software AG

Client Administration Guide

Using Encryption for Entire Net-Work

Version 6.5.1

October 2019

ADABAS & NATURAL

Table of Contents

Preface

The 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. This support is provided for Entire Net-Work on both mainframe and open systems; so Encryption for Entire Net-Work is a product that is installed in both environments.

Encryption for Entire Net-Work is packaged as one complete product, spanning both mainframe and open systems. The mainframe and open systems portions of the product are delivered on different media and have separate installations and activation.

This document describes the use of Encryption for Entire Net-Work on both mainframe and open systems.

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

Release Notes	Describes the changes, enhancements, prerequisites, migration considerations, and documentation for this release of Encryption for Entire Net-Work.
Concepts	Provides a high-level introduction to Encryption for Entire Net-Work.
Installing Encryption for Entire Net-Work	Describes the prerequisites for Encryption for Entire Net-Work and explains how to install it on the mainframe and on open systems.
Activating Encryption for Entire Net-Work	Describes the steps necessary to activate Encryption for Entire Net-Work on mainframe and open systems.
Using the Mainframe SSL/SSLI Line Driver's	Explains how to use the mainframe SSL/SSLI line driver's, the primary Encryption for Entire Net-Work component on mainframe systems.
Using the SSL Toolkit	Explains how to use the SSL Toolkit on open systems to create certificates for testing your use of Encryption for Entire Net-Work.
Access and Connection Definition Setup	Describes the target definition access and connection definition syntax required for Encryption for Entire Net-Work support.
Security Scenarios	Provides mainframe and open systems examples of Encryption for Entire Net-Work use.
Making SSL Certificates and Keys	Describes the procedures to generate your own certificates and keys.

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

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *folder.subfolder.service*, APIs, Java classes, methods, properties.
Italic	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

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

In addition, you can also access the cloud product documentation via **https://www.softwareag.cloud**. Navigate to the desired product and then, depending on your solution, go to "Developer Center", "User Center" or "Documentation".

Product Training

You can find helpful product training material on our Learning Portal at **https://knowledge.softwareag.com**.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

▪ Browse through our vast knowledge base.

▪ Ask questions and find answers in our discussion forums.

▪ Get the latest Software AG news and announcements.

▪ Explore our communities.

▪ Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

▪ Download products, updates and fixes.

▪ Search the Knowledge Center for technical information and tips.

▪ Subscribe to early warnings and critical alerts.

▪ Open and update support incidents.

▪ Add product feature requests.

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

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

Enhancements

Encryption between Mainframe Net-Work nodes on z/OS is now possible using the new SSLI line driver. Encrypted connections form mainframe Net-Work nodes on LUW nodes is still provided via the SSL line driver.

In LUW (Linux/Unix/Windows) environments prior to Entire Net-Work Version 7.6 SP1 and Entire Net-Work Client Version 1.6 SP1, Encryption for Entire-Net-Work was included in the base delivery and its use restricted by a license parameter. As of SP1 of both of these products, the SSL libraries have been removed, as has the licensing parameter. Use of Encryption for Entire Net-Work for both client and server installations now requires an add-on called "Encryption for Entire Network Version 1.3 SP1" be installed on any machines where encrypted communication via SSL is required. See **Installing Encryption for Entire Net-Work** for details on how to install Encryption for Entire Net-Work on LUW systems.

Encryption for Entire Net-Work also includes the open source SSL Toolkit that you can use to create certificates for testing.

Encryption is also available from Mainframe Net-Work to Mainframe Net-Work with the SSLI Driver, in addition to the connection to Open Systems Net-Work, via the SSL Driver.

> **Note:** Encryption for Entire Net-Work on open systems remains uncoupled from your Entire Net-Work open systems (Entire Net-Work Server) and Entire Net-Work Client installations at this time. If you uninstall these software packages and then reinstall them at a different location, you will need to uninstall and reinstall Encryption for Entire Net-Work separately.

> **Note:** The version number of the Encryption for Entire Network is the same as the Net-Work version that it runs on on mainframe. This is currently version 6.3.2 .

Prerequisites

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

- If you intend to use Encryption for Entire Net-Work on mainframe systems only, Entire Net-Work 6.3.2 (or later) must be installed. Also required is the APS library version 3.3.1 at update level 20 or later. Contact your Software AG support representative for assistance. All prerequisites of these products must also be met.

- If you intend to use Encryption for Entire Net-Work to provide secure message transmissions between mainframe and open systems, Entire Net-Work 6.3.2 (or later), APS library version

3.3.1 at update level 20 or later, and Entire Net-Work 7.4 (or later) or Entire Net-Work Client 1.5 (or later) must be installed. All prerequisites of these products must also be met.

- If Encryption for Entire Net-Work is to be used to provide secure message transmissions, the target definitions for each database accessed through the secure TCP/IP connection must be altered. These definitions are modified in their Adabas Directory Server entries or in the Entire Net-Work Client, Kernel, and server target entries in the System Management Hub. Descriptions of the target entries are provided in *Access and Connection Definition Setup*, elsewhere in this section.

- Encryption for Entire Net-Work on open systems is supported in both 32-bit and 64-bit environments. The SSL Toolkit is a 32-bit application that runs in both 32-bit and 64-bit environments.

Supported Platforms

Encryption for Entire Net-Work 6.3.2 is currently only available on z/OS. It would be possible to deliver in on z/VSE and BS2000 if there is a requirement. The open source SSL Toolkit is only available in Windows environments. However, the Entire Net-Work open systems OpenSSL support code is no longer part of the Encryption for Entire Net-Work installation; it is included in the Entire Net-Work 7.5 open systems installation and is available in both Windows and UNIX environments.

Before attempting to install Encryption for Entire Net-Work, ensure that the host operating system is at the minimum required level. For information on the platform versions supported by Software AG products, access the Software AG web site at *http://www.softwareag.com/corporate/products/bis/platforms/default.asp*.

Software AG generally provides support for the operating system versions supported by their respective manufacturers; when an operating system provider stops supporting a version of an operating system, Software AG will stop supporting that operating system version.

Before attempting to install this product, ensure that your host operating system is at the minimum required level. For information on the platform versions supported by Software AG products, 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 the supported platforms and releases for any Software AG product.

Use the fields on this application to filter its results. When you click the **Find** button, a list of the supported Software AG products that meet the filter requirements is shown. You can clear all filter selections using the **Clear** button.

> **Note:** Although it may be technically possible to run a new version of this product on an old operating system, Software AG cannot continue to support operating system versions that are no longer supported by the system's provider. If you have questions about support, or if you plan to install this product on a release, version, or type of operating system other than one listed on the Product Version Availability screen described above, consult Software AG technical support to determine whether support is possible, and under what circumstances.

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

Encryption for Entire Net-Work provides support for the Secure Sockets Layer (SSL) to manage the security of message transmissions. This support is provided for Entire Net-Work on mainframe and LUW operating systems.

On mainframe systems, only z/OS support is provided at this time. Mainframe support is provided using a new SSLI line driver for mainframe to mainframe support, and via the SSL Driver for support between mainframe and LUW operating systems which is distributed with Encryption for Entire Net-Work. On LUW operating systems, SSL Toolkit support is provided only in 32-bit Windows environments. LUW operating systems support is provided through Software AG's implementation of OpenSSL.

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
- Encryption

SSL uses TCP/IP for its physical communications. In addition, SSL uses public and private key encryption for both authentication and data encryption keys. These keys are obtained from a certificate authority, as described in *Making SSL Certificates and Keys*.

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.

