

Entire Net-Work

Using Encryption for Entire Net-Work

Version 2.0

October 2021

This document applies to Entire Net-Work Version 2.0 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2021 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: WCPOS-OWCPOSWSL-20-20210929

Table of Contents

Preface	v
1 Conventions	1
2 About this Documentation	3
Document Conventions	4
Online Information and Support	4
Data Protection	5
3 Release Notes	7
Enhancements	8
Prerequisites	8
Supported Platforms	8
End-of-Support Dates	8
4 Concepts	9
Authentication	10
Encryption and Decryption	10
Certificate Authorities	10
5 Installing Encryption for Entire Net-Work	13
LUW Installation	14
6 Activating Encryption for Entire Net-Work	17
Open Systems Activation	18
7 Using the SSL Toolkit	21
Gathering SSL Toolkit Information	22
Setting Up a Certificate Authority	25
Creating Certificates	26
Deploying Certificates	29
8 Access and Connection Definition Setup	31
Maintaining Target Definitions	32
Security Parameters	32
9 Sample Security Scenarios	35
Simple Encryption	36
Other Authentication Scenarios	37
Index	41

Preface

Encryption for Entire Net-Work is a Software AG product option that provides support for the Secure Sockets Layer (SSL) management of message transmissions via Entire Net-Work. Encryption for Entire Net-Work is installed in LUW environments running Entire Net-Work Server and/or Entire Net-Work Client.

This document describes the use of Encryption for Entire Net-Work on all supported LUW platforms.

This Encryption for Entire Net-Work documentation is organized as follows:

<i>Release Notes</i>	Describes the changes, enhancements, prerequisites, migration considerations, and documentation for this release of Encryption for Entire Net-Work.
<i>Concepts</i>	Provides a high-level introduction to Encryption for Entire Net-Work.
<i>Installing Encryption for Entire Net-Work</i>	Describes the prerequisites for Encryption for Entire Net-Work and explains how to install it on the mainframe and on open systems.
<i>Activating Encryption for Entire Net-Work</i>	Describes the steps necessary to activate Encryption for Entire Net-Work on mainframe and open systems.
<i>Using the SSL Toolkit</i>	Explains how to use the SSL Toolkit on open systems to create certificates for testing your use of Encryption for Entire Net-Work.
<i>Access and Connection Definition Setup</i>	Describes the target definition access and connection definition syntax required for Encryption for Entire Net-Work support.
<i>Security Scenarios</i>	Provides mainframe and open systems examples of Encryption for Entire Net-Work use.

1 Conventions

Notation *vrs* or *vr*: When used in this documentation, the notation *vrs* or *vr* stands for the relevant version, release, and system maintenance level numbers. For further information on product versions, see *version* in the *Glossary*.

2 About this Documentation

■ Document Conventions	4
■ Online Information and Support	4
■ Data Protection	5

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Monospace font	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <https://documentation.softwareag.com>.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at <https://empower.softwareag.com/>.

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.aspx and give us a call.

Software AG Tech Community

You can find documentation and other technical information on the Software AG Tech Community website at <https://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have Tech Community credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

3

Release Notes

■ Enhancements	8
■ Prerequisites	8
■ Supported Platforms	8
■ End-of-Support Dates	8

This chapter describes the changes, enhancements, migration considerations, and documentation for this release.

Enhancements

Version 2.0 of Encryption for Entire Net-Work is built using updated OpenSSL libraries to improve security and eliminate known vulnerabilities in previous releases.

Prerequisites

The following prerequisites must be met before you can install Encryption for Entire Net-Work:

- Encrypted communication with mainframe Entire Net-Work requires Entire Net-Work 6.5.2 or later on z/OS with IBM's AT-TLS (Application Transparent-Transport Layer Security) configured. For more information, please see the mainframe Entire Net-Work documentation.
- Any manually configured Entire Net-Work Client Adabas Access entries must conform to the details described in the section [Access and Connection Definition Setup](#) of this documentation.

Supported Platforms

Encryption for Entire Net-Work is supported on the same platforms as Entire Net-Work Server and Entire Net-Work Client.

End-of-Support Dates

For information on how long a product is supported by Software AG, access Software AG's Empower web site at <https://empower.softwareag.com>.

Log into Empower. Once you have logged in, you can expand **Products** in the left menu of the web page and select **Product Version Availability** to access the Product Version Availability application. This application allows you to review support information for specific products and releases.

4 Concepts

■ Authentication	10
■ Encryption and Decryption	10
■ Certificate Authorities	10

Encryption for Entire Net-Work provides support for the Secure Sockets Layer (SSL) to manage the security of message transmissions. On LUW systems this support is provided by the add-on product Encryption for Entire Net-Work, a Software AG implementation of OpenSSL. On mainframe systems, this support is provided by configuration of IBM's AT-TLS (Application Transparent-Transport Layer Security) product.

Secure Sockets Layer (SSL) is a standard protocol used to manage the security of message transmissions in an open communications network, such as the Internet. Two types of security are provided:

Authentication

Using *digital signatures*, the partners in a conversation (the client and server) can be authenticated.

