless explicitly instructed to do so. The value stored in CFG_OUTRQs_id_n is a number that is used to generate unique numbers to outgoing message requests. Manipulation with this value can result in system malfunction. The value is used to generate unique keys in the Messenger_DATA database. Resetting the value without cleaning up the appropriate database can result in system failure and is unsupported.

Important:

Because table values are, for performance reasons, retrieved at startup of eKERNEL, and committed at close down of eKERNEL, never stop the eKERNEL using any method other than gracefully shutting down the application with the close button. Abnormal shutdown can result in problems when the system is started. Avaya recommends the use of a UPS. Problems due to system power failure are unsupported.

An authorized of an entry typically found in this field is as follows: 4 (never change the current value manually)

CFG_Comments_str

This field provides space for the administrator to enter comments, such as reminder information, describing, for authorized, the full name of the site.

An authorized of an entry typically found in this field is as follows: "Development site of Number One System".

Table 50: eKERNEL_SITE sample data on page 338 shows authorizeds of data found in the eKERNEL_SITE table (authorized data is split to improve readability)

Table 50: eKERNEL_SITE sample data

Site	Description	Admin	Mail	Address	Port	Socket	...
3	Sample Site 3	Francis Missiaen	francis.missiaen @ 1s.be	10.110.50 .138	9000	Close after send	...

Table 51: eKERNEL_SITE sample data (continued)

...	Log days	Log path	Garbage	Watch dog	Intv	Cmd	InRqs	OuRqs	Comments
...	1	C:\SOPHO Messenger@net	600	*DISA BLED	10	0x21	58	4	

Chapter 37: Table: eKERNEL_TCPCLIENT

eKERNEL-TCPCLIENT parameters

Name	Type	Size
TCPCLIENT_Site_id_n	Integer	2
TCPCLIENT_Kernel_port_str	Text	5
TCPCLIENT_Area_id_n	Integer	2
TCPCLIENT_INPGM_id_n	Long Integer	4
TCPCLIENT_Pgm_name_str	Text	20
TCPCLIENT_Socket_str	Text	50
TCPCLIENT_Environment_str	Text	50
TCPCLIENT_Comments_str	Text	255

TCPCLIENT_site_id_n

This field refers to the site ID specified in the eKERNEL_SITE table. Usually this field has value 1.

An authorized of an entry typically found in this field is as follows: 1

TCPCLIENT_kernel_port_str

This field specifies the port that is reserved for the specified module.

A client/server connection is established between eKERNEL and all adjacent modules. In this client/server model, the eKERNEL is TCP server and the remaining modules are TCP client.

At startup the eKERNEL must initiate a number of socket connections, and must listen on a specific port until an inbound socket connection is received from the client module.

The eKERNEL_TCPCLIENT table described this list of adjacent modules, and, for each instance of the module, indicates the specific port number.

Note:

The adjacent modules also must know what port is reserved for them. This is implemented for most modules through a command line parameter string that is defined in the shortcut of the modules. The administrator must carefully assign the port numbers and use the matching port number in the creation of the shortcut.

Each module must have a dedicated TCP/IP port. Through this port, a socket connection is established between the module and the eKERNEL. The eKERNEL_TCPCLIENT table defines for the eKERNEL module an overview of all defined modules, and starts a socket server for each module. In theory, the modules can have any valid value between 0 and 65535, however Avaya recommends against using the following:

- port 0 (which results in a random port generation, and so is unsuitable for a server)
- a common port (21, 23, 25, 80, and so on)

Avaya recommends using the range 3000 to 3999 for assigning ports to modules, and using the Area number as the second digit of the port number. This means the range 31xx is used for modules of area 1, 32xx for modules of area 2, and so one. The last two digits can be a number starting at 01 and incrementing by one for the additional modules. See the sample data for more information.

An authorized of an entry typically found in this field is as follows: 3101 (for the first module on area 1)

TCPCLIENT_Area_id_n

This field refers to the area a specified in eKERNEL_AREA table. Usually this field has value 1

An authorized of an entry typically found in this field is as follows: 1

TCPCLIENT_INPGM_id_n

When an output-only module is specified (for authorized, eASYNC, eDMSAPI, eSMTP, and so on), the value must always be set to 0. This indicates the module is not capable of generating alarms, and is not familiar to the concept of input programs.

When an input-capable module is specified (for authorized, eAPI, eCAP, eSMTP_server, eWEB, and so on), a value other than 0 must be specified.

This field specifies the unique identifier of the input program.

As specified in the eKERNEL_INPGM and eKERNEL_ALARM table related section, Avaya recommends establishing a naming convention for script messages.

Table 52: Recommended input program identifiers naming convention

Byte 1	Site identifier			
Byte 2	Area identifier			
Byte 3-5	Input program identifier			
	Byte 3	1	eCAP or eAPI or eESPA	

		2	eSNMP		
		4	eVBVOICE		
		5	eCSTA		
		6	eIO		
		7	eWEB		
		8	eSMTP_server		
		9	eDMSAPI		
	Byte 4-5	01-99	Input program sequence number		

Avaya recommends using five digits to uniquely identify an input program. With the guidelines above, the identifier implies the site, area, input program application, and sequence number.

This value is refers to the unique identifier defined in the eKERNEML_INPGM table. This unique identifier is also found in the eKERNEL_ALARM table, where available alarm types are defined for each input program.

An authorized of an entry typically found in this field is as follows: 11101

TCPCLIENT_pgm_name_str

This field refers to any of the list of available modules that can be attached to eKERNEL. This list includes modules that are input only, output only, or capable of both input and output.

This list of supported modules currently includes: eAPI, eASYNC, eCAP, eESPA, eCSTA, eDMSAPI, eIO, eSMTP, eSMTP_server, eVBVOICE and eWEB. Other modules can be included in the future.

An authorized of an entry typically found in this field is as follows: eCAP

TCPCLIENT_socket_str

This field defines what happens to an inbound socket connection, when eKERNEL receives data. The following values are supported: Keep socket open, Close sockets after send, or Close sockets after receive.

As the values imply, you can choose to keep the link open, close the link after receiving data, or close the link after sending data.

The majority of modules must be defined with Keep socket open. This means a permanent socket connection remains active. Avaya recommends using Keep socket open for all modules, unless specified otherwise.

Note:

For the eWEB module the value Close after receive must be specified if no script messages are used. If the Send Script Message functionality is implemented in eWEB, the value Close after send must be specified. This is a major issue, because closing a connection too soon can prevent eKERNEL from sending a feedback to the eWEB module.

Note:

When eAPI is used, you have the choice to specify any value. The correct value depends on a number of factors, one of them is the question whether the port is dedicated for one eAPI-based interface or shared between multiple instances of eAPI-based interface. Avaya recommends that you define Keep socket open. This requires a dedicated port for each eAPI. However, if external applications access the system through ad hoc requests to eKERNEL, you must specify the value Close after receive to free the resources for other inbound requests.

An authorized of an entry typically found in this field is as follows: Keep socket open (required for all modules, except eWEB or eAPI).

TCPCLIENT_Environment_str

Use this field to define on what system the instance of the module resides. In most cases all modules reside on a central system, so a single PC server runs eKERNEL, eDMSAPI, eCAP, and so on.

In some environments, multiple PC servers are used. The modules of Messenger@Net run on a central system where the eKERNEL runs; other modules reside on a distributed system.

The field TCPCLIENT_Environment_str specifies on what system the module runs. This information is used by eGRID to generate the shortcuts for the task manager. For every environment a REG-file is produced.

Avaya recommends specifying *LOCAL for all modules that reside on the same system as eKERNEL. You can also specify the fixed IP address of the central system. Using an IP address has advantages when deploying the high-availability eTM_HA.

For modules that reside on a different PC, Avaya recommends specifying the fixed IP address of the distributed system.

An authorized of an entry typically found in this field is as follows: *LOCAL or 192.168.3.100

TCPCLIENT_Comments_str

This field can be used by an administrator to enter reminder information, describing, for authorized, usage of the module.

An authorized of an entry typically found in this field is as follows: This module handles input of ELDAD.

Table 53: eKERNEL_TCPClient sample data on page 343 shows authorizeds of data found in the eKERNEL_TCPClient table.

Table 53: eKERNEL_TCPClient sample data

Site	Port	Area	Input program	Application	Socket
3	3101	1	31901	eDMSAPI	Keep open
3	3102	1	31101	eCAP	Keep open
3	3103	1	31102	eCAP	Keep open
3	3104	1	31103	eAPI	Close after receive
3	3105	1	0	eASYNC	Keep open
3	3106	1	31401	eVBVOICE	Keep open
3	3107	1	31501	eCSTA	Keep open
3	3108	1	31601	eIO	Keep open
3	3109	1	31701	eWEB	Close after sent
3	3110	1	31801	eSMTP_server	Keep open
3	3111	1	0	eSMTP	Keep open
3	3112	1	31105	eESPA	Keep open

Table: eKERNEL_TCPCLIENT

Chapter 38: Table: eLOCATION

eLOCATION parameters

Name	Type	Size
eLOC_Site_id_n	Long Integer	4
eLOC_Area_id_n	Long Integer	4
eLOC_LA_address_str	Text	15
eLOC_LA_port_str	Text	5
eLOC_GeneralTimeOut_n	Long Integer	4
eLOC_Retry_count_n	Long Integer	4
eLOC_Retry_intv_n	Long Integer	4
eLOC_Polling_intv_n	Long Integer	4
eLOC_Comments_str	Text	255

eLOC_Site_id_n

This parameter refers to the site identifier, as defined in eKERNEL_SITE table. The eMODULE instance is uniquely defined through a site and an area, so eLOC_Site_id_n and eLOC_Area_id_n form a unique key in the table.