The following diagram depicts how certificates are used during authentication.
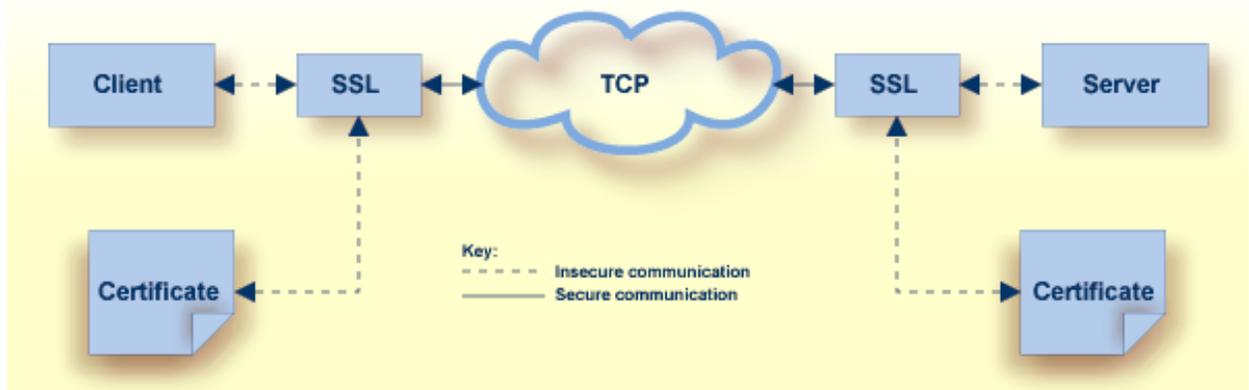
The following diagram depicts how certificates are used during data encryption.
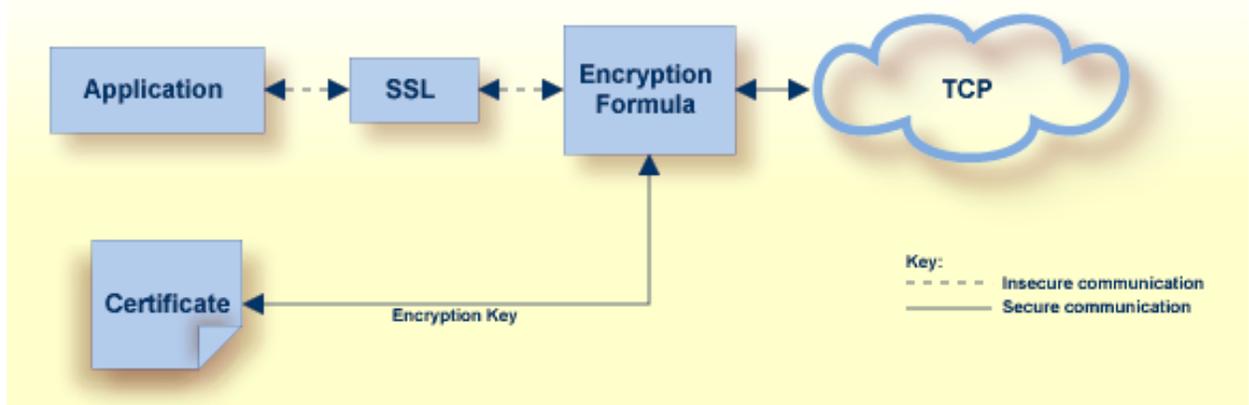
Various organizations, such as VeriSign, act as external certificate authorities for other companies and supply certificates for authentication and encryption as requested by their clients. For Entire Net-Work, you can use an external certificate authority to provide your certificates or, *for testing*

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

Before you install Encryption for Entire Net-Work, be sure you have met the requirements described in *Prerequisites*, elsewhere in this guide.

Mainframe Installation

On mainframe systems, the installation of Encryption for Entire Net-Work involves installing the software for the SSL/SSLI line driver's. This runs with the Entire Net-Work WCP V6.3.2 library. This section describes the steps required for this installation on z/OS platforms.

- Contents of the Release Tape
- Step 1. Unload the Entire Net-Work Libraries
- Step 2. Alter the Entire Net-Work Startup Job
- Step 3. Add the SSL DRIVER and LINK Statements
- Step 4. Configure Client Information
- Step 5. Start Entire Net-Work and Verify the Installation

Contents of the Release Tape

The following table describes the libraries included on the release tape when you purchase Encryption for Entire Net-Work. Once you have unloaded the libraries from the tape, you can change these names as required by your site.

Library Name	Description
APS*vrs*.LD*nn*	One or more Software AG internal libraries. The *vrs* in the library name represents the *version* of the internal library code, which is not necessarily the same as the version of Entire Net-Work
WSL*vrs*.LOAD	The z/OS load library for Encryption for Entire Net-Work. The *vrs* in the library name represents the *version* of Encryption for Entire Net-Work, which is not necessarily the same as the version of Entire Net-Work.

Step 1. Unload the Entire Net-Work Libraries

If not already performed, install the Entire Net-Work mainframe and Entire Net-Work TCP/IP Option libraries, using the procedure for your operating system environment. Then unload the SSL/SSLI line driver components from the installation tape as follows:

Platform	Installation Procedure
z/OS	IEBCOPY to restore the required data sets. Refer to the *Report of Tape Creation* for the correct data set sequence numbers and names.

Step 2. Alter the Entire Net-Work Startup Job

Make the following changes to the Entire Net-Work startup JCL.

Platform	Startup Job Change
z/OS	A sample startup JCL member called JCLNET is provided in the WCP source libraries. Add the SSL/SSLI line driver load library to the STEPLIB concatenation.

Step 3. Add the SSL DRIVER and LINK Statements

Use the existing Entire Net-Work configuration to create the necessary SSL/SSLI DRIVER and LINK statements for your environment in the Entire Net-Work DDKARTE input file.

> **Note:** If you are installing the SSL line driver the value of the SSL or SSLI DRIVER's API parameter must be "OES". The SSL format is as the TCPX Driver, the SSLI as the TCPI Driver in the WCP example jobs.

Step 4. Configure Client Information

For an LUW client to correctly send the database request to the Entire Net-Work node where the database is located, Adabas Directory Server entries must be added for each database. These entries tell the client application where the server (Entire Net-Work) is located and which databases it serves. A Directory Server access entry must be added for each database that the client will call via the security option of the SSL line driver.

For more information, read *Access and Connection Definition Setup*, elsewhere in this section.

Step 5. Start Entire Net-Work and Verify the Installation

Because of the many possible variations of the Entire Net-Work, Adabas, and applications topology, Software AG does not provide standard installation verification procedures. However, the following procedure is suggested for verifying the SSL/SSLI line driver installations:

1. Start the Entire Net-Work system and make connections to each link defined to the system.

2. Test the connections and verify that the links can be established from either side by connecting and disconnecting the links several times from each node. While the links are connected, issue the Entire Net-Work operator command DISPLAY TARGET to display the targets and the nodes on which they are located.

3. Test your applications running across Entire Net-Work. At first, run one application at a time, and then verify the results.

4. For the final verification test, run a load test through the network (that is, multiple users on each node accessing data on the partner node).