A digital signature is a digital code that can be attached to an electronically-transmitted message that uniquely identifies the sender. The purpose of a digital signature is to authenticate the identity of the individual sending the message using a private key to sign the message and a public key to verify the signed message. These keys are obtained from a certificate authority of some kind, as described in [Certificate Authorities](#), elsewhere in this section.

Encryption and Decryption

Using data *encryption* and *decryption*, messages are secured as they pass through the network.

Encryption is the conversion of data into ciphertext, which cannot be easily understood without access to the encryption or decryption key. Decryption is the process of converting encrypted data back into its original form, so it can be understood. To decrypt the contents of an encrypted message, a decryption key is required. Encryption keys are generated automatically after the successful handshake between the client and server. The handshake between the client and server is handled through the use of private and public keys, which are obtained from a certificate authority of some kind, as described in [Certificate Authorities](#), elsewhere in this section.

Certificate Authorities

A *certificate authority* issues and manages *certificates* for message encryption. It also verifies (authenticates) the information provided by the requestor of a digital certificate. If verification is successful, the certificate authority can then issue a certificate.

Various organizations, such as VeriSign, act as external certificate authorities for other companies and supply certificates for authentication and encryption as requested by their clients. For Entire

Net-Work, you can use an external certificate authority to provide your certificates or, *for testing only*, you can use the open source SSL Toolkit (provided with Encryption for Entire Net-Work) to become your own certificate authority.

For more information about the open source SSL Toolkit, read [Using the SSL Toolkit](#), elsewhere in this guide.

5

Installing Encryption for Entire Net-Work

■ LUW Installation	14
--------------------------	----

Before you install Encryption for Entire Net-Work, be sure you have met the requirements described in [Prerequisites](#), elsewhere in this guide.

LUW Installation

On LUW systems, the installation of Encryption for Entire Net-Work provides the required SSL libraries used in encryption. It also provides the open source SSL Toolkit that you can use to create certificates. Although delivered on all supported LUW platforms, the SSL Toolkit only runs on Windows. This section describes general information you should understand prior to completing the installation as well as providing installation and uninstallation steps for Windows environments.

- [Installation](#)
- [Uninstalling Encryption for Entire Net-Work](#)

Installation

Installation of Encryption for Entire Net-Work is accomplished by the following steps:

- Contact Software AG Support to obtain the delivery for the operating system(s) where you want to install Encryption for Entire Net-Work. The delivery is available as an archive (.zip for Windows, .tgz for all others) and will have the following format:

```
WSLv2.0.0.0_<platform>.zip/.tgz
```

...where <platform> corresponds to the operating system you are installing on.

- Extract the contents of the archive to a temporary folder on all machines where you already have either Entire Net-Work Server or Entire Net-Work Client already installed and you want to take advantage of SSL communication.
- Inside this folder, navigate to the WSL directory.
- Inside this directory, you will see a script called install.bat (Windows) or install.sh. Run this script and follow the prompts to install Encryption for Entire Net-Work on your system.
- Note that on Windows systems, install.bat requires a command line argument of either “all” or “toolkit”. The argument “toolkit” will install only the SSL Toolkit that can be used to generate test certificates. The argument “all” will first install the required SSL libraries in your Adabas Client environment, then will prompt the user whether the SSL Toolkit should be installed as well.
- The SSL Toolkit delivered is only for use on Windows. The supported Linux/Unix distributions of Encryption for Entire Net-Work deliver the SSL Toolkit as a .zip file that can be transferred to a Windows machine and extracted for use there.

- When the script completes, Encryption for Entire Net-Work is now installed. See [Activating Encryption for Entire Net-Work](#) for details on configuring your Entire Net-Work Server and Clients for SSL usage.

Uninstalling Encryption for Entire Net-Work

Encryption for Entire Net-Work is uninstalled simply by deleting the files installed by the installation script. These files can be found under the AdabasClient directory in your environment as follows:

- **Windows:** `AdabasClient\bin\sagoss15.dll` **and** `AdabasClient\bin32\sagoss15.dll`
- **UNIX/LINUX:** `AdabasClient/lib/libsagoss15.so` **and** `AdabasClient/lib_32/libsagoss15.so`.
- On Windows systems, if you have installed the SSL Toolkit, delete the entire SSLToolkit folder.

6 **Activating Encryption for Entire Net-Work**

■ Open Systems Activation	18
---------------------------------	----

This chapter describes the steps that must be completed to activate Encryption for Entire Net-Work.

Open Systems Activation

The following table lists the steps that must be completed to activate Encryption for Entire Net-Work on LUW. Click on a step number for more information.

Step	Description
1	Create or obtain certificates for encryption and authentication.
2	Deploy the certificates you have obtained.
3	Create the text file used to ensure random encryption (optional).
4	Alter the target definitions.

Step 1. Create or Obtain Certificates

Create or obtain the certificates you will need for encryption and authentication.

Various organizations, such as VeriSign, act as external certificate authorities for other companies and supply keys for authentication and encryption as requested by their clients. For Entire Net-Work, you can use an external certificate authority to provide your keys or, for testing only, you can use the open source SSL Toolkit, provided with Encryption for Entire Net-Work, to become your own certificate authority.

For more information about the open source SSL Toolkit, read [Using the SSL Toolkit](#), elsewhere in this guide.

To use an external organization to obtain your certificates, contact them for more information.