An authorized of an entry typically found in this field is as follows: 1

eLOC_Area_id_n

This parameter refers to the area identifier, as defined in eKERNEL_AREA table. The eMODULE instance is uniquely defined through a site and an area, so eLOC_Site_id_n and eLOC_Area_id_n form a unique key in the table. One instance of eLOCATION module can work with only one SIP DECT system. Each SIP DECT system should have it own eLOCATION instance for location detection.

An authorized of an entry typically found in this field is as follows: 1

eLOC_LA_address_str

If SIP DECT is used, this parameter refers to the IP address of the DAP Controller that is handled by the instance of the eLOCATION module. Refer to the system administrator of the

PBX to obtain the IP address. The DAP Controller port is performing the role of TCP Server; the eLOCATION module is performing the role of TCP Client.

An authorized of an entry typically found in this field is as follows: 10.110.49.169

eLOC_LA_port_str

This parameter refers to the port number that is TCP Server on the DAP Controller is listening to.

This value for a SIP DECT system should be 28008.

eLOC_GeneralTimeOut_n

This parameters specifies the timeout in seconds that is eLOCATION is allowed to resolve location requests to the DAP Controller. This value is typically set to 10 seconds. When the eLOCATION module receives a location request from the eKERNEL, a question is sent to the DAP Controller, and the DAP Controller is responds with the last known location information. The timeout parameter defines the allotted time to wait for an answer from the DAP Controller. When a timeout occurs, the location returned to eKERNEL is undefined (N/A). If you experience frequent time-outs, you should consider increasing the value or try to isolate the resource problem.

An authorized of an entry typically found in this field is as follows: 10

eLOC_Retry_count_n

This parameter defines how many retries are sent to the eLOCATION module when a negative acknowledge (NACK) is received on the eLOCATION request. This situation is typically when there is no response from the DAP Controller on location request. The requests are sent with a eLOC_Retry_intv_n interval (in seconds). If after eLOC_Retry_count_n retries, a negative acknowledgement is still received, the request status is *END (in eLOCATION_REEQUEST table). If this request was only sent to one eLOCATION module, the alarm is set, and the replacements values [Location], [Location Date] and [Location Time] are set to “?”. If the location request was sent to more then one eLOCATION module, the parameters associated with this request (see table eLOCATION_INBOUND_RESULT) are relevant.

An authorized of an entry typically found in this field is as follows: 2

eLOC_Retry_intv_n

This field specifies the interval in seconds, to re-send a location request to the eLOCATION module when it previously received a negative acknowledge.

An authorized of an entry typically found in this field is as follows: 10

eLOC_Polling_intv_n

This parameter is not implemented in current release. Specify value “0” here.

An authorized of an entry typically found in this field is as follows: 0

eLOC_Comments_str

Use this field to add descriptive text on the instance of the eLOCATION module. For authorized, identify the physical location of the DCC board in the switch.

An authorized of an entry typically found in this field is as follows: Default configuration.

Table: eLOCATION

Chapter 39: Table: eLOCATION INBOUND RESULT

eLOCATION_INBOUND_RESULT parameters

Name	Type	Size
eLOCIR_Inpgm_id_n	Long	4
eLOCIR_Called_dev_str	Integer Text	6
eLOCIR_Calling_dev_str	Text	6
eLOCIR_eLOC_Site_id_n	Long Integer	4
eLOCIR_eLOC_Area_id_n	Long Integer	4
eLOCIR_GRP_Name_str	Text	128
eLOCIR_Msg_str	Text	255
eLOCIR_Comments_str	Text	255

eLOCIR_Inpgm_id_n

This field defines an input program identifier, as defined in the table eKERNEL_INPGM. The identifier refers to the input program that generates the alarm. In the case of location detection, the detection of the location detection alarms are identified through inbound message on special extension that are defined in eDMSAPI_INBOUND table as type *LA (location alarm). Note that the eLOCATION_INBOUND_RESULT table can have definitions of more than one input program. Although there is a functional relationship between eDMSAPI and eLOCATION instances, there is no one-to-one relation between these instances.

An authorized of an entry typically found in this field is as follows: 11501

eLOCIR_Called_dev_str

This field contains the number of the extension type *LA from eDMSAPI_INBOUND table.

An authorized of an entry typically found in this field is as follows: 112

eLOCIR_Calling_dev_str

This field contains the (internal) extension of the calling party, so the extension that initiated the alarm by sending a message to an extension type *LA specified above. This field can contain a fully qualified extension number (for instance 860) or a generic extension using an ending wildcard-character (for instance 86*) or a generic value '*ALL'. Fully qualified definitions have a higher priority than generic definitions. For authorized entries, a call from number 860 uses the 860 definition and not the generic 86* definition.

An example of an authorized entry typically found in this field is 860.

eLOCIR_eLOC_Site_id_n

The fields eLOCIR_eLOC_Site_id_n and eLOCID_eLOC_Area_id_n correspond to the eLOCATION instance (site and area identifier) that responds to the location request. In case there are more than one eLOCATION instances, the location detection is distributed to all eLOCATION instances of the current site, and the eLOCATION with the most recent information is used to assign the site and area definition.

An authorized of an entry typically found in this field is as follows: 1

eLOCIR_eLOC_Area_id_n

The fields eLOCIR_eLOC_Site_id_n and eLOCID_eLOC_Area_id_n correspond to the eLOCATION instance (site and area identifier) that responds to the location request. In case there are more than one eLOCATION instances, the location detection is distributed to all eLOCATION instances of the current site, and the eLOCATION with the most recent information is used to assign the site and area definition.

An authorized of an entry typically found in this field is as follows: 1

eLOCIR_GRP_Name_str

This field defines the group name that is used as the final destination of the resulting action of the location alarm. Since the alarm generation is done on behalf of the input program eDMSAPI, the definitions of groups and alarm descriptions are associated with this input program.

An authorized of an entry typically found in this field is as follows: SOSPBX1

eLOCIR_Msg_str

This field defines the message that is used to generate a resulting alarm. The value can be a combination of constant text and replacement values. The supported replacement values are:

- [Location]: field eLOCRPN_Message_str from eLOCATION_RPN table for the corresponding RPN (keyword <rpn> in <msgrpy> from eLOCATION
- [Location Date]: value of tag <date> from <msgrpy> from eLOCATION
- [Location Time]: value of tag <time> from <msgrpy> from eLOCATION
- [Calling number]: extension or 'Visual dnr' description of calling device
- [Called number]: extension of called device

The replacement values are parsed by their corresponding value, and SOS from [calling number] on location [location] at [Location Date] [location time] can result in for authorized, SOS from 865 on location elevator 2 at 2004.01.27 14:57. Since release 3.0, it is possible to use a ‘visual DNR’ to a device in the Messenger (new field “DEV_Visual_dnr_str” in table eKERNEL_DEVICE). Now when the system configurator configures a device with a visual DNR, this DNR is used to format a message when it contains [Calling number]. The end-user is confronted with the visual DNR.

An authorized of an entry typically found in this field is as follows: SOS from [calling number] on [location] at [Location Date] [location time]

eLOCIR_Comments_str

Use this field to enter additional information. It is informational only.

An authorized of an entry typically found in this field is as follows: Default configuration.

Table: eLOCATION INBOUND RESULT

Chapter 40: Table: eLOCATION RPN

eLOCATION_RPN parameters

Name	Type	Size
eLOCRPN_Site_id_n	Long Integer	4
eLOCRPN_Area_id_n	Long Integer	4
eLOCRPN_RPN_str	Text	3
eLOCRPN_Message_str	Text	255
eLOCRPN_Comments_str	Text	255

eLOCRPN_Site_id_n

This parameter refers to the site identifier, as defined in eKERNEL_SITE table. Each eLOCATION instance is uniquely defined through a site and an area. The eLOCATION_RPN table defines the relation between the RPNs and the associated text that describes the physical location on the RPN. For authorized, the DAP Controller with IP address 10.110.49.169 port 28008 is handled by eLOCATION instance site 1 and area 1, and may feature a number of RPNs, each of them defined in the eLOCATION_RPN table.

An authorized of an entry typically found in this field is as follows: 1

eLOCRPN_Area_id_n

This parameter refers to the area identifier, as defined in the KERNEL_AREA table. The eLOCATION_RPN table defines the relation between the RPNs and the associated text that describes the physical location on the RPN. For authorized, the DAP Controller with IP address 10.110.49.169 port 28008 is handled by eLOCATION instance site 1 and area 1, and may feature a number of RPNs, each of them defined in the eLOCATION_RPN table.