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:

```
WSLv1.3_SP1<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 v7.6_SP1 or Entire Net-Work Client v1.3_SP1 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.

- 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

The SSL Toolkit 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\sagossl5.dll` and `AdabasClient\bin32\sagossl5.dll`

- UNIX/LINUX: `AdabasClient/lib/libsagossl5.so` and `AdabasClient/lib_32/libsagossl5.so`.

- On Windows systems, if you have installed the SSL Toolkit, delete the entire SSLToolkit folder.

6 Activating Encryption for Entire Net-Work

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

Mainframe Activation

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

Step	Description
1	Create or obtain certificates for encryption and authentication.
2	Make Entire Net-Work library changes.
3	Deploy the certificates you have obtained.
4	Create the text file used to ensure random encryption.
5	Verify the parameters in the SYSPARMS member.
6	Alter the Entire Net-Work startup JCL.
7	Add the SSL DRIVER and LINK statements to the Entire Net-Work startup JCL.
8	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* or make your own Certificates. See *Making SSL Certificates and Keys* for further information.

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

> **Note:** The certificates must have EBCDIC encoding and a record length of 251 bytes.

Step 2. Make Entire Net-Work Library Changes

Define the following data sets. These data sets are required for Encryption for Entire Net-Work:

- NETWRK.*vrs*.SAGSSL.CERTS, where *vrs* represents the version number of Encryption for Entire Net-Work.

  This data set will store the certificates and keys provided by the certificate authority. It must be defined with the following attributes: `DSORG=PO`, `RECFM=FB`, `LRECL=251`, and `BLKSZ=6024`. It should also be write and read-protected by your company's security subsystem, ideally so only Entire Net-Work can access it.

- NETWRK.*vrs*.SAGSSL.RANDOM, where *vrs* represents the version number of Encryption for Entire Net-Work.

  This data set will store a text file that will be used to ensure encryption occurs in a random manner. The data set must be defined with the following attributes: `DSORG=PO`, `RECFM=FB`, `LRECL=80`, and `BLKSZ=3120`.

Step 3. 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, copy them to the NETWRK.*vrs*.SAGSSL.CERTS data set defined in Step 2.

> **Notes:**

1. Certificates can be copied or renamed as required. They must have EBCDIC encoding and a record length of 251 bytes. All files, except the random file (see **Step 4**), must be in EBCDIC. Therefore, when transferring ASCII files from a personal computer using FTP, do not specify the binary option for these files. The binary option should be specified for the random file only.

2. The password must end with a null `-x'00'`. If you use FTP to transfer the password file from a personal computer to the mainframe, FTP may have converted the null to a space. If so, edit the file and insert a null at the end of the password string.

Step 4. Create the Text File Used to Ensure Random Encryption

In the NETWRK.*vrs*.SAGSSL.RANDOM data set, 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.

Step 5. Verify the Parameters in the SYSPARMS Member

The sample SYSPARMS member is stored in the Entire Net-Work TCP/IP Option source library. This member can be renamed, but if you do so, you must also alter the startup JCL references to it.

The following table describes the parameters listed in the sample SYSPARMS member and explains what values are expected for Encryption for Entire Net-Work.

Parameter	Description	Valid Values
ABEND_RECOVERY	Indicates whether a recovery environment is established for a logical process in the APS (Software AG internal) environment. When "NO" is specified, recovery or cleanup does not occur when an ABEND occurs for a process.	Valid values are YES and NO. For Encryption for Entire Net-Work, this parameter must be set to NO. The default is YES.
ASCII	Indicates whether ASCII runtime conversion should occur.	Valid values are YES and NO. For Encryption for Entire Net-Work, this parameter must be set to YES. The default is NO.
SYSTEM_ID	A name that uniquely identifies the POSIX server instance. The specified string is included in all messages issued to the operator during the execution of the POSIX server (excluding some startup and termination messages). It may also be used in the future by the POSIX server system to uniquely identify itself within a machine.	Valid values include any one to eight-character string. The default is "SysName."
THREAD_ABEND_RECOVERY	Indicates whether a recovery environment is established for a pthread created in the APS (Software AG internal) environment. When NO is specified, recovery or cleanup does not occur when an ABEND occurs in a pthread.	Valid values are YES and NO. For Encryption for Entire Net-Work, this parameter must be set to YES. The default is YES.

Step 6. Alter the Entire Net-Work Startup JCL

Make the following changes to the Entire Net-Work startup JCL. (A sample startup JCL member called JCLNET is provided in the z/OS source library) .

1. Add the APS (Software AG internal software) load library to your library concatenation. In z/OS environments, this version of Encryption for Entire Net-Work requires that level 20 of APS 3.3.1 or higher to be used.

   In z/OS, you would add this DD statement:

   ```
   //    DD DISP=SHR,DSN=APSvrs.MVSLD00
   ```

2. Add DD (z/OS) statements to the appropriate Entire Net-Work startup JCL. In z/OS, you would add these statements:

   ```
   //APSLOG   DD SYSOUT=*
   //APSTRCF  DD SYSOUT=*
   //SYSPARM  DD DISP=SHR,DSN=NETWRK.vrs.nnnnnnnnn(SYSPARMS)
   //NETPC    DD DISP=SHR,DSN=NETWRK.vrs.SAGSSL.CERTS(public-key-certificate-member)
   //NETPK    DD DISP=SHR,DSN=NETWRK.vrs.SAGSSL.CERTS(private-key-member)
   //NETCAF   DD DISP=SHR,DSN=NETWRK.vrs.SAGSSL.CERTS(CA-certification-member)
   //NETPSW   DD DISP=SHR,DSN=NETWRK.vrs.SAGSSL.CERTS(pem-passphrase-member)
   //NETRND   DD DISP=SHR,DSN=NETWRK.vrs.SAGSSL.RANDOM(random-member)
   ```

   The following table describes the symbolic names and the data set names and member names expected for each.

Symbolic Name	References
APSLOG	The SYSOUT specification for APS (Software AG internal library) logs.
APSTRCF	The SYSOUT specification for APS (Software AG internal library) traces.
SYSPARM	The SYSPARMS member in the source library for Entire Net-Work. The values in supplied in this sample member SYSPARMS were maintained in **Step 5**.
NETPC	A data set containing your company's public key and the signature of the certificate authority . This is the NETWRK.*vrs*.SAGSSL.CERTS data set defined in **Step 2**. Your company's public key file was deployed into this data set in **Step 3** of these instructions. This JCL statement is required if the SSL DRIVER statement is specified.
NETPK	A data set containing your company's private key . This is the NETWRK.*vrs*.SAGSSL.CERTS data set defined in **Step 2**. Your company's private key file was deployed into this data set in **Step 3**. This JCL statement is required if the SSL DRIVER statement is specified.
NETPSW	A single sequential data set or a member of a partitioned data set containing the password (*pem pass phrase*) required to decrypt the private key referenced by the NETPK JCL statement. The sequential or partitioned data set should be write and read-protected by

Symbolic Name	References
	your company's security subsystem, ideally so only Entire Net-Work can access it. The password specified must be null-terminated. Leading, embedded, and trailing blanks up to the null are treated as part of the password. This JCL statement is required if the SSL DRIVER statement is specified.
NETRND	The random file member you created in **Step 4**. This file is stored in the NETWRK.*vrs*.SAGSSL.RANDOM data set defined in **Step 2** or it can be stored in the same data set used for //NETPC, //NETPK, //NETCAF, and //NETPSW.

Step 7. Add SSL/SSLI DRIVER and LINK Statements

Add an SSL/SSLI DRIVER and LINK statements in the Entire Net-Work startup job. For complete information on the SSL/SSLI DRIVER statements and parameters, read *SSL/SSLI DRIVER Statement*, elsewhere in this guide. For complete information on the SSL/SSLI LINK statement and its parameters, read *SSL/SSLI LINK Statement*, elsewhere in this guide.

Step 8. Alter the Target Definitions (LUW Systems only)

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" or "SSLI" 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*, elsewhere in this guide.

Open Systems Activation

The following table lists the steps that must be completed to activate Encryption for Entire Net-Work on open systems. 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*, elsewhere in this guide.

7 Using the Mainframe SSL/SSLI Line Driver's

The SSL/SSLI line driver's are the security code provided with Encryption for Entire Net-Work on the mainframe. It includes SSL/SSLI DRIVER and LINK statements.

SSL/SSLI DRIVER Statement

The SSL/SSLI DRIVER statement has the syntax shown below. The SSLCAFIL, SSLVRF, and SSLVRS parameters must be specified to correctly implement Encryption for Entire Net-Work.

```
DRIVER  SSL{I}  [ACCEPTUI = { N | Y }]
                API = { HPS | BS2 | CNS | CMS | OES }
                [APITRACE = (N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y )]
                [CONNQUE = { n | 10 }]
                [DRVCHAR = { character | # }]
                [DRVNAME = { name |<Driver name>}]
                [KEEPALIV = { Y | N }]
                [MULTSESS = { N |Y }]
                [NUMUSERS = { number | 100 }]

                [PSTATS = { Y | N }]
                [RESTART = (interval, retries} ]
                [RSTATS = { Y | N }]
                [SERVERID = { n | 1996}]

                [STATINT = { stat-interval | 3600 }]
                [SUBSYS = { subsys-name | VMCF }]
                [TRACE = { Y | N }]
                [TRACELEV = (N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y )]
                [TRACESIZ = { size | 4096 }]

                [USERID =    { userid | TCPIP }
                             {  nn  |  00}            ]
```

The DRIVER statement parameters are read from a sequential file during system startup, and can be modified after startup using the ALTER operator command. Some parameters can be modified when the line driver is open or closed. Others can be modified only when the line driver is closed. Read about the ALTER and CLOSE commands in *Operator Commands*, elsewhere in this section. The open/closed requirement for each parameter is included in the parameter descriptions.

This section describes all of the parameters that can be used for the SSL DRIVER statement.

- ACCEPTUI Parameter
- API Parameter
- APITRACE Parameter (SSL Driver only)
- CONNQUE Parameter
- DRVCHAR Parameter
- DRVNAME Parameter
- KEEPALIV Parameter
- MULTSESS Parameter
- NUMUSERS Parameter (SSL Driver only)
- PSTATS Parameter
- RESTART Parameter
- RSTATS Parameter
- SERVERID Parameter
- STATINT Parameter
- SUBSYS Parameter
- TRACE Parameter
- TRACELEV Parameter
- TRACESIZ Parameter

- USERID Parameter

ACCEPTUI Parameter

ACCEPTUI = { Y | N }

This optional parameter determines whether the line driver will accept connections from systems that have not been previously defined with LINK statements. The ACCEPTUI parameter can be modified when the line driver is open or closed.

Valid values are "Y" (Yes) or "N" (No).

- If "Y" is specified, Entire Net-Work will accept connection requests from an undefined system and the required control blocks are built dynamically. Normal handshaking procedures with the new connections are performed. This is the default.
- If "N" is specified, Entire Net-Work will reject incoming requests from unknown source nodes.

API Parameter

API = { BS2 | CNS | OES }

This required parameter specifies the name of the TCP/IP application program interface being used. The API parameter can be modified only when the line driver is closed. Supported values are shown in the table below. There is no default.

Value	Description	Valid for Platforms
OES	Loads the IBM OpenEdition sockets interface NWTCPOES.	z/OS

> **Note:** At this time the API parameter must be set to "OES" for z/OS systems (this is the default) .

APITRACE Parameter (SSL Driver only)

APITRACE= { <u>N</u> | Y, <u>N</u> | Y, <u>N</u> | Y, <u>N</u> | Y, <u>N</u> | Y, <u>N</u> | Y, <u>N</u> | Y, <u>N</u> | Y, <u>N</u> | Y, <u>N</u> | Y}

This optional parameter specifies debugging trace levels. It is a series of flags that are passed to the API routine, which then determines the events to be traced. The APITRACE specification must be enclosed in parentheses. For example:

```
APITRACE=(N,N,N,N,N,N,N,N,N,N)
```

Trace levels are positional within the parameter syntax-example and are set using "Y" (Yes) and "N" (No). It is recommended that all flags remain set to N, the default value. If your system is experiencing problems, contact your Software AG technical support representative for the settings that will produce the appropriate trace information. The APITRACE parameter can be modified when the line driver is open or closed.

CONNQUE Parameter

CONNQUE = {*n* | **10**}

This optional parameter specifies the number of connect queue entries. The value specified must accommodate the maximum number of simultaneous connection requests from remote nodes.

After the connection is accepted or rejected, connect queue entries are reused. If the value of this parameter is not high enough, the API routine is not able to process the incoming connection and the partner will eventually time out. Depending on the API being used, a message may be displayed indicating that an error has occurred. Values can range from 1 to 64; a value greater than 64 is reset to 64. The default value is 10. The CONNQUE parameter can be modified only when the line driver is closed.

DRVCHAR Parameter

DRVCHAR = { *char* **| # }**

This optional parameter specifies the special character used to designate that an operator command should be directed to this line driver rather than to a specific link. The DRVCHAR parameter can be modified only when the line driver is closed.

The default for this parameter is "#".

DRVNAME Parameter

DRVNAME = {*name* |<Driver name>}

This optional parameter specifies the 4-byte driver name. The DRVNAME parameter can be modified only when the line driver is closed.

The default for this parameter is the "Driver name".

The DRVNAME parameter enables sites to make multiple TCP/IP API routines available at the same time. For example, the IBM APIs can be made available within the same Entire Net-Work address space. This parameter also allows two or more drivers to be defined so that Entire Net-Work can listen on multiple ports simultaneously.

KEEPALIV Parameter

KEEPALIV = { Y | N}

This optional parameter allows you to maintain connections when there is no other traffic with the remote links. Valid values are "Y" or "N".

■ When this value is set to "Y", it causes internal TCP messages to be sent periodically to all remote links, thus maintaining the connections when there is no other traffic with the remote links. The amount of time between messages is determined by an initialization parameter in the TCP stack.

■ When this value is set to "N", internal TCP messages are no longer sent periodically and the connections are not maintained.

The default for this parameter is "N".

KEEPALIV can also be set for individual remote links. For more information, read about the KEEPALIV parameter associated with the SSL LINK statement.

MULTSESS Parameter

MULTSESS = { N | **Y** }

This optional parameter determines whether a connect request from a host that has an active connection is treated as a new link. This parameter can be modified when the line driver is open or closed.

A value of "Y" indicates that the connect request is treated as a new link; a value of "N" indicates that the connect request is rejected. The default for this parameter is "Y".

NUMUSERS Parameter (SSL Driver only)

NUMUSERS = { *number* **| 100 }**

This parameter specifies the estimated maximum number of concurrently active clients. For performance reasons, a table of individual client entries is preallocated based on this number. During the Entire Net-Work session, if the number of active clients is exceeded, the table is automatically expanded by 50% of the current value. The size of each entry in the table is 256 bytes. The minimum value is 10; the maximum is 32767. The default is 100.

This parameter can only be altered when the driver is closed.

PSTATS Parameter

PSTATS = { Y | N }

This optional parameter determines whether statistics are printed. It does not affect the STATS command and can be modified when the driver is open or closed.

A value of "Y" indicates that statistics should be printed to DDPRINT when the statistics interval expires; a value of "N" indicates that the statistics should not be printed. The default for this parameter is "N".

RESTART Parameter

RESTART = (*interval, retries*)

This optional parameter specifies the retry interval in seconds (interval) and the number of retries (retries) that Entire Net-Work will attempt to reopen the access method with the API after a shutdown due to a failure. The RESTART parameter can be modified when the line driver is open or closed.

If RESTART is not specified, or interval is specified as zero, no retry is attempted. By specifying (retries) as zero, an infinite number of retries can be requested.

The RESTART parameter is particularly useful with Encryption for Entire Net-Work when Entire Net-Work is started at IPL and communication with the API is unsuccessful because TCP/IP is not yet fully initialized. Using this parameter, you can instruct Entire Net-Work to reopen the TCP/IP session, thereby giving TCP/IP sufficient time to become active.

The TIMER parameter on the NODE statement affects the RESTART parameter (read about the NODE statement in your Entire Net-Work reference documentation.) The retry interval should not be less than the TIMER parameter, and should be a multiple of this value. If a retry interval other than zero is specified that is less than the value of the TIMER parameter, the TIMER value is used instead.

RSTATS Parameter

RSTATS = { Y | N }

This optional parameter determines whether statistics are reset. It can be modified when the line driver is open or closed.

A value of "Y" indicates that statistics should be reset when the statistics interval expires; a value of "N" indicates that the statistics should not be reset. The default for this parameter in "N".

SERVERID Parameter

SERVERID = { *n* **| 1996}**

This optional parameter specifies a well-known port number used by Entire Net-Work while awaiting connection requests from participating Entire Net-Work partners. Values may range from 1 to 65535. The SERVERID parameter can be modified only when the line driver is closed.

When specified in a DRIVER statement, the SERVERID parameter specifies the port number of the Entire Net-Work being initialized. If SERVERID is not specified for a link, the SERVERID specified for the driver is used as the default port for the link.

The default for this parameter is 1996. Only the DRIVER statement has a SERVERID parameter.

STATINT Parameter

STATINT = {*interval* | **3600**}

This optional parameter specifies the amount of time, in seconds, before statistics are automatically printed or reset. The default is "3600". The STATINT parameter can be modified when the line driver is open or closed.

Acceptable values range from 1 to 2147483647. Any value outside this range is in error.

SUBSYS Parameter

SUBSYS = { *name* **| VMCF }**

This parameter specifies the name of the subsystem to be accessed by the API routines that use subsystem control blocks in interaddress space communications. The default value is "VMCF". The SUBSYS parameter can be modified only when the line driver is closed.

In a z/OS environment, the IBM API routines communicate to the system address space by locating the subsystem control table and retrieving the information required to perform cross-memory communication. If the subsystem is specified incorrectly, the driver is not able to perform its open processing and no connections are possible.

TRACE Parameter

TRACE = { Y | N }

This parameter indicates whether tracing for this line driver should be active "Y" or not "N". When tracing is activated, trace information is placed in the trace table. The default is "N" (No). The TRACE parameter can be modified when the line driver is open or closed.

This is equivalent to specifying `TRACE=linedriver-code` or `TRON=linedriver-code` in the NODE statement (for example, TRACE=SSL).

TRACELEV Parameter

TRACELEV = (Y | N̲, Y | N̲, Y | N̲, Y | N̲, Y | N̲, Y | N̲, Y | N̲, Y | N̲, Y | N̲, Y | N̲)

This optional parameter specifies the levels of tracing that the line driver will perform. It is a series of flags that determine which events are traced. The TRACELEV specification must be enclosed in parentheses. For example:

```
TRACELEV=(N,N,N,N,N,N,N,N,N,N)
```

Trace levels are positional within the parameter syntax and are set using "Y" (Yes) or "N" (No). It is recommended that all settings within the TRACELEV parameter be "N". If your system experiences problems, contact your Software AG technical support representative for the settings that produce the appropriate trace information. The TRACELEV parameter can be modified when the line driver is open or closed.

> **Note:** The tracing information provided is sent to the DDPRINT data set. In addition to setting the TRACELEV flags, the trace must also be turned on using either the DRIVER statement parameter TRACE=Y or the operator command `TRACE=linedriver-name`. Tracing dramatically affects the overall performance and throughput of Entire Net-Work.

TRACESIZ Parameter

TRACESIZ = { *size* **|** **4096** **}**

This optional parameter specifies the size, in bytes, of the driver-specific trace table. It can be modified only when the line driver is closed.

This parameter is also used as the default size of the link specific trace table when the LINK statement does not include a TRACESIZ specification.

Valid values can range from 4096 to 4194304. A value less than 4096 is reset to 4096; a value greater than 4194304 is reset to 4194304. The default for this parameter is "4096".

USERID Parameter

$$USERID = \left\{ \begin{array}{l} \{ \textit{userid} \mid \textbf{\underline{TCPIP}} \} \\ \{ \textit{nn} \mid \underline{\textbf{00}} \} \end{array} \right\}$$

This parameter's value can be modified only when the line driver is closed. It has two possible specifications:

- For IBM APIs (i.e., the NWTCPIBM interface, the NWTCPHPS interface, the NWTCPOES interface, and the NWTCPCMS interface), the USERID parameter specifies the name of the started task, job, or virtual machine in which the IBM TCP/IP protocol stack is running. The value is 1-8 characters. The default value is "TCPIP".

- For the CNS API (i.e., the NWTCPCNS interface), the USERID parameter is used to direct traffic to a particular TCP/IP stack. The value is a two-digit number that must match the ID= value in the PARM field of the TCP/IP stack. The default value is 00, which is also the default for the TCP/IP stack. The value of USERID is not validated; if its value does not match the ID= value of an active TCP/IP stack, the TCP/IP driver will fail to open and it will return messages similar to the following:

```
NETP571W TCP API ERROR ON OPEN - RC = 0008
NET0101I  TCPI DRIVER OPEN FAILED - RC = 0004
```

To use multiple TCP/IP stacks, one DRIVER statement must be provided for each stack, and each link must specify the driver associated with the stack it will use. When defining multiple drivers, copy the module NETTCPI.phase and change the last four characters of the name to match the DRIVER name. For example, to define the following, NETTCPI would be copied (not renamed) to NETTCP2:

```
*Links using Driver TCPI use TCP/IP stack with ID=00
DRIVER TCPI  API=CNS
LINK PC01 TCPI,INETADDR=(x,x,x,x)
*Links using Driver TCP2 use TCP/IP stack with ID=02
DRIVER TCP2 API=CNS,USERID=02,DRVNAME=TCP2
LINK PC02 TCP2,INETADDR=(x,x,x,x)
```

SSL/SSLI LINK Statement

The LINK statement and its parameters are used to define the characteristics of the remote client. With Encryption for Entire Net-Work, links are not normally predefined; they are dynamically allocated as clients initiate communication. However, links may be predefined to override defaults or provide some control over clients.

```
LINK linkname SSL{I}   [ADJHOST=Internet-host-name, ]
                       [INETADDR=(n1.n2.n3.n4) , ]
                       [KEEPALIV={ Y | N } , ]
                       [MULTSESS={ N | Y } , ]
                       [PSTATS={ N | Y } , ]
                       [RSTATS={ N | Y } , ]
                       [SAF={ Y | L | N } , ]
                       [SENDTIME={ time | 90 } , ]
                       [SERVERID= { n | 1996 }, ]
                       [STATINT={ interval | 3600 } , ]
                       [TRACESIZ=size ]
```

The LINK statement parameters are read from a sequential file during system startup, and can be modified after startup using the ALTER operator command. Some parameters can be modified when the link is open or closed. Others can be modified only when the link is closed. Read about the ALTER and CLOSE commands in *Operator Commands*, elsewhere in this section. The open/closed requirement for each parameter is included in the parameter descriptions.

- linkname SSL{I} Specification
- ADJHOST Parameter
- INETADDR Parameter
- KEEPALIV Parameter
- MULTSESS Parameter
- PSTATS Parameter
- RSTATS Parameter
- SAF Parameter
- SENDTIME Parameter
- SERVERID Parameter (SSLI Link only)
- STATINT Parameter

- TRACESIZ Parameter

linkname SSL{I} Specification

The *linkname* part of this required specification provides the name by which this link is to be known. It is positional, and must be specified immediately after the LINK keyword and immediately before the driver name ("SSL"); the link name must be unique on the node. All operator commands affecting the link must specify this name.

If the link name begins with the characters "MODEL", the link is defined as a model link. See the section *Model Links*, later in this section.

"SSL" is required and specifies the protocol name that defines the driver associated with this link. It must be the same as the value specified for the DRVNAME parameter in the SSL DRIVER statement and "SSLI" is required and specifies the protocol name that defines the driver associated with this link. It must be the same as the value specified for the DRVNAME parameter in the SSLI DRIVER statement.

ADJHOST Parameter

[ADJHOST= *internet-host-name* **]**

This optional parameter specifies the Internet host name of a node with which a connection is to be established. Its value can be 1 - 255 characters. The ADJHOST parameter can be modified only when the link is closed.

The ADJHOST parameter uses Domain Name Services (DNS), as follows:

- The GetHostByName function is used to determine the IP address of a host name specified with ADJHOST. IP address is used both for connecting to another node and for locating the link for an incoming connection.

- The GetHostByAddr function is used to determine the host name of a node that is trying to connect to this node. This is necessary when the IP address of a host name specified with AD-JHOST changes after the link has been opened.

Software AG recommends the use of the ADJHOST parameter for sites that assign IP addresses via the DHCP protocol. Entire Net-Work will use the GetHostByName function for every outgoing connection on nodes that have ADJHOST specified as long as INETADDR is not specified.

The following table lists the APIs that support Domain Name Services:

API	GetHostByName	GetHostByAddr
BS2	Yes	Yes
OES	Yes	Yes

For performance reasons, Software AG recommends that all LINK statements containing an AD-JHOST value be defined after the LINK statements containing an INETADDR specification. If neither of these parameters is specified, the link is not usable. If both are specified, the INETADDR parameter takes precedence.

INETADDR Parameter

[INETADDR = $n1.n2.n3.n4$]

This optional parameter specifies the IP (Internet Protocol) address of the remote host associated with this link. The INETADDR parameter can be modified only when the link is closed.

The IP address is used both for connecting to another node and for locating the link for an incoming connection. It is provided to Entire Net-Work in the form of INETADDR=(n1,n2,n3,n4) or INET-ADDR=(n1.n2.n3.n4) where each value represents 8 bits of the 32-bit IP address. Acceptable values are between 0 and 255 and may be separated by commas or periods.

For example:

```
INETADDR=(157,182,17,20)
```

or

```
INETADDR=(157.182.17.20)
```

On most UNIX-based machines, this address can be found in the /etc/hosts file. The following are examples of the information in the /etc/hosts file:

```
157.182.17.20 DALLAS dallas (VMS)
157.182.17.18 DENVER denver (UNIX)
```

Each host on the INTERNET is assigned a unique IP address, which is used by the internet protocol and higher level protocols to route packets through the network. The IP address is logically made up of two parts: the network number and the local address. This IP address is 32 bits in length and can take on different formats or classes. The class defines the length (number of bits) of each part. There are four classes (A, B, C, and D); the class is identified by the allocation of the initial bit.

For performance reasons, Software AG recommends that all LINK statements containing an INET-ADDR value be defined before the LINK statements containing an ADJHOST specification. If neither of these parameters is specified, the link is not usable. If both are specified, the INETADDR parameter takes precedence.

KEEPALIV Parameter

```
[KEEPALIV = { Y | N } ]
```

KEEPALIV=Y (Yes) causes internal TCP messages to be sent periodically to the remote node, thus maintaining the connection when there is no other traffic with the node. The amount of time between messages is determined by an initialization parameter in the TCP stack. If no KEEPALIV value is specified for the link, it defaults to the KEEPALIV value on the SSL DRIVER statement.

MULTSESS Parameter

[MULTSESS = { Y | N }]

This optional parameter determines whether a connect request from a host that has an active connection will be treated as a new link "Y", or a reconnection of an existing link "N". The default value is the value specified for the MULTSESS parameter in the SSL DRIVER statement (N or Y, with Y as the default). The MULTSESS parameter can be modified when the link is open or closed.

PSTATS Parameter