Step 2. Deploy the Certificates

Once you have created or obtained your certificates (Step 1), they must be deployed. When you obtain your certificates (regardless of whether you used an external certificate authority or the SSL Toolkit) you are supplied with the following files:

1. A public key certificate for your company or installation.
2. A private key for your company or installation.
3. A public key certificate for the certificate authority itself.
4. A password for decrypting the certificates (sometimes called a *pem pass phrase*).

These files must be deployed before they can be used. To deploy these files:

1. Transport the certificates and key files to the systems where they are to be used. You can use the *ftp* utility to do this. You can also copy and rename certificates and key files as required.
2. Make sure the location of the certificates and keys is clear on the systems where they are being used. If they are not in the current directory, identify their location using the appropriate SSL parameters and settings as described in [Access and Connection Definition Setup](#), elsewhere in this guide.

Step 3. Create the Text File Used to Ensure Random Encryption (Optional)

Optionally, create a text file member that contains at least 14 random characters. The random characters in this file will be used by the encryption routines, thus ensuring that encryption itself occurs in a random manner.



Note: A random file is not required in Windows environments, but is in some UNIX environments.

Make sure the location of the random file is clear on the systems where it is being used. If it is not in the current directory, identify its location using the appropriate RANDOM_FILE parameter as described in [Access and Connection Definition Setup](#), elsewhere in this guide.

Step 4. Alter the Target Definitions

To use Encryption for Entire Net-Work, the existing target definitions for your Adabas databases (on mainframe and open systems) must be updated to support secured communications. Each definition must be altered so that the protocol type "SSL" is specified in the access or connection definition and appropriate security parameters are specified. For more information on maintaining your target entries and on the security parameters, read [Access and Connection Definition Setup](#).

7

Using the SSL Toolkit

■ Gathering SSL Toolkit Information	22
■ Setting Up a Certificate Authority	25
■ Creating Certificates	26
■ Deploying Certificates	29

Secure Sockets Layer (SSL) is a standard protocol used to manage the security of message transmissions in an open communications network, such as the Internet. It uses TCP/IP for its physical communications. In addition, it uses public and private key encryption for both authentication and data encryption keys. These certificates are obtained from a certificate authority.



Note: The SSL Toolkit is a 32-bit application that runs in both 32-bit and 64-bit environments.

Various organizations, such as VeriSign, act as external certificate authorities for other companies and supply certificates for authentication and encryption as requested by their clients. You can use an external certificate authority to provide your certificates or, *for testing only*, you can use the SSL Toolkit, provided with Encryption for Entire Net-Work, to become your own certificate authority.

The SSL Toolkit allows you to create your own certificate authority (CA) and certificates for C code. It is available in Windows environments only.

➤ To use the SSL Toolkit:

- 1 Collect the information described in [Gathering SSL Toolkit Information](#), elsewhere in this chapter. This information is requested when running the SSL Toolkit.
- 2 At a command prompt, make the SSL Toolkit directory on your Windows machine the current directory.
- 3 Create a certificate authority for the Windows machine. For more information, read [Setting Up a Certificate Authority](#), elsewhere in this chapter.
- 4 Create the certificates you need. For more information, read [Creating Certificates](#), elsewhere in this chapter.
- 5 When the certificates you need have been created, deploy them on the system on which they are needed. For more information, read [Deploying Certificates](#), elsewhere in this chapter.
- 6 Update the appropriate target definitions in the Entire Net-Work Client, Kernel, and server target entries or in the Directory Server entries to support secure transmissions. For more information, read [Access and Connection Definition Setup](#).

Gathering SSL Toolkit Information

When you use the SSL Toolkit, it will prompt you for the information described in the following table. Use the following table to collect this information prior to using the SSL Toolkit. The order in which this information is requested varies by what you attempt to create: a certificate authority (CA) or a certificate and key. All of this information is not necessarily requested during SSL Toolkit processing.

Information Requested	Description	Used to Create
City or Town (Locality)	<p>The name of your city or town. If a default is provided, it is shown in brackets next to the prompt.</p> <p>This information is used as part of the distinguished name (DN) for a certificate or CA. The contents of the DN for each certificate must be unique; this means that at least one of the fields that comprise the DN for each certificate must be unique.</p>	<p>Certificate authority</p> <p>C certificates</p>
Common Name	<p>Your name or the name of your application. If a default is provided, it is shown in brackets next to the prompt. A maximum of 64 characters can be specified.</p> <p>If you are creating a server certificate, and clients will verify the server's name, this value should be the fully qualified hostname of the server. The verification of the server name is controlled by the parameter VERIFY=8 on the client access URL. See the section Access and Connection Definition Setup in this documentation for details on the VERIFY parameter.</p> <p>This information is used as part of the distinguished name (DN) for a certificate or CA. The contents of the DN for each certificate must be unique; this means that at least one of the fields that comprise the DN for each certificate must be unique.</p>	<p>Certificate authority</p> <p>C certificates</p>
Country Name	<p>A two-letter code for your country. If a default is provided, it is shown in brackets next to the prompt.</p> <p>This information is used as part of the distinguished name (DN) for a certificate or CA. The contents of the DN for each certificate must be unique; this means that at least one of the fields that comprise the DN for each certificate must be unique.</p>	<p>Certificate authority</p> <p>C certificates</p>
E-mail Address	<p>Your e-mail address. The default is "Security@YourCompany.com". A maximum of 40 characters can be specified.</p> <p>This information is used as part of the distinguished name (DN) for a certificate or CA. The contents of the DN for each certificate must be unique; this means that at least one of the fields that comprise the DN for each certificate must be unique.</p>	<p>Certificate authority</p> <p>C certificates</p>
Organization Unit	<p>The name of your department within the organization. If a default is provided, it is shown in brackets next to the prompt.</p> <p>This information is used as part of the distinguished name (DN) for a certificate or CA. The contents of the DN for each certificate must be unique; this means that at least one of the fields that comprise the DN for each certificate must be unique.</p>	<p>Certificate authority</p> <p>C certificates</p>