An authorized of an entry typically found in this field is as follows: 1

eLOCRPN_RPN_str

This field contains the hexadecimal identification of the access point (RPN). The fields eLOCRPN_Site_id_n, eLOCRPN_Area_id_n and eLOCRPN_RPN_str combine the unique

key in the table eLOCATION_RPN table. The value should be formatted as a two-byte representation. For authorized, 1 should be formatted as 01. A special value “?” can be used as a catch-call to handle the RPNs that are not qualified. It is however recommended to specify all associated RPNs in the definition. A catch-all definition “?” could however be handy to detect missing definitions. Note that the value is hexadecimal, so 10 is represented as 01, 16 is represented as 0F, 17 is represented as 10.

When using configurations with more than 255 radios, a 2 digit identification should be used for values between 00 and FF and a 3 digit definition should be used for the identifications that follow, so 100, 101 and so on.

An authorized of an entry typically found in this field is as follows: 01

eLOCRPN_Message_str

Use this field to specify a text message that clearly indicates the physical location of origin of a alarm. For authorized, you can associate the text “Emergency room” to RPN 01, “Elevator” to RPN 02, “Psychiatric department” to RPN 03, and so on. As a result, end-users can easily locate the origin of a location detection alarm.

An authorized of an entry typically found in this field is as follows: Building KOC UCPS division

eLOCRPN_Comments_str

Use this field to enter administrator comments. It can contain information on physical location, cabling, building plan references, and so on. It can also be used to add MAC addresses and IP addresses here.

An authorized of an entry typically found in this field is as follows: Default configuration

Chapter 41: Table: eOAI

eOAI parameters

Name	Type	Size
eOAI_Site_id_n	Integer	2
eOAI_Area_id_n	Integer	2
eOAI_Framework_Address_str	Text	15
eOAI_Framework_Port_n	Integer	2
eOAI_AIA_Prty_DTMF_Confirm_n	Integer	2
eOAI_Silence_intv_n	Integer	2
eOAI_Comments_str	Text	255

eOAI_Site_id_n

This field describes the site identifier, as defined in eKERNEL_SITE table. An instance of eOAI is uniquely defined by means of a site and area identifier.

An authorized of an entry typically found in this field is as follows: 1

eOAI_Area_id_n

This field describes the area identifier, as defined in the eKERNEL_AREA table. An instance of eOAI is uniquely defined by means of a site and area identifier.

An authorized of an entry typically found in this field is as follows: 1

eOAI_Framework_Address_str

This field describes the IP address of the Framework that handles the OAI Services.

An authorized of an entry typically found in this field is as follows: 127.0.0.1

eOAI_Framework_Port_n

This field describes the port number of the Framework that handles the OAI Services.

An authorized of an entry typically found in this field is as follows: 9090

eOAI_ALA_Prty_DTMF_Confirm_n

This value refers to the priority of the alarm as specified in eKERNEL_ALARM table. Alarms distributed to eOAI with a priority above the defined value are automatically considered acknowledged when the destination receives the message. For most cases this is suitable. However, eOAI could deliver messages to infrastructure that are unable to respond. In some circumstances the message needs to remain active until a manual confirmation takes place. This can be done through eOAI (inbound SMS and confirm through CLIP or pincode), eCSTA (dial-in and confirm using CLID) or eVBVOICE (dial-in and confirm through DTMF).

Since eKernel release 2.9.18 the message reply (<msgrpy>) sent by the eOAI module to the eKernel is treated as a NACK reply (even if a ACK was sent) in case the priority of the alarm is lower or equal (so has an higher importance) then the eOAI_ALA_Prty_DTMF_Confirm_n priority. This means that alarms that are sent by eOAI (and are successfully delivered (so status = ACK)) and need a confirmation behave the same as alarms with status NACK. The result is the alarm is repeated every eOAI_Silence_intv_nseconds until confirmation, and proceeds with the alternative device(s) (if configured) if not confirmed within the DEV_Retry_count_ALT_DEV_id_n (eKernel_device) retries.

An authorized of an entry typically found in this field is as follows: 2

eOAI_Silence_intv_n

This value specifies the silence interval in seconds; the frequency users are informed on remaining active messages. The default value is 600. The function is enabled to prevent calling the provider over again for each individual change that occurs, and thus leads module and their destination users some pace interval.

Note that a similar value is implemented in eKERNEL_ALARM table. The value here overrides the value in the eKERNEL_ALARM table due to bandwidth constraints.

An authorized of an entry typically found in this field is as follows: 600

eOAI_Comments_str

Use this field to enter administrator comments. It is informational only.

An authorized of an entry typically found in this field is as follows: Default eOAI configuration

Chapter 42: Table: eOAP

eOAP parameters

Name	Type	Size
eOAP_Site_id_n	Integer	2
eOAP_Area_id_n	Integer	2
eOAP_Framework_Address_str	Text	15
eOAP_Framework_Port_n	Integer	2
eOAP_AIA_Prty_DTMF_Confirm_n	Integer	2
eOAP_Silence_intv_n	Integer	2
eOAP_Comments_str	Text	255

eOAP_Site_id_n

This field describes the site identifier, as defined in eKERNEL_SITE table. An instance of eOAP is uniquely defined by means of a site and area identifier.

An authorized of an entry typically found in this field is as follows: 1

eOAP_Area_id_n

This field describes the area identifier, as defined in eKERNEL_AREA table. An instance of eOAP is uniquely defined by means of a site and area identifier.

An authorized of an entry typically found in this field is as follows: 1

eOAP_Framework_Address_str

This field describes the IP address of the Framework that handles the OAP Services.

An authorized of an entry typically found in this field is as follows: 127.0.0.1

eOAP_Framework_Port_n

This field describes the port number of the Framework that handles the OAP Services.

An authorized of an entry typically found in this field is as follows: 9090

eOAP_ALA_Prty_DTMF_Confirm_n

This value refers to the priority of the alarm as specified in eKERNEL_ALARM table. Alarms distributed to eOAP with a priority above the defined value are automatically considered acknowledged when the destination receives the message. For most cases this is suitable.

However, eOAP could deliver messages to infrastructure that are unable to respond. In some circumstances the message needs to remain active until a manual confirmation takes place. This can be done through eOAP (inbound SMS and confirm through CLIP or pincode), eCSTA (dial-in and confirm using CLID) or eVBVOICE (dial-in and confirm through DTMF).

Since eKernel release 2.9.18 the functionality is implemented that the message reply (<msgrpy>) sent by the eOAP module to the eKernel is treated as a NACK reply (even if a ACK was sent) in case the priority of the alarm is lower or equal (so has an higher importance) then the eOAP_ALA_Prty_DTMF_Confirm_n priority. This means that alarms that are sent by OAP (and are successfully delivered (so status = ACK)) and need a confirmation, behave the same as alarms with status NACK. The result is the alarm is repeated every eOAP_Silence_intv_nseconds until confirmation, and proceeds with the alternative device(s) (if configured) if not confirmed within the DEV_Retry_count_ALT_DEV_id_n (eKernel_device) retries.

An authorized of an entry typically found in this field is as follows: 2

eOAP_Silence_intv_n

This value specifies the silence interval in seconds, the frequency users are informed on remaining active messages. The default value is 600. The function is enabled to prevent calling the provider over again for each individual change that occurs, and thus leads module and their destination users some pace interval.

Note that a similar value is implemented in eKERNEL_ALARM table. The value here overrides the value in the eKERNEL_ALARM table due to bandwidth constraints.

An authorized of an entry typically found in this field is as follows: 600

eOAP_Comments_str

Use this field to enter administrator comments. It is informational only.

An authorized of an entry typically found in this field is as follows: Default eOAP configuration

Chapter 43: Table: eSMTP_CLIENT

eSMTP_CLIENT parameters

Name	Type	Size
eSMTP_Site_id_n	Integer	2
eSMTP_Area_id_n	Integer	2
eSMTP_Srv_ip_str	Text	15
eSMTP_Srv_port_str	Text	5
eSMTP_Srv_domain_str	Text	128
eSMTP_ALA_Prty_DTMF_Confirm_n	Integer	2
eSMTP_Silence_intv_n	Integer	2
eSMTP_From_address_str	Text	50
eSMTP_Comments_str	Text	255

eSMTP_Site_id_n

This field specifies the site identifier, as defined in eKERNEL_SITE table. In most environments, the value is 1.

An authorized of an entry typically found in this field is as follows: 1

eSMTP_Area_id_n

This field specifies the area identifier, as defined in eKERNEL_AREA table. In most environments, the value is 1.

An authorized of an entry typically found in this field is as follows: 1

eSMTP_srv_ip_str

This field specifies the IP address of the SMTP server that is used to handle outbound SMTP messages. This is typically an SMTP compatible system, for authorized, Lotus Notes, Lotus Domino, Microsoft Exchange, AS400 SMTP Server, iSeries400 SMTP server, Windows 2000 SMTP server, and so on.