```
[PSTATS = { Y | N } ]
```

This optional parameter determines whether (Y or N) statistics are printed to DDPRINT when the statistics interval expires. The default value is the value specified for the PSTATS parameter in the SSL DRIVER statement (for which the default is N). This parameter does not affect the STATS operator command. The PSTATS parameter can be modified when the link is open or closed.

RSTATS Parameter

[RSTATS = { Y | N }]

This optional parameter determines whether (Y or N) statistics are automatically reset when the statistics interval expires. The default value is the value specified for the RSTATS parameter in the SSL DRIVER statement. The RSTATS parameter can be modified when the link is open or closed.

SAF Parameter

```
[SAF = { Y | L | N } ]
```

If SAF=Y or SAF=L is specified, Entire Net-Work will call the SAF Interface for all incoming requests on this link; failure to load the Interface is considered a security violation and Entire Net-Work will shut down. If SAF=L, the calls are traced and the output directed to DDPRINT. An error code is transmitted to the user if access to SAF is denied. The SAF parameter can be modified when the link is open or closed. The default value is "N" (No).

SENDTIME Parameter

[SENDTIME = { *seconds* **|** **90** **}]**

This optional parameter specifies the time (in seconds) that Encryption for Entire Net-Work allows for a send to complete. When this time is exceeded, a message is written to the operator console indicating a possible error condition on the remote node. The connection is considered severed and link disconnect processing is initiated. The default value is 90 seconds. The SENDTIME parameter can be modified when the link is open or closed.

SERVERID Parameter (SSLI Link only)

SERVERID = *port-number*

This parameter specifies a well-known port number associated with a remote partner. Entire Net-Work will attempt to use this port number when connecting to the remote partner. If this parameter is not specified, or is set to zero (0), the value specified for the SERVERID parameter on the DRIVER statement is used. The SERVERID parameter can be modified only when the link is closed.

STATINT Parameter

[STATINT = { *seconds* **| 3600 }]**

This optional parameter specifies the amount of time, in seconds, before statistics are automatically printed or reset. Acceptable values are 1 - 2147483647. Any value outside this range is in error. The default value is "3600". The STATINT parameter can be modified when the link is open or closed.

TRACESIZ Parameter

[TRACESIZ = *size*]

This optional parameter specifies the size of the TCP/IP link specific trace table. Value can be 4096 - 4194304. A value less than 4096 is reset to 4096. A value greater than 4194304 is reset to 4194304. The default value is the value specified for the TRACESIZ parameter in the SSL DRIVER statement. The TRACESIZ parameter can be modified only when the link is closed.

Operator Commands

Encryption for Entire Net-Work can process operator commands that are directed to a specific link or directly to its driver. This section describes the operator commands you can use.

- Command Syntax
- DRIVER Commands
- LINK Commands

For information about entering operator commands under z/VSE, refer to the Entire Net-Work VSE installation instructions.

Command Syntax

Under z/OS, the line driver operator commands have the following format:

F NETWORK, SSL *target cmd*

The following table describes this syntax.

Syntax Representation	Description
SSL	Informs Entire Net-Work that the command is destined for Encryption for Entire Net-Work. If more than one SSL DRIVER statement exists, use the name specified on the DRVNAME parameter of the SSL DRIVER statement instead of SSL.
target	A value that informs Entire Net-Work what the target of the command is, as follows: ■ Specify an asterisk (*) if the target is all links. ■ Specify the DRVCHAR value ("#" is the default) if the target is the driver itself (see the DRVCHAR parameter on the SSL DRIVER statement).

Syntax Representation	Description
	Specify the link name if the target is a specific link.
cmd	The operator commands to be carried out. Multiple commands can be specified in a single command statement. When the ALTER command is specified, it must be the last command in the statement, because everything following the ALTER command is treated as a DRIVER or LINK statement parameter. One or more DRIVER or LINK statement parameters must be specified.

The following are examples of line driver operator commands:

```
F NETWORK,SSL * CLOSE
SSL # STATS
```

F NETWORK, SSLI *target cmd*

The following table describes this syntax.

Syntax Representation	Description
SSLI	Informs Entire Net-Work that the command is destined for Encryption for Entire Net-Work. If more than one SSLI DRIVER statement exists, use the name specified on the DRVNAME parameter of the SSLI DRIVER statement instead of SSLI.
target	A value that informs Entire Net-Work what the target of the command is, as follows: ■ Specify an asterisk (*) if the target is all links. ■ Specify the DRVCHAR value ("#" is the default) if the target is the driver itself (see the DRVCHAR parameter on the SSLI DRIVER statement). Specify the link name if the target is a specific link.
cmd	The operator commands to be carried out. Multiple commands can be specified in a single command statement. When the ALTER command is specified, it must be the last command in the statement, because everything following the ALTER command is treated as a DRIVER or LINK statement parameter. One or more DRIVER or LINK statement parameters must be specified.

The following are examples of line driver operator commands:

```
F NETWORK,SSLI * CLOSE
SSLI # STATS
```

DRIVER Commands

Encryption for Entire Net-Work supports the commands listed in the following table when the target is the driver. The underlined portion of the command is the minimum abbreviation.

Command	Description
<u>AL</u>TER *driver-parms*	Dynamically changes the driver configuration. The ALTER command is followed by the driver configuration parameters to be altered. The driver configuration parameters are the same as those specified in the DRIVER statement. For example: `SSL # ALTER ACCEPTUI=Y`
<u>CL</u>OSE	Disconnects and closes all links that are connected to other nodes. Releases all resources held by the driver as well as all open links. Closes the driver.
<u>OP</u>EN	Reopens the driver after it is closed with the CLOSE operator command or because of an access method failure. Allocates all the resources needed by the driver to communicate with TCP/IP. Also attempts to resolve any unresolved host names.
<u>RES</u>ET	Resets all statistics for the driver. Statistics are printed only if the STATS command precedes the RESET command.
<u>SH</u>OW	Displays the current configuration of the driver. The current configuration is always shown automatically following an ALTER command.
<u>SN</u>AP	Causes all control blocks specific to the link to be snapped (printed in hexadecimal). Driver-specific control blocks and Entire Net-Work-specific control blocks are not snapped.
STATS	Causes the immediate printing of statistics and restarts the statistics interval. This command has no effect on the next automatic printing of statistics. To print and reset statistics, specify RESET immediately after the STATS command. For example: `SSL # STATS RESET`
<u>STATUS</u>	Displays the current status of the driver as well as a count of messages received and sent.
<u>TR</u>ACE	Causes Encryption for Entire Net-Work to format and print the driver-specific trace table. The trace table is formatted and printed in hexadecimal automatically when the SNAP command is processed.
USERS	Displays the Adabas user ID in character and hexadecimal formats, the Context ID and Context Verifier values (these are part of the internal message header and can be used to help identify the client), and the number of database calls received for the client.

> **Note:** When the driver is closed, it does not recognize the commands CLOSE, STATS, or RESET.

LINK Commands

Encryption for Entire Net-Work supports the commands listed in the following table when the target is a link or all links. The underlined portion of the command is the minimum abbreviation.

Command	Description
<u>AL</u>TER *link-parms*	Dynamically changes the link configuration. The ALTER command is followed by the link configuration parameters to be altered. The link configuration parameters are the same as those specified in the LINK statement. For example: `SSL LINK1 ALTER ADJHOST=DALLAS`
<u>CL</u>OSE	Disconnects the link if it is connected to another node and releases all resources held by the link.
<u>DI</u>SCONNECT	Starts the disconnect sequence for the target link(s). If the link is already disconnected or is in the process of disconnecting, the command is ignored.
LOGLON *linkname*	Turns on selective logging for the specified link.
LOGLOFF *linkname*	Turns off selective logging for the specified link.
<u>OP</u>EN	Allocates all the resources needed by the link to communicate with SSL. Does not initiate a connect to the remote node. The status of the link displayed via the SHOW operator command is not affected by the OPEN request.
<u>RESE</u>T	Resets all statistics for the link. Statistics are printed only if the STATS command precedes the RESET command.
<u>RESU</u>ME	Restarts processing on a link that was temporarily stopped due to a SUSPEND command.
<u>SH</u>OW	Displays the current configuration of the link. The current configuration is always shown automatically following an ALTER command.
<u>SN</u>AP	Causes all link-specific control blocks and the link-specific trace table to be snapped (printed in hexadecimal). Driver-specific control blocks and Entire Net-Work-specific control blocks are not snapped.
STATS	Causes the immediate printing of statistics and restarts the statistics interval. This command has no effect on the next automatic printing of statistics. To print and reset statistics, specify RESET immediately after the STATS command. For example: `SSL LINK1 STATS RESET`
<u>STATU</u>S	Displays the current status of the link as well as a count of messages received and sent.
<u>SU</u>SPEND	Temporarily stops all processing on a link. Processing can be restarted with the RESUME command.
<u>TR</u>ACE	Causes the link-specific trace table to be formatted and printed. The trace table is formatted and printed in hexadecimal automatically when the SNAP command is processed.

Command	Description
USERS	Displays the ADABAS user ID in character and hexadecimal formats, the IP address for the link, the Context ID and Context Verifier values (these are part of the internal message header and can be used to help identify the client), and the number of database calls received for the client.

Model Links

Encryption for Entire Net-Work supports dynamically added links, thus reducing the time required to set up and maintain the SSL LINK statements. Model links can be coded and used to define new links as they are added.

Encryption for Entire Net-Work is permitted to add links dynamically if ACCEPTUI=Y is coded on the DRIVER statement. A new link block is created and is used to control all further communications on the link. The link block can be initialized with default values that will be applied to each new link. Alternatively, one or more model links can be defined to override the values contained in the link block.

The model link statement is identical to other LINK statements, except that the link name begins with the characters "MODEL". Most of the model link parameters, such as PSTATS and RSTATS, are copied into the dynamically-built link block. Some parameters, such as INETADDR, are not copied because they are truly link-specific.

8   Using the SSL Toolkit

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*, elsewhere in this guide.

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)	The name of your city or town. If a default is provided, it is shown in brackets next to the prompt. 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.	Certificate authority C certificates
Common Name	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. 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.	Certificate authority C certificates
Country Name	A two-letter code for your country. If a default is provided, it is shown in brackets next to the prompt. 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.	Certificate authority C certificates
E-mail Address	Your e-mail address. The default is "Security@YourCompany.com". A maximum of 40 characters can be specified. 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.	Certificate authority C certificates
Organization Unit	The name of your department within the organization. If a default is provided, it is shown in brackets next to the prompt. 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.	Certificate authority C certificates
Organization Name	The name of your organization. If a default is provided, it is shown in brackets next to the prompt. 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.	Certificate authority
PEM Pass Phrase	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	Certificate authority C certificates

Information Requested	Description	Used to Create
	you use when setting up the certificate authority should be the same as the PEM password requested when creating a certificate. PEM passwords can be between 4 and 20 alphanumeric characters long, including blanks. They are case-sensitive.	
State or Province	The name of your state or province. If a default is provided, it is shown in brackets next to the prompt. 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.	Certificate authority C certificates
Optional Challenge Password	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. Challenge passwords can be between 4 and 20 alphanumeric characters long.	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
cacerts	Stores certificate authority files.
certs	Stores certificate files, signed or unsigned.
newcerts	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* : CA certificate that can be used on open systems.

- *cakey.pem*: A CA key file that can be used on open 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 [prefix]
```