Information Requested	Description	Used to Create
Organization Name	<p>The name of your organization. If a default is provided, it is shown in brackets next to the prompt.</p> <p>This information is used as part of the distinguished name (DN) for a certificate or CA. The contents of the DN for each certificate must be unique; this means that at least one of the fields that comprise the DN for each certificate must be unique.</p>	Certificate authority
PEM Pass Phrase	<p>A Public Encryption Method (PEM) password phrase used by the certificate authority to sign certificates. This PEM password phrase is also requested when you create a certificate. The PEM password you use when setting up the certificate authority should be the same as the PEM password requested when creating a certificate.</p> <p>PEM passwords can be between 4 and 20 alphanumeric characters long, including blanks. They are case-sensitive.</p>	Certificate authority C certificates
State or Province	<p>The name of your state or province. If a default is provided, it is shown in brackets next to the prompt.</p> <p>This information is used as part of the distinguished name (DN) for a certificate or CA. The contents of the DN for each certificate must be unique; this means that at least one of the fields that comprise the DN for each certificate must be unique.</p>	Certificate authority C certificates
Optional Challenge Password	<p>An optional password you can request when you create a C certificate. This password must be different from the PEM password and must be different for each certificate.</p> <p>Challenge passwords can be between 4 and 20 alphanumeric characters long.</p>	C certificates
Optional Company Name	An optional company name	C certificates

You can set defaults for some of these values in the *genca.template* file located in the SSL Toolkit directory. However, the defaults you specify in this file only pertain to setting up a certificate authority or generating C certificates.



Caution: Before you change the *genca.template* file, be sure to save a copy of the original for later reference.

Setting Up a Certificate Authority

Only one certificate authority can be set up on a single Windows machine. If you run the procedure described in this document more than once on the same machine, the new certificate authority overwrites the old one.

➤ **To set up a certificate authority:**

- 1 At a DOS command prompt, make the SSL Toolkit directory on your Windows machine the current directory. Then enter the following command:

```
makeca
```

The certificate authority setup process is started. You are prompted to answer a number of questions, as described in the remaining steps.

- 2 At the PEM password phrase prompt, enter the PEM password phrase you want to use for this certificate authority. The password phrase is used by the certificate authority to sign C certificates. For more information about PEM password phrases, read [Gathering SSL Toolkit Information](#), earlier in this section.
- 3 When you are prompted to repeat the PEM password phrase, enter it again exactly as you did in Step 2. Remember that PEM password phrases are case-sensitive.

The PEM password phrase you enter in this step is compared and verified using the one PEM password phrase you entered in Step 2. If a mismatch occurs, you are prompted to enter the original PEM password phrase (Step 2) and to verify it (Step 3) again.

- 4 At the country prompt, enter a two-letter country code you want used when creating a distinguished name (DN) for use by the certificate authority. If you press `Enter` without specifying a value, the default shown in brackets is used.
- 5 At the state or province prompt, enter the name of the state or province you want used for the distinguished name (DN) for the certificate authority. If you press `Enter` without specifying a value, the default shown in brackets is used.
- 6 At the city or town prompt, enter the name of the city or town you want used for the distinguished name (DN) for the certificate authority. If you press `Enter` without specifying a value, the default shown in brackets is used.
- 7 At the organizational name prompt, enter the name of your organization. This name is used for the distinguished name (DN) for the certificate authority. If you press `Enter` without specifying a value, the default shown in brackets is used.
- 8 At the organization unit prompt, enter the name of your department within the organization. This name is used for the distinguished name (DN) for the certificate authority. If you press `Enter` without specifying a value, the default shown in brackets is used.

- 9 At the common name prompt, enter your name or the name of your application. This name is used for the distinguished name (DN) for the certificate authority.
- 10 At the e-mail address prompt, enter the e-mail address you want used for the distinguished name (DN) for the certificate authority. If you press **Enter** without specifying a value, the default shown in brackets is used.

The certificate authority is set up. You can now use it to create certificates.

When you complete these steps, three new subdirectories are added in the SSL Toolkit directory: *cacerts*, *certs*, and *newcerts*.

Subdirectory Name	Use
<i>cacerts</i>	Stores certificate authority files.
<i>certs</i>	Stores certificate files, signed or unsigned.
<i>newcerts</i>	For internal use only. Used during the SSL Toolkit certificate creation process.

In addition, the following files are created in the *cacerts* subdirectory:

- *cacert.mf*: A CA certificate that can be used on mainframe systems.
- *cacert.pem*: A CA certificate that can be used on LUW systems.
- *cakey.pem*: A CA key file that can be used on LUW systems.