The SMTP server must be properly configured to allow inbound SMTP requests from the DECT Messengerapplications (relaying, and so on).

An authorized of an entry typically found in this field is as follows: 10.110.17.6

eSMTP_srv_port_str

This field specifies the port number used for SMTP access. In most environments, this is value 25.

An authorized of an entry typically found in this field is as follows: 25

eSMTP_srv_domain_str

This field refers to the domain name used in the SMTP connection. Refer to the RFC821 specifications on the process involved in message delivery through SMTP. The domain parameter is associated to the HELO command in the SMTP dialog.

An authorized of an entry typically found in this field is as follows: ibsbe.be

eSMTP_ALA_Prty_DTMF_Confirm_n

This field specifies what alarm priority levels require a confirmation. Alarm priority is defined in the eKERNEL_ALARM table.

Alarms that do not meet the requirements are automatically confirmed when the DECT MessengerSMTP client sends a message to an external SMTP server. The message is considered sent when it reaches the server. However, at this stage, there is no guaranteed message delivery, because there is no read indication. This situation is similar to eASYNC, where SMS and PAGING as well do not foresee end user confirmation. An SMTP mail can be pending between intermediate server (for authorized, in an internet environment) or remain unread in the mailbox for a large amount of time.

Confirmation techniques can be appropriate to force mail destinations to respond to the alarm request. This can be accomplished by calling back to a predefined DID number. In eKERNEL release 2.9.18 and later, the functionality is implemented that if the priority of the alarm is lower than or equal to this value (so has an higher importance), the message reply (<msgrpy>) sent by the eSMTP module to the eKERNEL is treated as a NACK reply (even if a ACK was sent). Therefore, alarms that are sent using eSMTP (and are successfully delivered (so status = ACK)), and that need a confirmation, have the same behavior as alarms with status NACK.

This results in the alarm repeating every eSMTP_Silence_intv_n seconds until confirmation. If the alarm is not confirmed within the DEV_Retry_count_ALT_DEV_id_n (eKERNEL_device) retries, the alarm is sent to the alternative devices (if configured).

A value of, for authorized, 2 indicates alarms with priority 0,1 and 2 are considered to be confirmed using this callback procedure.

An authorized of an entry typically found in this field is as follows: 2

eSMTP_Silence_intv_n

This field specifies the silence interval, the time between repeating outstanding messages that need confirmation. The parameter corresponds with the parameter available in the eKERNEL_ALARM table, but overrules the latter value. Due to bandwidth restrictions, a larger value than specified in eKERNEL_ALARM table is suitable. For authorized, repeating unconfirmed alarms every two minutes in a mail destination environment is not desirable. A typical value is ten minutes. The value must be expressed in seconds.

An authorized of an entry typically found in this field is as follows: 600

eSMTP_From_address_str

This field specifies the e-mail address of the sender of both eSMTP module and eWEB module (form Send SMTP Message). The specified value is used in the MAIL FROM tag of the mail composition process, as spec RFC821 and RFC1521.

Note:

In R3.0, there is now the ability to specify a friendly name as well. The module eSMTP and eWEB now support any of the following three syntax:

francis.missiaen@ibsbe.be
<francis.missiaen@ibsbe.be>
Francis Missiaen <francis.missiaen@ibsbe.be>

An authorized of an entry typically found in this field is as follows: francis.missiaen@ibsbe.be

eSMTP_Comments_str

This field can contain remarks from the administrator and is informational only.

Table 54: eSMTP_CLIENT sample data on page 361 shows authorizeds of data found in the eSMTP_CLIENT table.

Table 54: eSMTP_CLIENT sample data

Site	Area	Address	Port	Domain	Confirm	Interval	Comments
1	1	10.110.17.6	25	1s.be	1	600	

Table: eSMTP_CLIENT

Chapter 44: Table: eSMTP_SERVER

eSMTP_SERVER parameters

Name	Type	Size
eSMTPS_Site_id_n	Integer	2
eSMTPS_Area_id_n	Integer	2
eSMTPS_Email_dir_str	Text	255
eSMTPS_Poll_intv_n	Integer	2
eSMTPS_Email_dir_processed_str	Text	255
eSMTPS_Email_keep_processed_n	Integer	2
eSMTPS_Email_dir_error_str	Text	255
eSMTPS_Email_keep_error_n	Integer	2
eSMTPS_Delivery_text_str	Text	255
eSMTPS_NonDelivery_text_str	Text	255
eSMTPS_ALA_id_n	Long Integer	4
eSMTPS_Comments	Text	255

eSMTP_Site_id_n

This field denotes the site identifier, as defined in eKERNEL_SITE table. In most environments, this field has a value of 1.

An authorized of an entry typically found in this field is as follows: 1

eSMTPS_Area_id_n

This field denotes the area identifier, as defined in eKERNEL_AREA table. In most environments, this field has a value of 1.

An authorized of an entry typically found in this field is as follows: 1

eSMTPS_Email_dir_str

This field specifies the directory that is polled upon arrival of incoming e-mail. In the Windows 2000 environment with the Internet Information Server component SMTP server activated, this is typically c:\inetpub\mailroot\drop. The specified directory is the directory where the Windows shipped SMTP server drops incoming mail.

This directory contains e-mail files (with the extension .EML) that are processed by DECT MessengereSMTP_server module, which analyzes the inbound e-mail files and handles them as alarm input.

An authorized of an entry typically found in this field is as follows: c:\inetpub\mailroot\drop

eSMTPS_Poll_intv_n

This parameter defines the interval between individual poll operations the eSMTP_server module handles to look for inbound mail. The value is expressed in seconds, and typically has a value of 10 seconds.

Specifying a smaller value requires additional system resources, and can speed up the detection process of inbound e-mail based alarm generation. Note however that e-mail processing is as such a technology that is not designed to guarantee lightning-speed response, and therefore a very small interval does not lead to substantial benefit. Only in very special environments with internal LAN-only mail exchange and dedicated resources are time-critical intervals suitable.

An authorized of an entry typically found in this field is as follows: 10

eSMTPS_Email_dir_processed

Once an inbound e-mail is detected, the eSMTP_server module moves the processed e-mail message an archive storage location.

A special value *NONE can be defined here, indicating the processed e-mail messages are not kept online, and are removed from the hard disk. Although some kind of logging information is often still available, the originating mail message is destroyed.

In most cases, a directory name is specified, and defines the location where the processed e-mail messages are temporarily archived. This archive allows system administrators to perform more detailed problem analysis.

Warning: the value specified must be different from the value specified in the eSMTPS_Email_dir_str parameter, or otherwise an infinite looping condition occurs. The eSMTP_server module attempts to create the hierarchical directory structure if the path does not exist.

An authorized of an entry typically found in this field is as follows: c:\inetpub\mailroot\drop \processed.

eSMTPS_Email_keep_processed_n

This field specifies the number of days the archive of processed e-mail messages is kept on the hard disk. The value is expressed in days, and has typically a value of 5 days.

Adjust this value to accommodate for the number of inbound e-mail messages, the requested archive period, and the available disk space.

An authorized of an entry typically found in this field is as follows: 5

eSMTPS_Email_dir_error_str

Once an inbound e-mail is detected, the eSMTP_server module moves the processed e-mail message to some kind of archive storage location. This location is defined in eSMTPS_Email_dir_processed_str. Mail that cannot be processed is moved to a separate location, defined in eSMTPS_Email_dir_error_str.

A special value *NONE can be defined here, indicating the e-mail messages in error are not kept online, and are removed from the hard disk. Although some kind of logging information is often still available, the originating mail message is destroyed.

In most cases, a directory name is specified, and defines the location where the e-mail messages in error are temporarily archived. This archive allows system administrators to perform more detailed problem analysis.

Warning: the value specified must be different from the value specified in the eSMTPS_Email_dir_str parameter, or otherwise infinite looping condition occurs. The eSMTP_server module attempts to create the hierarchical directory structure if the path does not exist.

An authorized of an entry typically found in this field is as follows: c:\inetpub\mailroot\drop\error

eSMTPS_Email_keep_error_n

This field specifies the number of days the archive of e-mail messages in error is kept on the hard disk. The value is expressed in days, and has typically a value of 5 days.

Adjust this value to accommodate the number of inbound e-mail messages, the requested archive period, and the available disk space.

An authorized of an entry typically found in this field is as follows: 5

eSMTPS_Delivery_text_str

When an inbound e-mail message is accepted by eKERNEL, the sender receives a delivery report. This delivery report is sent through eSMTP client. (The eSMTP module is a prerequisite.)

The message text for the delivery messages is defined in the eSMTPS_Delivery_text_str field.

An authorized of an entry typically found in this field is as follows: MESSAGE SUCCESSFULLY DELIVERED

eSMTPS_NonDelivery_text_str

When an inbound e-mail message is rejected by eKERNEL, the sender receives a non-delivery report. This non-delivery report is sent through eSMTP client. (The eSMTP module is a prerequisite.)

The message text for the non-delivery messages is defined in the eSMTPS_NonDelivery_text_str field.