where *prefix* is the prefix you want used in the certificate file names. All of the certificate and key files produced by the `makeccerts` command will begin with the prefix you specify.

The prefix specification is optional. If you do not specify a prefix, the prefix "myapp" is used. If you enter the same prefix twice, the newer certificate and key definitions will overwrite the older certificate and key definitions.

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.

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

- `<prefix>`cert.mf: Certificate file that can be used on mainframe systems.

- `<prefix>`cert.pem: Certificate file that can be used on open systems.

- `<prefix>`key.mf: Key file that can be used on mainframe systems.

- `<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.

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*

- *myappcert.pem*

- *myappkey.mf*

- *myappkey.pem*

- *myappCertreq.pem*

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.

9 Access and Connection Definition Setup

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
	If a certificate is received that is signed by a CA other than the CA specified by CAFILE, then the CAPATH is searched. **Note:** The file name specified may include the path information, unless a value for parameter CAPATH is specified.		
CAPATH	The location (path) where the CAFILE resides or where additional certificates of certificate authorities (CA) reside. **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. If parameter CAFILE includes location information, the value of CAPATH should be ".", which is also the CAPATH default.	Required only for client authentication.	Required only for server authentication.
CERT_FILE	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. **Note:** The file name specified may include the path information. This is useful if the certificate is not in the current directory.	Always required.	Required only for client authentication.
CERT_PSSWD	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. **Note:** You can specify a fully qualified file name for this parameter. In this case, the file name you provide must contain the password.	Always required.	Required only for client authentication.
KEY_FILE	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. **Note:** The file name specified may include the path information. This is useful if the certificate is not in the current directory.	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
	this file are used by the encryption routines to ensure that encryption itself occurs in a random manner. Some platforms (such as Solaris) require the use of a random file.		
VERIFY	The level of certificate verification to perform. Valid values are: ■ 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.) 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". **Note:** This parameter must be set to "3" if you are performing client authentication.	Use VERIFY=1 to request a client certificate and verify that it is sent. Use VERIFY=2 to force the sending of a client certificate. Use VERIFY=4 to limit the client certificate request to a single occurrence. VERIFY=8 is not valid for server processing.	Use VERIFY=0 (the C client default) to request a certificate but proceed even if certificate errors are found. Use VERIFY=1 to validate the server certificate. VERIFY=2 is not valid for client processing. VERIFY=4 is not valid for client processing. Use VERIFY=8 to validate that the common name of the received certificate matches the host name specified in the target entry.
VERSION	The version of SSL to use for processing. Valid values range from 1 through 4: ■ 1: (TLSv1) ■ 2: (SSLv2) ■ 3: (SSLv23). For C applications only, this indicates that Version 2 or 3 should be used. ■ 4: (SSLv3)	Optional	Optional

10   Security Scenarios

This chapter describes various mainframe and open system SSL scenarios using Encryption for Entire Net-Work.

Mainframe Scenarios

The following information is supplied for each mainframe scenario described in this section:

■ The client-side alterations you need to make to your database entries in either your Directory Server definitions or the Entire Net-Work Client definitions in the System Management Hub, as well as what parameters you must specify on those entries. For more information about these definitions, read *Access and Connection Definition Setup*, elsewhere in this guide.

■ The server-side data sets that must be defined in the Entire Net-Work startup JCL as well as the SSL DRIVER statement parameters that are expected.

The scenarios that are described are:

- Simple Encryption
- Client and Server Authentication
- Simple Encryption Between Entire Net-Work 7 and Entire Net-Work on the Mainframe

Simple Encryption

≫ **To perform simple encryption from a client:**

■ Change the communication protocol type to "SSL". For example, suppose the existing entry specified this:

```
TCPIP://ahost:9734
```

In this example, you would change the entry to look like this:

```
SSL://ahost:9734
```

This update can be made using the System Management Hub in either your Directory Server definitions or the Entire Net-Work Client definition. For more information, read *Access and Connection Definition Setup*, elsewhere in this guide.

≫ **To perform simple encryption from a server:**

1 Make sure the data sets and members defined by the following symbolic names are supplied in the Entire Net-Work startup JCL:

■ NETPC

- NETPK

- NETRND

For more information about each of these data sets, read *Step 6. Alter the Entire Net-Work Startup JCL* in *Mainframe Activation*, elsewhere in this guide.

2   Specify the SSL DRIVER statement in the Entire Net-Work startup JCL. For more information, about the SSL/SSLI DRIVER statement, read *SSL/SSLI DRIVER Statement*, elsewhere in this guide.

Client and Server Authentication

> **To perform client and server authentication from a client:**

■   Change the communication protocol type to "SSL" and specify values for the CAFILE, CAPATH, CERT_FILE, CERT_PSSWD, KEY_FILE, and VERIFY parameters. For example, suppose the existing entry specified this:

```
TCPIP://ahost:9734
```

In this example, you might change the entry to look like this:

```
SSL://ahost:9734?CAFILE=cacert.pem&CAPATH=path&VERIFY=1&CERT_FILE=testcert.pem&KEY_FILE=testkey.pem&CERT_PSSWD=testing
```

This update can be made using the System Management Hub in either your Directory Server definitions or the Entire Net-Work Client definition. For more information, read *Access and Connection Definition Setup*, elsewhere in this guide.

> **To perform client and server authentication from a server:**

■   Make sure the data sets and members defined by the following symbolic names are supplied in the Entire Net-Work startup JCL:

- NETCAF

- NETPC

- NETPK

- NETPSW

- NETRND

For more information about each of these data sets, read *Step 6. Alter the Entire Net-Work Startup JCL* in *Mainframe Activation*, elsewhere in this guide.