Creating Certificates

Once you have set up a certificate authority, you can create C code certificates and their associated keys using the SSL Toolkit.

➤ To create C code certificates:

- 1 At a command prompt, make the SSL Toolkit directory on your Windows machine the current directory. Then enter the following command:

```
makeccerts ↵
```

You will be prompted for a certificate name which will be used as a prefix for your generated certificates. The default is “myapp”.

The C certificate and key creation process is started. You are prompted to answer a number of questions, as described in the remaining steps.

- 2 At the PEM password phrase prompt, enter the PEM password phrase you want to use. This should be the same PEM password phrase you specified when you set up the certificate authority (CA).

For more information about PEM password phrases, read [Gathering SSL Toolkit Information](#), earlier in this section.

- 3 When you are prompted to repeat the PEM password phrase, enter it again exactly as you did in Step 2. Remember that PEM password phrases are case-sensitive.

The PEM password phrase you enter in this step is compared and verified using the one PEM password phrase you entered in Step 2. If a mismatch occurs, you are prompted to enter the original PEM password phrase (Step 2) and to verify it (Step 3) again.

- 4 At the country prompt, enter a two-letter country code you want used when creating a distinguished name (DN) for use by the certificate and key. If you press `Enter` without specifying a value, the default shown in brackets is used.
- 5 At the state or province prompt, enter the name of the state or province you want used for the distinguished name (DN) for the certificate and key. If you press `Enter` without specifying a value, the default shown in brackets is used.
- 6 At the city or town prompt, enter the name of the city or town you want used for the distinguished name (DN) for the certificate and key. If you press `Enter` without specifying a value, the default shown in brackets is used.
- 7 At the organizational name prompt, enter the name of your organization. This name is used for the distinguished name (DN) for the certificate. If you press `Enter` without specifying a value, the default shown in brackets is used.
- 8 At the organization unit prompt, enter the name of your department within the organization. This name is used for the distinguished name (DN) for the certificate. If you press `Enter` without specifying a value, the default shown in brackets is used.
- 9 At the common name prompt, enter your name or the name of your application. This name is used for the distinguished name (DN) for the certificate.
- 10 At the e-mail address prompt, enter the e-mail address you want used for the distinguished name (DN) for the certificate. If you press `Enter` without specifying a value, the default shown in brackets is used.
- 11 Optionally, at the challenge password prompt, enter the challenge password you want used for this certificate.

For more information about challenge passwords, read [Gathering SSL Toolkit Information](#), earlier in this section.

- 12 Optionally, enter your company name at the optional company name prompt.

The basic information for the certificate is complete. The process to sign the certificate is started.

- 13 At the PEM password phrase prompt, enter the PEM password phrase you selected for the certificate authority (CA) when you set it up.

If you enter the incorrect CA PEM password phrase, the certificate creation process aborts. Otherwise, the process to sign the certificate continues.

- 14 You must enter "y" at the **Sign the certificate?** prompt. If you do not, the certificate will not work.
- 15 Enter "y" at the commit prompt. If you do not, the certificate will not work.

The process to sign the C certificate completes. The certificate is certified.

- 16 At this point, you will be prompted whether you want to create mainframe (EBCDIC) certificates for use with Entire Net-Work versions prior to v6.5.2. Answer "Y" to this prompt only if you are using an older version of Entire Net-Work with the Encryption for Entire Net-Work option on the mainframe.

Once created, these certificates can be transferred to the mainframe host using the supplied utility `ftpcerts.bat` in the `SSLToolkit` directory. Note that in order for this to work correctly, the mainframe target.pds must be: `DSORG=PO, RECFM=FB and LRECL=251`.

- 17 At the next step, you will be prompted whether you want to create a pkcs12 file. A pkcs12 file is a single certificate containing both the certificate and key.

It may be useful if you are creating a certificate for use on the mainframe with Entire Net-Work v6.5.2 or higher using the IBM AT-TLS implementation of encryption. Answering the prompt with "Y" will create the pkcs12 file, entering "N" will exit the utility.

The following files with names in the following formats are created in the `/certs` directory:

- (Optionally) `<prefix>cert.mf`: Certificate file that can be used on mainframe systems with Entire Net-Work prior to the v6.5.2 release.
- `<prefix>cert.pem`: Certificate file that can be used on open systems.
- (Optionally) `<prefix>key.mf`: Key file that can be used on mainframe systems with Entire Net-Work prior to the v6.5.2 release.
- `<prefix>key.pem`: Key file that can be used on open systems.
- `<prefix>Certreq.pem`: This file is used internally by the SSL Toolkit for C certificate processing.
- (Optionally) `<prefix>.p12`: This is the combined key and certificate in a pkcs12 file.

where `<prefix>` is the prefix you specified when you ran the `makeccerts` program in Step 1. For example, if you used the default prefix "myapp", the following files would be created:

- `myappcert.mf` (Optionally)
- `myappcert.pem`
- `myappkey.mf` (Optionally)
- `myappkey.pem`
- `myappCertreq.pem`
- `myapp.p12` (Optionally)

Deploying Certificates

➤ **To deploy certificates and their associated keys:**

- 1 Transport the certificates and key files to the systems where they are to be used. You can use the *ftp* utility to do this. You can also copy and rename certificates and key files as required.
- 2 Make sure the location of the certificates and keys is clear on the systems where they are being used. If they are not in the current directory, identify their location using the appropriate SSL parameters and settings as described in [Access and Connection Definition Setup](#), elsewhere in this guide.