An authorized of an entry typically found in this field is as follows: MESSAGE COULD NOT BE DELIVERED

eSMTPS_ALA_id_n

When an inbound e-mail message is accepted or rejected by eKERNEL, the sender receives a delivery or non-delivery report. This report is sent from eKERNEL to eSMTP client. (The eSMTP module is a prerequisite.)

To produce such outbound message, eKERNEL must known the alarm identifier that is used to produce the message for eSMTP. This value must match the value specified in eKERNEL_ALARM table. Verify the length of the delivery and non-delivery messages specified in eSMTPS_Delivery_text_str and eSMTPS_NonDelivery_text_str.

An authorized of an entry typically found in this field is as follows: 1180101

eSMTPS_Comments

This field can contain remarks from an administrator and is informational only.

Chapter 45: Table: eWEB

eWEB parameters

Name	Type	Size
eWEB_Address_str	Text	15
eWEB_Site_id_n	Integer	2
eWEB_Area_id_n	Integer	2
eWEB_eKernel_address_str	Text	15
eWEB_Branding_str	Text	50
eWEB_Comments_str	Text	50

eWEB_Address_str

This field specifies the IP address of the system where the Apache Web Server is running.

You can obtain the address with the IPCONFIG command. The eWEB module uses this address to obtain its site, area number, and the address of the eKERNEL (based upon eWEB table) and to obtain the port number at which eKERNEL listens (based upon eKERNEL_TCPCLIENT table).

This process is carried out in the PHP-scripts that run on the Apache Web Server. As a result, the Web Server can use its own IP address to retrieve the configuration data from the database. The values are needed in eWEB to set up a proper socket connection to eKERNEL module, and to give the user access to the correct site and area-related data. You can define multiple addresses for the same eWEB module.

An authorized of an entry typically found in this field is as follows: 10.100.50.138

eWEB_Site_id_n

This field specifies the site number associated to the eWEB instance obtained by the IP address of the Web Server. In most cases this value is 1, as defined in eKERNEL_SITE.

An authorized of an entry typically found in this field is as follows: 1

eWEB_Area_id_n

This field specifies the area number associated to the eWEB instance obtained by the IP address of the Web Server. In most cases this value is 1, as defined in eKERNEL_AREA.

An authorized of an entry typically found in this field is as follows: 1

eWEB_eKERNEL_address_str

This field specifies the IP address of the eKERNEL. In the current release, this value is the same as the eWEB_Address_str field. Therefore, eKERNEL and the Apache Web Server must reside on the same computer. Future releases can implement the architecture of distributed web servers that reside on another system (for authorized, located in a DMZ).

An authorized of an entry typically found in this field is as follows: 10.100.50.138

eWEB_Branding_str

This field is introduced in R3.0 and defines the branding information shown in eWEB user interface.

Note that tampering with branding information without permission is a copyright violation.

An authorized of an entry typically found in this field is as follows: AVAYA

eWEB_Comments_str

This field can contain remarks from the administrator and is informational only.

Table 55: eWEB sample data on page 368 shows authorizeds of data found in the eWEB table.

Table 55: eWEB sample data

Address	Site	Area	Kernel address	Comments
10.110.50.138	1	1	10.110.50.138	
10.110.53.138	1	1	10.110.53.138	
127.0.0.1	1	1	127.0.0.1	

Chapter 46: Table: eWEB_SCRIPT

eWEB parameters

Name	Type	Size
WSC_Site_id_n	Integer	2
WSC_Area_id_n	Integer	2
WSC_Script_id_n	Integer	2
WSC_Script_Descr_str	Text	50
WSC_GRP_Name_str	Text	128
WCS_ALA_id_n	Long	4
WSC_Msg_str	Text	255
WSC_Min_dev_cnt_str	Text	50
WSC_Max_Active_n	Text	50
WSC_Currently_Active_n	Integer	2
WSC_Comments_str	Text	255

WSC_Site_id_n

This field specifies the site, as defined in eKERNEL_SITE table. In most environments, the value is 1.

An authorized of an entry typically found in this field is as follows: 1

WSC_Area_id_n

This field specifies the area, as defined in eKERNEL_AREA table. In most cases the value is 1.

An authorized of an entry typically found in this field is as follows: 1

WSC_Script_id_n

This field specifies the unique identifier of the script message within one site.

Script messages are a special type of message requests with the unique feature of being traceable.

Although you are free to enter a numeric value of choice, Avaya recommends establishing a naming convention for script messages.

In the field ALA_Trace_b of the eKernel_alarm table, the administrator can activate this field (note that this feature is supported only for script messages in the current release), which means that the whole call flow is logged in the data database.

An authorized of an entry typically found in this field is as follows: 1

WSC_Script_Descr_str

This field is a description of the script message.

In the eWeb module, the visualization of the script message is performed with the description of the script message, and never with the script ID.

An authorized of an entry typically found in this field is as follows: EMERGENCY

WSC_GRP_Name_str

This parameter specifies the name of the group as defined in the field GRP_Name_str in the eKernel_group table or another valid text is *ALL.

If this field is equal to *ALL, the user can select a group, otherwise the group (the message destinations) are fixed.

The groups are presented as message destinations.

If the group name defined does not match a group name in the eKernel_group table, no devices are shown, so the alarm is not processed.

An authorized of an entry typically found in this field is as follows: EVACUATION

WCS_ALA_id_n

This field must have a value that corresponds with any of the definitions in the eKernel_alarm table for input program related to eWEB. For authorized, if eWEB is input program 11701 and eKernel_alarm table contains alarm identifiers 1170101 and 1170102, one of these defined values must be used. In most cases, a number of alarm identifiers are defined to handle different message lengths and different message priorities.

An authorized of an entry typically found in this field is as follows: 1170101

WSC_Msg_str

This field describes the message that is sent to the group members. Avaya recommends that you add a descriptive message that allows the recipient sufficient information to handle the alarm condition. Mobile users often lack immediate access to other information resources, such as a site map or technical specification. Avaya recommends that you keep the message length less than or equal to the maximum length defined in the associated eKERNEL_ALARM table.

*FREE is the only other valid entry. This keyword enables the end user to enter a message.

An authorized of an entry typically found in this field is as follows: EVACUATION is active

WSC_Min_dev_cnt_str

This field specifies the minimum number of devices that must be selected from the group by the end user, before a script message can be activated. The only other valid entry in the current release is *ALL; therefore, all devices from the group receive the message, so the end user does not have the opportunity to select devices.

Warning: you must not specify a value larger then the number of devices present in the group.

Note:

In the current release, this parameter has nothing to do with the number of devices that must receive the message before clearing the message for all other devices from the group.

An authorized of an entry typically found in this field is as follows: *ALL

WSC_Max_Active_n

This field specifies the maximum number of times this script message can be active. The keyword *NOMAX can be used to indicate there is no limit.

An authorized of an entry typically found in this field is as follows: 1 (for EVACUATION) or *NOMAX (for informative messages)

WSC_Currently_Active_n

This field specifies the number of script messages currently active.

This field is used by the eKernel application, and has nothing to do with configuration of the database.

WSCA_Comments_str

This field can be used by an administrator to enter remarks. The field is informational only.

Chapter 47: Table: eWEB_SCRIPT_SET_AUTH

eWEB_SCRIPT_SET_AUTH parameters

Name	Type	Size
WSSA_Site_id_n	Integer	2
WSSA_Area_id_n	Integer	2
WSSA_Script_id_n	Integer	2
WSSA_UserID_str	Text	10
WSSA_Comments_str	Text	255

WSSA_Site_id_n

This field specifies the site, as defined in eKERNEL_SITE table. In most environments, the value is 1.

An authorized of an entry typically found in this field is as follows: 1

WSSA_Area_id_n

This field specifies the area, as defined in eKERNEL_AREA table. In most cases the value is 1.

An authorized of an entry typically found in this field is as follows: 1

WSSA_Script_id_n

This field must have a value that corresponds with any of the definitions in eWEB_script table for the eWEB interface.

An authorized of an entry typically found in this field is as follows: 1

WSSA_UserID_str

This field must have a username that corresponds with the USERA_UserID_str field of the eWeb_user_auth table or can be the keyword *ALL.

If the value *ALL is entered, any user can set this script message. If one or more users are defined, only those users can set the related script message.

If nothing configured in this table for a specific script message, no one can activate this script message.

An authorized of an entry typically found in this field is as follows: KDS

WSSA_Comments_str

This field can be used by an administrator to enter remarks. The field is informational only.

Chapter 48: Table: eWEB_SCRIPT_TRACE_AUTH

Note:

An alarm is only traceable for script message if the ALA_Trace_b alarm ID related to the script message has the field ALA_Trace_b in the eKERNEL_ALARM table set to True.

Note:

In the current release, traceable alarms are only supported for script messages.

eWEB_SCRIPT_TRACE_AUTH parameters

Name	Type	Size
WSTA_Site_id_n	Integer	2
WSTA_Area_id_n	Integer	2
WSTA_Script_id_n	Integer	2
WSTA_UserID_str	Text	10
WSTA_Auth_str	Text	50
WSTA_Comments_str	Text	255

Figure 198: eWEB_SCRIPT_TRACE_AUTH parameters

WSTA_Site_id_n

This field specifies the site, as defined in eKERNEL_SITE table. In most environments, the value is 1.