Simple Encryption Between Entire Net-Work 7 and Entire Net-Work on the Mainframe

≫ **To perform simple encryption between Entire Net-Work 7 (open systems) and Entire Net-Work on the mainframe:**

■ Change the communication protocol type to "SSL" and specify values for the CAFILE, CAPATH, CERT_FILE, CERT_PSSWD, KEY_FILE, and VERIFY parameters. For example, suppose the existing entry specified this:

```
TCPIP://USZHOST:9734?retry=32767&retryint=60&reconnect=yes
```

In this example, you might change the entry to look like this:

```
SSL://USZHOST:9734?retry=32767&retryint=60&reconnect=yes
```

This update can be made using the System Management Hub in either your Directory Server definitions or the Entire Net-Work Client definition. For more information, read *Access and Connection Definition Setup*, elsewhere in this guide.

Open Systems Scenarios

For each open systems 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
- Client-Only Authentication
- Server-Only Authentication
- Client and Server Authentication
- Authentication with Certificates Elsewhere
- Authentication with a Hidden Password

Simple Encryption

≫ **To perform simple encryption for an Entire Net-Work Client:**

1 Access the Entire Net-Work Client access definition to Adabas databases in the System Management Hub. For more information, read *Maintaining Adabas Access Definitions*, in the *Entire Net-Work Client Installation and Administration Guide*.

2 Change the communication protocol type to "SSL" by selecting the **SSL** or **SSL Protocol** radio button in the definition.

3   Save the definition.

> **To perform simple encryption for a Kernel connection:**

1   Access the Kernel connection definition in the System Management Hub. For more information, read *Maintaining Connection Definitions*, in the *Entire Net-Work Server LUW Installation and Administration Guide*.

2   Change the communication protocol type to "SSL" by selecting the **SSL** or **SSL Protocol** radio button in the definition.

3   Save the definition.

> **To perform simple encryption for an Entire Net-Work Server:**

1   Access the Entire Net-Work Server Kernel access definitions in the System Management Hub. For more information, read *Adding Kernel Definitions* and *Maintaining Access Definitions*, in the *Entire Net-Work Server LUW Installation and Administration Guide*.

2   For each Kernel definition that needs to support SSL, verify that either the **E-business SSL Access** or **E-business SSL Client Access** option is selected and that appropriate port numbers are specified.

3   For both **E-business SSL Access** and **E-business SSL Client Access**, 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.

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 the System Management Hub. For more information, read *Maintaining Adabas Access Definitions*, in the *Entire Net-Work Client Installation and Administration Guide*.

2   Change the communication protocol type to "SSL" by selecting the **SSL** or **SSL Protocol** radio button in the definition.

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 a Kernel connection:**

1   Access the Kernel connection definition in the System Management Hub. For more information, read *Maintaining Connection Definitions*, in the *Entire Net-Work Server LUW Installation and Administration Guide*.

2   Change the communication protocol type to "SSL" by selecting the **SSL** or **SSL Protocol** radio button in the definition.

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 the System Management Hub. For more information, read *Adding Kernel Definitions* and *Maintaining Access Definitions*, in the *Entire Net-Work Server LUW Installation and Administration Guide*.

2   For each Kernel definition that needs to support SSL client-only authentication, verify that the **E-business SSL Client Access** option is selected and that an appropriate port number is specified.

3   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.

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 the System Management Hub. For more information, read *Maintaining Adabas Access Definitions*, in the *Entire Net-Work Client Installation and Administration Guide*.

2   Change the communication protocol type to "SSL" by selecting the **SSL** or **SSL Protocol** radio button in the definition.

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 connection:**

1   Access the Kernel connection definition in the System Management Hub. For more information, read *Maintaining Connection Definitions*, in the *Entire Net-Work Server LUW Installation and Administration Guide*.

2   Change the communication protocol type to "SSL" by selecting the **SSL** or **SSL Protocol** radio button in the definition.

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 the System Management Hub. For more information, read *Adding Kernel Definitions* and *Maintaining Access Definitions*, in the *Entire Net-Work Server LUW Installation and Administration Guide*.

2   For each Kernel definition that needs to support SSL server-only authentication, verify that the **E-business SSL Access** option is selected and that an appropriate port number is specified.

3   For **E-business SSL 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   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 definition to Adabas databases in the System Management Hub. For more information, read *Maintaining Adabas Access Definitions*, in the *Entire Net-Work Client Installation and Administration Guide*.

2   Change the communication protocol type to "SSL" by selecting the **SSL** or **SSL Protocol** radio button in the definition.

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 a Kernel connection:**

1   Access the Kernel connection definition in the System Management Hub. For more information, read *Maintaining Connection Definitions*, in the *Entire Net-Work Server LUW Installation and Administration Guide*.

2   Change the communication protocol type to "SSL" by selecting the **SSL** or **SSL Protocol** radio button in the definition.

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   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 the System Management Hub. For more information, read *Adding Kernel Definitions* and *Maintaining Access Definitions*, in the *Entire Net-Work Server LUW Installation and Administration Guide*.

2   For each Kernel definition that needs to support SSL client and server authentication, verify that either the **E-business SSL Access** or **E-business SSL Client Access** option is selected and that appropriate port numbers are specified.

3   For **E-business SSL 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 target entries:**

■   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=xxcert.pem&KEY_FILE=xxkey.pem&CERT_PSSWD=FILE://C:/certs/certpswd.txt&VERIFY=3
```

11 Making SSL Certificates and Keys

This involves utilities provided on the OpenSSL web site at **https://www.openssl.org/** in Binary Distributions.

1. Download and extract the file on the chosen platform.

2. Add a folder "certs".

3. Add the following files:

   - .rand (contains about 20 lines of garbage data / hit the keyboard)

   - serial (contains the first serial number of new certs, example 1000)

   - index.txt (empty)

4. Edit the *genca.cnf* keywords with the following values:

```
[ CA_default ]

dir                 = .                 # Where everything is kept
certs               = $dir/certs        # Where the issued certs are kept

certificate         = $certs/WCPcacert.pem   # The CA certificate
serial              = $dir/serial.TXT        # The current serial number
crl                 = $certs/WCPCrl.pem       # The current CRL
private_key         = $certs/WCPCaKey.pem # The private key
RANDFILE            = $dir/.rnd              # private random number file


[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)    #Prompt
countryName_default         = DE

stateOrProvinceName         = State or Province      #Prompt
stateOrProvinceName_default =

localityName                = City or Town    #Prompt
localityName_default        = Darmstadt
```

```
O.organizationName               = Organizational Name (eg, company)  #Prompt
O.organizationName_default       = Software AG

organizationalUnitName           = Organization unit (group or department) #Prompt
organizationalUnitName_default = Network
```

Notes:

a. The *countryName_default* is your country (DE is for Germany, US - United States etc...).

b. Change the defaults above as necessary.

Setting the defaults saves typing later.

5. Make Selfsigned CA Key:

```
openssl req -config genca.cnf -newkey rsa:2048 -x509 -keyout WCPcakey.pem -out ↵
WCPcacert.pem -days 365
```

Reply:

```
 Enter PEM pass phrase: <your_password>
   ... confirm ....
   Country Name  :
   State         :
   City          :
   Organ. Name   :
   Organ. Unit   :
   Common Name   : <DefaultCA_name>
   Email Addr.   :
```

The other fields are set in *genca.cnf*.

6. Make application certificate & request:

```
openssl req -config genca.cnf -newkey rsa:2048 -out WcpAppCertReq.pem -keyout ↵
WcpAppKey.pem -days 365
```

Reply:

```
Enter PEM pass phrase: <your_password>
   ... confirm ....
   Country Name  :
   State         :
   City          :
   Organ. Name   :
   Organ. Unit   :
   Common Name   : <localhost_name>
   Email Addr.   :
   ...extra attributes:  <leave empty>
```

The other fields are set in *genca.cnf*.

7. Sign the app cert req:

```
openssl ca -config genca.cnf -policy policy_anything -out WcpAppCert.pem -infiles ↵
WcpAppCertReq.pem
```

Reply:

```
   Entr pass phrase..: <your_password>
   Sign the cert?    : y
   commit?           : y
```

8. Password file

Make a file containing the password with a null character at the end.

9. At this point, the folder *certs* should contain:

```
WCPAppCert.pem
    WCPAppKey.pem
    WCPCACert.pem
    WCPCAKey.pem
    <random_file>
    <passwored_file>
```

> **Note:** Make sure that the files contain DOS <CR><LF> at the end of each line.

Index