8

Access and Connection Definition Setup

■ Maintaining Target Definitions	32
■ Security Parameters	32

To use Encryption for Entire Net-Work, the existing target definitions for your Adabas databases (on mainframe and open systems) must be updated to support secured communications. Each definition must be altered so that the protocol type "SSL" is specified in the access or connection definition and appropriate security parameters are specified.

These definitions are altered via their Adabas Directory Server entries or the Entire Net-Work Client, Kernel, and server access or connection definitions in the System Management Hub.

Maintaining Target Definitions

The target definitions for each database that will be accessed through a secure connection must be altered to specify "SSL" as the protocol type. The format of a secured target entry is:

```
SSL://host:port[?parm=value]&parm=value...
```

In addition to specifying appropriate host and port numbers, you must change the communication protocol type to "SSL" (as shown) and specify any security parameters that may be required. To determine which specific qualifiers and parameters should be supplied for different security situations, read [Security Scenarios](#), elsewhere in this guide. The possible parameters are documented in [Security Parameters](#), in this section.

The port number must match the setting on the SSL line driver SERVERID parameter. If one line driver will serve multiple databases, an entry for each database is required, but these entries would all specify the same port number.

Security Parameters

The following table describes the security parameters that can be used to support secured transmissions with Entire Net-Work.

Parameter	Description	Server Requirements	Client Requirements
CAFILE	The name of the file containing the trusted certificate authority's (CA) certificates. The certificate of the CA that signed an inbound certificate must reside in this file or in the CAPATH directory. It is a good idea to store this file on a protected network drive. If a specified certificate is corrupt, secured transmissions will fail.	Required only for client authentication.	Required only for server authentication.

Parameter	Description	Server Requirements	Client Requirements
	<p>If a certificate is received that is signed by a CA other than the CA specified by CAFILE, then the CAPATH is searched.</p> <p>Note: The file name specified may include the path information, unless a value for parameter CAPATH is specified.</p>		
CAPATH	<p>The location (path) where the CAFILE resides or where additional certificates of certificate authorities (CA) reside.</p> <p>Note: The hash values of the names of the CA certificate files should be used in this location. Hash names are generated by the OpenSSL tool.</p> <p>If parameter CAFILE includes location information, the value of CAPATH should be ".", which is also the CAPATH default.</p>	Required only for client authentication.	Required only for server authentication.
CERT_FILE	<p>The file containing the participant's digital certificate. The certificate file may contain the participant's private key. It is a good idea to store this file on a protected network drive.</p> <p>Note: The file name specified may include the path information. This is useful if the certificate is not in the current directory.</p>	Always required.	Required only for client authentication.
CERT_PSSWD	<p>The password for extracting information from the certificate file specified in the CERT_FILE parameter. It is a good idea to store this file on a protected network drive.</p> <p>Note: You can specify a fully qualified file name for this parameter. In this case, the file name you provide must contain the password.</p>	Always required.	Required only for client authentication.
KEY_FILE	<p>The name of the file containing the server's private key. This parameter must be specified if the private key is kept separate from the certificate file. It is a good idea to store this file on a protected network drive.</p> <p>Note: The file name specified may include the path information. This is useful if the certificate is not in the current directory.</p>	Always required.	Required only for client authentication.
RANDOM_FILE	Identifies a text file that contains at least 14 random characters. The random characters in	Optional	Optional

Parameter	Description	Server Requirements	Client Requirements
	<p>this file are used by the encryption routines to ensure that encryption itself occurs in a random manner.</p> <p>Some platforms (such as Solaris) require the use of a random file.</p>		
VERIFY	<p>The level of certificate verification to perform. Valid values are:</p> <ul style="list-style-type: none"> ■ 0 (No peer verification occurs.) ■ 1 (The application requests that the peer certificate be verified.) ■ 2 (The application requests that the peer certificate be verified. A fatal condition occurs if there is no certificate.) ■ 4 (The application requests that the peer certificate be verified only once.) ■ 8 (The application requests that the issuer name is checked against the host name.) <p>Values 1, 2, and 4 can be specified in combination. For example, if you want to specify both 1 and 2, you would add them and set the VERIFY parameter to "3".</p> <p>Note: This parameter must be set to "3" if you are performing client authentication.</p>	<p>Use VERIFY=1 to request a client certificate and verify that it is sent.</p> <p>Use VERIFY=2 to force the sending of a client certificate.</p> <p>Use VERIFY=4 to limit the client certificate request to a single occurrence.</p> <p>VERIFY=8 is not valid for server processing.</p>	<p>Use VERIFY=0 (the C client default) to request a certificate but proceed even if certificate errors are found.</p> <p>Use VERIFY=1 to validate the server certificate.</p> <p>VERIFY=2 is not valid for client processing.</p> <p>VERIFY=4 is not valid for client processing.</p> <p>Use VERIFY=8 to validate that the common name of the received certificate matches the host name specified in the target entry.</p>

9

Sample Security Scenarios

■ Simple Encryption	36
■ Other Authentication Scenarios	37

This chapter describes various sample SSL scenarios using Encryption for Entire Net-Work.