An authorized of an entry typically found in this field is as follows: 1

WSTA_Area_id_n

This field specifies the area, as defined in eKERNEL_AREA table. In most cases the value is 1.

An authorized of an entry typically found in this field is as follows: 1

WSTA_Script_id_n

This field must have a value that corresponds with any of the definitions in eWEB_script table for the eWEB interface.

An authorized of an entry typically found in this field is as follows: 1

WSTA_UserID_str

This field must have a username that corresponds with the USERA_UserID_str field of the eWeb_user_auth table or can be the keyword *ALL.

If the value *ALL is entered, any user can trace this script message. If one or more users are defined, only those users can trace the related script message.

If nothing configured in this table for a specific script message, no one can trace this script message.

An authorized of an entry typically found in this field is as follows: KDS

WSTA_Auth_str

This field is provided for security enhancements in future releases.

Only the value *VIEW and *EXCLUDE are supported in the current release.

If the end user must have the authority to trace a script message, this field must be *VIEW. *EXCLUDE is similar to not entering a record.

An authorized of an entry typically found in this field is as follows: *VIEW

WSTA_Comments_str

This field can be used by an administrator to enter remarks. The field is informational only.

Chapter 49: Table: eWEB_SCRIPT_CANCEL_AUTH

eWEB_SCRIPT_CANCEL_AUTH parameters

Name	Type	Size
WSCA_Site_id_n	Integer	2
WSCA_Area_id_n	Integer	2
WSCA_Script_id_n	Integer	2
WSCA_UserID_str	Text	10
WSCA_Comments_str	Text	255

WSCA_Site_id_n

This field specifies the site, as defined in eKERNEL_SITE table. In most environments, the value is 1.

An authorized of an entry typically found in this field is as follows: 1

WSCA_Area_id_n

This field specifies the area, as defined in eKERNEL_AREA table. In most cases, the value is 1.

An authorized of an entry typically found in this field is as follows: 1

WSCA_Script_id_n

This field must have a value that corresponds with any of the definitions in eWEB_script table for the eWEB interface.

An authorized of an entry typically found in this field is as follows: 1

WSCA_UserID_str

This field must have a username that corresponds with the USERA_UserID_str field of the eWeb_user_auth table or can be the keyword *ALL.

If the value *ALL is entered, any user can cancel this script message. If one or more users are defined, only those users can cancel the related script message.

If nothing configured in this table for a specific script message, no one can cancel this script message.

An authorized of an entry typically found in this field is as follows: Admin

WSCA_Comments_str

This field can be used by an administrator to enter remarks. The field is informational only.

Chapter 50: Table: eWEB_SNDGRPMSG

eWEB_SNDGRPMSG parameters

Name	Type	Size
WGM_Site_id_n	Integer	2
WGM_Area_id_n	Integer	2
WGM_GRP_Name_str	Text	128
WGM_Sequence_n	Integer	2
WGM_Message_str	Text	80
WGM_AlA_id_n	Long Integer	4
WGM_Comments_str	Text	255

WGM_Site_id_n

This field specifies the site identifier, as described in table eKERNEL_SITE. In most cases this value is 1.

An authorized of an entry typically found in this field is as follows: 1

WGM_Area_id_n

This field specifies the area identifier, as described in table eKERNEL_AREA. In most cases this value is 1.

An authorized of an entry typically found in this field is as follows: 1

WGM_GRP_Name_str

This field specifies the group, as defined in eKERNEL_GROUP table. The Send Group Message function in eWEB allows sending a predefined message to a group. The table eWEB_SNDGRPMSG allows a system administrator to predefine a number of messages that are automatically presented to a web user in the web-based Send Group Message functionality.

The field can either contain a qualified group name or can have the generic special value *ALL. This special value *ALL means the message is automatically defined for all groups. You must use this value only when appropriate, as sharing messages affects all groups.

When entering a value in this field, ensure that the specified group name exists in the eKERNEL_GROUP table, and that the eKERNEL_GROUP_MEMBER contains at least one member.

An authorized of an entry typically found in this field is as follows: 00001 (qualified group) or *ALL (generic group)

WGM_Sequence_n

This field is a sequence number and makes the records unique in the database. The field allows you to define the sequence used to present the data in the Send Group Message function. Avaya recommends that you start with a value of 1 and increase by one for subsequent messages.

An authorized of an entry typically found in this field is as follows: 1

WGM_Message_str

This field specifies the message that is shown to the eWEB user in the Send Group Message functionality, and is finally sent to the destination users.

Note the length of the message must be smaller than or equal to the maximum length associated with the WGM_AlA_id_n definition in eKERNEL_ALARM table. For authorized, when an alarm identifier defines maximum length in eKERNEL_ALARM table of 48 bytes, the specified message must not be longer that 48 bytes. A special value *FREE can be defined, enabling the end user to enter a message.

An authorized of an entry typically found in this field is as follows: Evacuation (qualified) or *FREE (user-defined message)

WGM_AlA_id_n

This field must have a value that corresponds with any of the definitions in eKERNEL_ALARM table for the eWEB interface. For authorized, if eWEB is input program 11701 and ALARM table contains alarm identifiers 1170101 and 1170102 and 1170103, one of these defined values must be used. In most cases, a number of alarm identifiers are defined to handle different message lengths and different message priorities.

An authorized of an entry typically found in this field is as follows: 1170101

WGM_Comments_str

This field can be used by an administrator to enter some remarks. The field is informational only.

Table 56: eWEB_SNDGRPMSG sample data on page 381 shows authorizeds of data found in the eWEB_SNDGRPMSG table.

Table 56: eWEB_SNDGRPMSG sample data

Site	Area	Group	Sequence	Message	Alarm id	Comments
3	1	*ALL	1	Emergency - evacuation	3170103	
3	1	*ALL	2	*FREE	3170102	
3	1	1	1	AS400 failure	3170102	
3	1	1	2	NT failure	3170102	
3	1	1	3	Domino failure	3170102	
3	1	1	4	Firewall failure	3170102	
3	1	2	1	Check invoice	3170102	
3	1	2	2	Check mailbox	3170102	
3	1	2	3	Check quotations	3170102	
3	1	2	4	Check received goods	3170102	
3	1	RAMPENPLAN	1	Fase 1 - start	3170102	
3	1	RAMPENPLAN	2	Fase 2 - start	3170102	
3	1	RAMPENPLAN	3	Fase 3 - start	3170102	
3	1	RAMPENPLAN	4	Fase 1 - end	3170102	

Table: eWEB_SNDGRPMSG

Site	Area	Group	Sequence	Message	Alarm id	Comments
3	1	RAMPENPLAN	5	Fase 2 - end	3170102	
3	1	RAMPENPLAN	6	Fase 3 - end	3170102	
3	1	VSK_F	1	Brand - gelijkvloers	3170102	
3	1	VSK_F	2	Brand - verdieping 1	3170102	
3	1	VSK_F	3	Brand - verdieping 2	3170102	

Chapter 51: Table: eWEB_SNDUSRMSG

eWEB_SNDUSRMSG parameters

Name	Type	Size
WUM_User_id_str	Text	10
WUM_Sequence_n	Integer	2
WUM_Message_str	Text	80
WUM_AlA_id_n	Long Integer	4
WUM_Comments_str	Text	255

WUM_User_id_str

This field specifies the user, as defined in eWEB_USER_AUTH table. The user is defined at the login process, where the web user enters a valid user and password. This user name is stored in the Web browser and reused as needed when authentication is needed for Web requests. The table eWEB_SNDUSRMSG allows a system administrator to predefine a number of messages that are automatically presented to a web user in the web-based Send User Message functionality.

The field can either contain a qualified username or can have the generic special value *ALL. This special value *ALL means the message is defined for all users.

An authorized of an entry typically found in this field is as follows: 00001 (qualified user) or *ALL (generic user)

WUM_Sequence_n

This field is a sequence number and makes the WUM_User_id_str and WUM_Sequence_n a unique key. Use WUM_Sequence_n to define the sort sequence of the available predefined messages. Avaya recommends that you start with a value of 1 and increase by one for subsequent messages.

An authorized of an entry typically found in this field is as follows: 1

WUM_Message_str

This field specifies the message that is shown to the eWEB user in the Send User Message functionality, and finally is sent to the destination users. Note the length of the message must be smaller than or equal to the maximum length associated with the WUM_AlA_id_n definition in eKERNEL_ALARM table. For authorized, when an alarm identifier defines maximum length in eKERNEL_ALARM table of 48 bytes, the specified message must not be longer that 48 bytes. A special value *FREE can be defined, enabling the end user to enter a message.

An authorized of an entry typically found in this field is as follows: Evacuation (qualified) or *FREE (user-defined message)

WUM_AlA_id_n

This field must have a value that corresponds with any of the definitions in ALARM table for the eWEB interface. For authorized, if eWEB is input program 11701 and eKERNEL_ALARM table contains alarm identifiers 1170101 and 1170102 and 1170103, one of these defined values must be used. In most cases, a number of alarm identifiers are defined to handle different message lengths and different message priorities.