For each scenario described in this section, the client-side alterations you need to make to your Kernel and Entire Net-Work Client access and connection definitions are given.

The scenarios that are described are:

Simple Encryption

➤ To configure simple encryption for an Entire Net-Work Client:

- 1 If the database access URL for the target database is written to the Directory Server by an LUW Entire Net-Work configured for SSL access, no configuration for the client is required. The client will load SSL once it finds the protocol SSL in the URL and communication will be encrypted.
- 2 If the database access is manually configured in the Entire Net-Work Client configuration, set the target access entry to "SSL" using Adabas Manager by using the protocol drop-down menu and save the entry.

➤ To configure simple encryption for an Entire Net-Work Server:

- 1 Access the Entire Net-Work Server Kerne lAccess definitions in the Adabas Manager.
- 2 For each Server Access definition that needs to support SSL, verify that the Protocol type for the entry is SSL and that appropriate port numbers are specified.
- 3 Edit the definition and specify valid values for the SSL CERT_FILE, KEY_FILE, and CERT_PSSWD parameters in the **Additional Parameters** field. In the following example, *xxcert.pem* is the certificate file, *xxkey.pem* is the certificate key file, and "pempswd" is the Public Encryption Method (PEM) password:

```
&CERT_FILE=xxcert.pem&CERT_PSSWD=pempswd&KEY_FILE=xxkey.pem
```

- 4 Save the definition.

Other Authentication Scenarios



Note: In all the following scenarios, the Client Authentication parameters are best configured manually by first configuring an Adabas Client Access definition to the database target. Use Adabas Manager to add the entry and edit the entry as described in the following sections.

- [Client-Only Authentication](#)
- [Server-Only Authentication](#)
- [Client and Server Authentication](#)
- [Authentication with Certificates Elsewhere](#)
- [Authentication with a Hidden Password](#)

Client-Only Authentication

➤ To perform client-only authentication for an Entire Net-Work Client:

- 1 Access the Entire Net-Work Client access definition to Adabas databases in Adabas Manager.
- 2 Ensure the Protocol field in the Host address line is SSL.
- 3 Specify values for the CERT_FILE, KEY_FILE, and CERT_PSSWD parameters in the **Additional Parameters** field. For example:

```
&CERT_FILE=testcert.pem&KEY_FILE=testkey.pem&CERT_PSSWD=pempswd
```

- 4 Save the definition.

➤ To perform client-only authentication for an Entire Net-Work Server:

- 1 Access the Entire Net-Work Server Kernel Access definitions in Adabas Manager.
- 2 For each Client Access definition that needs to support SSL client-only authentication, verify that the Protocol type is SSL and that an appropriate port number is specified.
- 3 In the Additional Parameters field for the entry, specify valid values for the CAFILE, CAPATH, CERT_FILE, KEY_FILE, CERT_PSSWD, and VERIFY parameters in the **Additional Parameters** field. The VERIFY parameter must be set to "3" for client authentication.
- 4 Save the definition.

Server-Only Authentication

➤ To perform server-only authentication for an Entire Net-Work Client:

- 1 Access the Entire Net-Work Client Access definition to Adabas databases in Adabas Manager. Find the appropriate Connection definition.
- 2 Ensure the Protocol type is SSL..
- 3 Specify values for the CAFILE, CAPATH, and VERIFY parameters in the **Additional Parameters** field. For example:

```
&CAFILE=cacert.pem&CAPATH=path&VERIFY=1
```

- 4 Save the definition.

➤ To perform server-only authentication for a Kernel to Kernel connection:

- 1 Access the Kernel Access definitions in Adabas Manager. Find the appropriate Connection definition.
- 2 Ensure the Protocol type is SSL.
- 3 Specify values for the CAFILE, CAPATH, and VERIFY parameters in the **Additional Parameters** field. For example:

```
&CAFILE=cacert.pem&CAPATH=path&VERIFY=1
```

- 4 Save the definition.

➤ To perform server-only authentication for an Entire Net-Work Server:

- 1 Access the Entire Net-Work Server Kernel Access definitions in Adabas Manager.
- 2 For each Client Access definition that needs to support SSL server-only authentication, verify that Protocol type is SSL and that an appropriate port number is specified.
- 3 In the Additional Parameters field for the entry, specify valid values for the CERT_FILE, KEY_FILE, and CERT_PSSWD parameters in the **Additional Parameters** field. In the following example, where *xxcert.pem* is the certificate file, *xxkey.pem* is the certificate key file, and "pempswd" is the Public Encryption Method (PEM) password:

```
&CERT_FILE=xxcert.pem&CERT_PSSWD=pempswd&KEY_FILE=xxkey.pem
```

- 4 Save the definition.

Client and Server Authentication

➤ To perform client and server authentication for an Entire Net-Work Client:

- 1 Access the Entire Net-Work Client Access definitions in Adabas Manager.
- 2 Ensure the Protocol field in the Host address line is SSL.
- 3 Specify values for the CAFILE, CAPATH, CERT_FILE, KEY_FILE, CERT_PSSWD, and VERIFY parameters in the **Additional Parameters** field. For example:

```
&CAFILE=cacert.pem&CAPATH=path&CERT_FILE=xxcert.pem&KEY_FILE=xxkey.pem&CERT_PSSWD=pempswd&VERIFY=1
```

- 4

```
&CERT_FILE=testcert.pem&KEY_FILE=testkey.pem&CERT_PSSWD=pempswd
```

- 5 Save the definition.

➤ To perform client and server authentication for an Entire Net-Work Server:

- 1 Access the Entire Net-Work Server Kernel Access definitions in Adabas Manager.
- 2 For each Access definition that needs to support SSL client and server authentication, verify that either the Protocol type is SSL and that appropriate port numbers are specified.
- 3 For Server Access, specify valid values for the CERT_FILE, KEY_FILE, and CERT_PSSWD parameters in the **Additional Parameters** field. In the following example, *xxcert.pem* is the certificate file, *xxkey.pem* is the certificate key file, and "pempswd" is the Public Encryption Method (PEM) password:

```
&CERT_FILE=xxcert.pem&CERT_PSSWD=pempswd&KEY_FILE=xxkey.pem
```

- 4 For **E-business SSL Client Access**, specify valid values for the CAFILE, CAPATH, CERT_FILE, KEY_FILE, CERT_PSSWD, and VERIFY parameters in the **Additional Parameters** field. The VERIFY parameter must be set to "3" for client authentication.
- 5 Save the definition.

Authentication with Certificates Elsewhere

➤ To perform client or server authentication from a client or a server when the certificates and certificate authorities are not in the current directory:

- Complete the authentication steps described in other scenarios in this section, but specify the path to the certificate authority and certificate files in the `CAFILE`, `CERT_FILE`, and `KEY_FILE` parameters.



Note: If parameter `CAFILE` includes path information, the value of `CAPATH` should be `"."`.

Authentication with a Hidden Password

➤ To perform client or server authentication from a client or a server without specifying the Public Encryption Method password directly in the URL:

- Complete the authentication steps described in other scenarios in this section, but specify the fully-qualified file name of a file that contains the password in the `CERT_PSSWD` parameter. For example:

```
&CAFILE=cacert.pem&CAPATH=path&CERT_FILE=xcert.pem&KEY_FILE=xkey.pem&CERT_PSSWD=FILE://C:/certs/certpswd.txt&VERIFY=3
```

Index

A

- access definitions, SSL setup, 31
- authentication
 - certificates elsewhere, 40
 - client, 37
 - client and server, 39
 - defined, 10
 - hidden password, 40
 - server, 38

C

- CA (see certificate authority (CA))
- CAFILE parameter, 32
- CAPATH parameter, 33
- CERT_FILE parameter, 33
- CERT_PSSWD parameter, 33
- certificate authority (CA)
 - defined, 10
 - not in current directory, 40
 - setting up, 25
- certificates
 - creating, 26
 - defined, 10
 - deploy, 18
 - deploying, 29
 - not in current directory, 40
 - obtaining, 18
- client authentication, 37, 39
- connection definitions
 - security parameters, 32
 - SSL setup, 31
- creating certificates, 26

D

- decryption, defined, 10
- defining a certificate authority, 25
- deploying certificates, 29
- digital signatures, defined, 10

E

- Empower
 - platform support, 8
- encryption
 - defined, 10

- Encryption for Entire Net-Work, 15
 - access and connection definition setup, 31
 - activating, 17
 - end-of-support dates, 8
 - enhancements, 8
 - installing and uninstalling, 13
 - installing on open systems, 14
 - open systems activation steps, 18
 - overview, 9
 - prerequisites, 8
 - release notes, 7
 - scenarios, 35
 - security parameters, 32
 - supported platforms, 8
 - target definition setup, 32
 - using,
 - using the SSL Toolkit, 21

H

- hidden password, 40

I

- installing Encryption for Entire Net-Work, 13
 - on LUW, 14
 - on open systems, 14

K

- KEY_FILE parameter, 33

L

- LUW
 - installation Encryption for Entire Net-Work, 14

M

- Microsoft Windows support, 8

O

- open systems Encryption for Entire Net-Work activation
 - alter target definitions, 19
 - create random file, 19
 - deploy certificates, 18
 - obtain certificates, 18
 - steps, 18

open systems Encryption for Entire Net-Work installation, 14
operating system coverage, 8

P

platform support, 8
product support
 supported platforms, 8

R

random file
 creating, 19
RANDOM_FILE parameter, 33
release notes, 7
requirements
 operating system coverage, 8

S

scenarios, 35
Secure Sockets Layer (SSL)
 access and connection definition setup, 31
 scenarios, 35
 security parameters, 32
 target definitions, 32
 using the SSL Toolkit, 21
security parameters, 32
security scenarios, 35
server authentication, 38-39
setting up a certificate authority, 25
SSL Toolkit
 creating certificates, 26
 deploying certificates, 29
 gathering information for, 22
 overview, 21-22
 setting up certificate authority, 25
support
 platforms supported, 8
supported operating systems, 8

T

target definitions
 altering, 19
 security parameters, 32
 syntax for SSL, 32

U

uninstalling Encryption for Entire Net-Work, 15

V

VERIFY parameter, 34