An authorized of an entry typically found in this field is as follows: 1170101

WGM_Comments_str

This field can be used by an administrator to enter some remarks. The field is informational only.

Table 57: eWEB_SNDUSRMSG sample data on page 384 shows authorizeds of data found in the eWEB_SNDUSRMSG table.

Table 57: eWEB_SNDUSRMSG sample data

User	Sequence	Message	Alarm id	Comments
*ALL	1	Normal message 1 for *ALL	3170101	
*ALL	2	Shared message 2 for *ALL	3170101	
*ALL	3	Shared message 3 for *ALL	3170101	
*ALL	4	Shared message 4 for *ALL	3170101	
*ALL	5	Shared message 5 for *ALL	3170101	
*ALL	6	Shared message 6 for *ALL	3170101	
*ALL	7	*FREE	3170101	

User	Sequence	Message	Alarm id	Comments
FMI	1	Private message 1 for FMI	3170103	
FMI	2	Private message 2 for FMI	3170103	
FMI	3	Private message 3 for FMI	3170103	
FMI	4	Private message 4 for FMI	3170103	
KDS	1	Private message 1 (Medium)	3170102	
KDS	2	Private message 2 (Short)	3170101	
KDS	3	Private message 3 (Long)	3170103	

Table: eWEB_SNDUSRMSG

Chapter 52: Table: eWEB_TOC

eWEB_TOC parameters

Name	Type	Size
WTC_Site_id_n	Integer	2
WTC_Group_n	Integer	2
WTC_Item_n	Integer	2
WTC_Language_str	Text	4
WTC_Text_str	Text	35
WTC_Link_str	Text	80
WTC_Sec_n	Integer	2
WTC_Comments_str	Text	255

WTC_Site_id_n

This field specifies the site identifier, as defined in eKERNEL_SITE table. The site is in most cases equal to 1.

The Web Server determines its site and area based upon its own IP address, as defined in the eWEB table.

An authorized of an entry typically found in this field is as follows: 1.

WTC_Group_n

This field contains a numeric sequence number, which is combined with WTC_Item_n and WTC_Language_str to generate a key. The key is unique within the site. WTC_Group_n is used to logically sort the table of contents in groups and items. Avaya recommends starting the first group at 1 and incrementing by 1.

An authorized of an entry typically found in this field is as follows: 1.

WTC_Item_n

This field contains a numeric sequence number, which is combined with WTC_Group_n, and WTC_Language_str to generate a key. The key is unique within the site. WTC_Item_n is used

to logically sort table of contents in groups and items. Avaya recommends starting the first item in a group at 1 and incrementing by 1.

An authorized of an entry typically found in this field is as follows: 1.

WTC_Language_str

This field contains a 4-byte language code. Refer to the documentation of the Table: eWEB_USER_AUTH on page 393 for a list of language codes. This field contains a number, and when combined with WTC_Group_n, and WTC_Item_n, results in a key, which is not duplicated within a site.

This field specifies the language used in the field WTC_Text_str, and in the PHP script of HTML documents defined in WTC_Link_str.

This field allows the table of contents to be multilingual. With the correct definition, English users see the table of contents in English, Dutch users in Dutch, and so on.

To implement a new language:

1. Define the appropriate language code in the eWEB_USER_AUTH table.
2. Translate the descriptions of the links in the eWEB_TOC table.
3. Edit the eWeb_mri.php file that is located in C:\SOPHO Messenger@Net\Web \htdocs.
4. Provide an additional section for the new language. The eWeb_mri.php is provided in English (2909), and Dutch (2963).

An authorized of an entry typically found in this field is as follows: 2909.

WTC_Text_str

This field specifies the text that the web user sees in the table of contents. Avaya recommends using the same language as specified in the field WTC_Language_str.

An authorized of an entry typically found in this field is as follows: Welcome (in English - 2909) or Welkom (in Dutch - 2963).

WTC_Link_str

This field specifies the hyperlink associated with the table of contents. If blank, the hyperlink is inactive. This is typically used to logically group menu options in different sections, and define such empty link for the header of each section. See the sample in Table 58: Valid WTC_Link_str values on page 389 for more information.

In most cases, this field contains a valid filename of a PHP-script, a HTML- filename of another valid string understood by a browser (for authorized, mailto:francis.missiaen@1s.be).

Table 58: Valid WTC_Link_str values on page 389 provides a list of valid links that can be used. The files are shipped with eWEB module and are located in C:\SOPHO Messenger@Net \Web\htdocs.

Table 58: Valid WTC_Link_str values

eWEB_alarm_inquiry.php
eWEB_chgpwd.php
eWEB_device_inquiry.php
eWEB_eDMSAPI.php
eWEB_eSMTP.php
eWEB_group_inquiry.php
eWEB_script.php
eWEB_sndgrpmsg_1.php
eWEB_sndsrvmsg.php
eWEB_sndusrmsg_1.php
eWEB_table_view.php
eWeb_wrkgrp_1.php
info.html
mailto:francis.missiaen@1s.be
1s/launch.htm

An authorized of an entry typically found in this field is as follows: eWEB_eDMSAPI.php

WTC_Sec_n

This field specifies whether a user can see table of contents items. For authorized, a user with security level 20 defined in the eWEB_USER_AUTH table sees only the table of contents items defined in the eWEB_TOC table with a WTC_Sec_n value lower than or equal to 20. WTC_Sec_n provides a method to restrict access to some functionality to a subset of users.

An authorized of an entry typically found in this field is as follows: 20

WTC_Comments_str

This field can be used by an administrator to enter remarks. The field is informational only.

Table 59: eWEB_TOC sample configuration on page 390 shows authorizeds of data found in the eWEB_TOC table. Figure 199: eWEB_TOC sample result (language 2909 and language 2963) on page 392 shows an authorized of the eWEB_TOC result for language 2909 and language 2963.

Table 59: eWEB_TOC sample configuration

Site	Group	Item	Language	Text	Link	Level
3	2	0	2909	Send a message		10
3	2	0	2963	Zend een boodschap		10
3	2	1	2909	Send DMS-API message	eWEB_eDMSAPI.php	10
3	2	1	2963	Zend DMS-API boodschap	eWEB_eDMSAPI.php	10
3	2	2	2909	Send SMTP message	eWEB_eSMTP.php	10
3	2	2	2963	Zend SMTP boodschap	eWEB_eSMTP.php	10
3	2	3	2909	Send Server Message	eWEB_sndsrvmsg.php	10
3	2	3	2963	Zend Server boodschap	eWEB_sndsrvmsg.php	10
3	2	4	2909	Send Group Message	eWEB_sndgrpmsg_1.php	10
3	2	4	2963	Zend Groep boodschap	eWEB_sndgrpmsg_1.php	10
3	2	5	2909	Send User Message	eWEB_sndusrmsg_1.php	10
3	2	5	2963	Zend Gebruiker boodschap	eWEB_sndusrmsg_1.php	10
3	3	0	2909	Send a script message		40
3	3	0	2963	Zend een script boodschap		40
3	3	1	2909	Work with Script messages	eWEB_script.php	40
3	3	1	2963	Werken met Script boodschappen	eWEB_script.php	40
3	4	0	2909	Inquiry		20

Site	Group	Item	Language	Text	Link	Level
3	4	0	2963	Overzicht		20
3	4	1	2909	Alarm Inquiry	eWEB_alarm_inquiry.php	20
3	4	1	2963	Alarm overzicht	eWEB_alarm_inquiry.php	20
3	4	2	2909	Device Inquiry	eWEB_device_inquiry.php	20
3	4	2	2963	Device overzicht	eWEB_device_inquiry.php	20
3	4	3	2909	Group Inquiry	eWEB_group_inquiry.php	20
3	4	3	2963	Groeps overzicht	eWEB_group_inquiry.php	20
3	4	4	2909	Table View	eWEB_table_view.php	20
3	4	4	2963	Tabel bekijken	eWEB_table_view.php	20
3	5	0	2909	Maintenance		30
3	5	0	2963	Onderhoud		30
3	5	1	2909	Work with Groups	eWeb_wrkgrp_1.php	30
3	5	1	2963	Werken met groepen	eWeb_wrkgrp_1.php	30
3	6	0	2909	Security		10
3	6	0	2963	Beveiliging		10
3	6	1	2909	Change Password	eWEB_chgpwd.php	10
3	6	1	2963	Paswoord wijzigen	eWEB_chgpwd.php	10
3	7	0	2909	Help		40
3	7	0	2963	Help		40
3	7	1	2909	Info	info.html	40
3	7	1	2963	Info	info.html	40
3	7	2	2909	Contact me	mailto:francis.missiaen@1s.be	40
3	7	2	2963	Kontakteer mij	mailto:francis.missiaen@1s.be	40
3	7	3	2909	Number One Systems	1s/launch.htm	40
3	7	3	2963	Number One Systems	1s/launch.htm	40

Table: eWEB_TOC

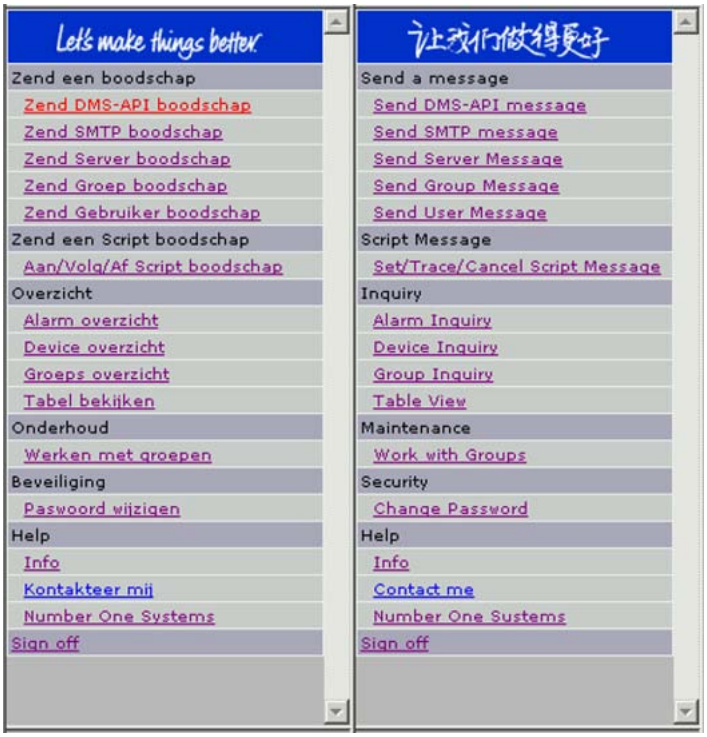


Figure 199: eWEB_TOC sample result (language 2909 and language 2963)

Chapter 53: Table: eWEB_USER_AUTH

EWEB_USER_AUTH parameters

Name	Type	Size
USERA_UserID_str	Text	10
USERA_Password_str	Text	10
USERA_Sec_level_n	Integer	2
USERA_Description_str	Text	50
USERA_Language_str	Text	4
USERA_Email_str	Text	100
USERA_Allobj_b	Yes/No	1
USERA_Secadm_b	Yes/No	1
USERA_Service_b	Yes/No	1
USERA_Comments_str	Text	255

USERA_UserID_str

This field contains a User ID. The eWEB module must define at least one user profile for authentication purposes. Avaya recommends that you create a user profile for each user that has access to the eWEB interface, to avoid alarms generated by unauthenticated users.

Note:

In many environments, other computer infrastructure is in use, such as iSeries 400, Windows NT, Lotus Notes, and so on, and users often desire to use the same username on every platform. In this case, Avaya recommends that you ask the ICT manager for a list of existing user profiles, so that DECT Messenger can use the same User IDs. On iSeries 400 the OS/400, command WRKUSRPRF can be used to determine defined users.

Note:

The USERA_UserID_str field is restricted to a length of 10 bytes.

An authorized of an entry typically found in this field is as follows: FMI.

USERA_Password_str

This field contains a 10-byte password. The eWEB interface allows users to change their own password. Therefore you can create new users with default passwords (for authorized, the

same as the User ID), and ask users to change their password when they log in for the first time.

Note:

eWEB stores passwords without encryption in the Access 2000 database, and are therefore available to anyone who can access the DECT Messengersystem. Depending on your configuration, table information is accessible through eWEB. Because the security mechanism is limited, Avaya recommends that users not use the same password used on other systems that contain secured information, as that poses a serious security risk. Inform all users of this important issue.

An authorized of an entry typically found in this field is as follows: SOPHO.

USERA_Sec_level_n

The security level is a number between 00 and 99. The higher the number, the more authority a user has. The value 99 is the highest level, and gives full access to all functionality. The value 00 is the lowest possible value. Avaya recommends that you initially assign values in 2 or 3 levels and handle increment by 10. For instance, start with the following values: 20 for low-end users, 40 for power users, and 60 for administrators.

Note:

The security level is related to the values specified in the eWEB_TOC table, where the field WTC_Sec_n level specifies the minimum required user security level that is needed for a specified function. For authorized, a user with level 20 can execute all the functions in WTC_Sec_n with level 00–20.

An authorized of an entry typically found in this field is as follows: 40.

USERA_Description_str

This field contains a description of the user, which usually consists of the first and last name of the User ID. This field is informational only.

An authorized of an entry typically found in this field is as follows: Francis Missiaen.

USERA_Email_str

This field specifies the e-mail address of the user. This field is important when eWEB module is activated, and the Send SMTP Message function is available to the users. When a user sends an e-mail message through the Send SMTP Message script, the system checks the username of the eWEB user, as specified during the login procedure. The e-mail address of

the user is retrieved based on the User ID, and is used in the MAIL FROM tag of the mail composition process, as defined in the RFC821 specifications.

An authorized of an entry typically found in this field is as follows: francis.missiaen@1s.be.

USERA_Allobj_b

This field specifies whether a user has the authority to access all objects. In most cases the value False (0) is used. This means the user does not have authority to access all objects. Instead, the user only has access to maintain the groups he or she has been granted access to, as defined in the eKERNEL_GROUP_AUTH table.

If your environment requires it, you can create users with administrator privileges, who are allowed to maintain any existing group through the eWEB based Work with Groups. To do so, set this field to True (-1) to grant the all object special authority to these users. Users with all object special authority do not need to be granted authority in the eKERNEL_GROUP_AUTH table.

Avaya recommends giving this special authority only to system administrators and service staff.

An authorized of an entry typically found in this field is as follows: 0 (denotes False).

USERA_Secadm_b

This field specifies whether a user has security administrator special authority. If this value is set to False (0), the user has access to all tables in the Table View within eWEB, except eWEB_USER_AUTH, which shows usernames and passwords in plain text.

If your environment requires it, you can create users with administrator privileges, who are allowed to maintain any user profile in eWEB. For those users, set this field to True (-1) to allow those users to consult the table eWEB_USER_AUTH, and see the user and password information.

Note:

The web interface only supports inquiry to the tables. Maintenance of the tables must be performed using the eGRID interface.

An authorized of an entry typically found in this field is as follows: 0 (denotes False).

USERA_Service_b

This value is not implemented in the current release. Avaya recommends using the value False (0). This feature is used in future releases to grant access to service functions that can be implemented in eWEB at a later stage.

An authorized of an entry typically found in this field is as follows: 0 (denotes False).

USERA_Language_str

This field contains a 4-byte identifier that denotes the language used for eWEB-access and eGRID-access. Enter one of the valid language codes provided in Table 60: Currently supported language values in eWEB on page 396. The codes are in the range 2900–2999. A small number of languages are currently supported, but additional languages can be implemented if needed.

Table 60: Currently supported language values in eWEB on page 396 shows the codes for currently supported languages, while Table 61: Language values reserved for future implementation on page 396 shows codes reserved for future language support.

Table 60: Currently supported language values in eWEB

Code	Language
2909	Belgian English
2963	Belgian Dutch
2966	Belgian French

Table 61: Language values reserved for future implementation

Code	Language
2902	Estonian
2903	Lithuanian
2904	Latvian
2905	Vietnamese
2906	Lao
2911	Slovenian
2912	Croatian
2913	Macedonian
2914	Serbian Cyrillic

Code	Language
2922	Portuguese
2923	Dutch Netherlands
2924	English
2925	Finnish
2926	Danish
2928	French
2929	German
2931	Spanish
2932	Italian
2933	Norwegian
2937	Swedish
2938	English Uppercase Support for
	Double-Byte Character Set (DBCS)
2939	German Multinational Character Set
2940	French Multinational Character Set
2942	Italian Multinational Character Set
2950	English Uppercase
2954	Arabic
2956	Turkish
2957	Greek
2958	Icelandic
2961	Hebrew
2962	Japanese Double-Byte Character Set (DBCS)
2963	Belgium Dutch
2966	Belgium French
2972	Thai
2974	Bulgarian
2975	Czech
2976	Hungarian
2978	Polish
2979	Russian

Code	Language
2980	Brazilian Portuguese
2981	Canadian French
2984	English Uppercase and Lowercase
	Support for Double-Byte Character Set (DBCS)
2986	Korean Double-Byte Character Set (DBCS)
2987	Traditional Chinese Double-Byte
	Character Set (DBCS)
2989	Simplified Chinese Double-Byte
	Character Set (DBCS) (PRC)
2992	Romanian
2994	Slovakian
2995	Albanian
2996	Portuguese Multinational Character Set
2998	Farsi

Note:

The language-code corresponds with an entry in eGRID that provides a directory where the language dependent files are stored. This path is usually C:\SOPHO Messenger@Net\pdf \mri29xx. The concept of multilingual support in the eWEB module is implemented in the file eWeb_mri.php that is located in C:\SOPHO Messenger@Net\Web\htdocs.

An authorized of an entry typically found in this field is as follows: 2909.

USERA_Comments_str

Use this field to record remarks about the user.

An authorized of an entry typically found in this field is as follows: Technical manager.

Index

Special Characters

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

X