Activating Encryption for Entire Net-Work

This chapter describes the steps that must be completed to activate Encryption for Entire Net-Work. It is organized in the following topics:

- Mainframe Activation
- Open Systems Activation

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*.

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.

In z/VSE environments, if a certificate file (such as the private key, public key, password file, or random file) fits in 80-byte records, the file may be stored in a Librarian member. If the file exceeds 80-byte records, it must be stored as a sequential file.

If you need to FTP files from a personal computer to a z/VSE sequential file or Librarian member, examples are provided here:

- PC-to-VSE Sequential File FTP Example
- PC-to-VSE Librarian Member FTP Example

PC-to-VSE Sequential File FTP Example

The following code is part of a batch job you could use to FTP a certificate file from a personal computer to a z/VSE sequential file:

```
// EXEC FTP
   LOPEN 10.20.46.111
     LUSER SYSA
     LPASS SYSA
     LPWD
   OPEN 10.156.70.238
   LQUOTE SITE RECFM FB
   LQUOTE SITE LRECL 251
   LQUOTE SITE BLOCK 6024
     USER FTP
     PASS FTP
     BINARY
     GET rnd.pem SEQTEST
/*
// UPSI 1
// DLBL SEQTEST,'seq.test.file',0,SD
// EXTENT SYS004,DOSRES
// ASSGN SYS004,DISK,VOL=DOSRES,SHR
```

PC-to-VSE Librarian Member FTP Example

The following code is part of a batch job you could use to FTP a certificate file from a personal computer to a z/VSE Librarian member:

```
// EXEC FTP
   LOPEN 10.20.46.111
     LUSER SYSA
     LPASS SYSA
     LCD   SAGLIB
     LCD   WSL111
     LPWD
   OPEN 10.156.70.238
   LQUOTE SITE RECFM FB
   LQUOTE SITE LRECL 80
   LQUOTE SITE BLOCK 6080
     USER FTP
     PASS FTP
     GET CAPPCERT     TSTCERT.PEM
```

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; a sample startup JCS member called JCSNET is provided in the z/VSE 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 11 of APS 2.7.2 be used; in z/VSE environments, this version of Encryption for Entire Net-Work requires that level 18 of APS 2.7.2 be used.

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

   ```
   //    DD DISP=SHR,DSN=APS272.MVSLD00
   ```

   In z/VSE, you would add the following DLBL statement:

   ```
   // DLBL SAGLIB,' NETWRK.Vvrs.LIBRARY',99/365,SD
   ```

   In addition, in z/VSE, you would verify that the Encryption for Entire Net-Work (WSL) and APS libraries are in your LIBDEF search chain. For example:

   ```
   LIBDEF PHASE,SEARCH=(SAGLIB.WCPnnnZ,SAGLIB.WCPnnn,                  X
                 SAGLIB.WTCvrs,SHRLIB.WALvrs,                            X
                 SAGLIB.WSLvrsZ,SAGLIB.WSLvrs,                           X
                 SAGLIB.BTEvrsCS, SAGLIB.BTEvrsDS,                       X
                 SAGLIB.APS27218, SAGLIB.APS272)
   ```

2. Add DD (z/OS) and DLBL (z/VSE) 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)
   ```

   In z/VSE, you would add these statements if the certificate files were stored in librarian members:

   ```
   // DLBL NETPSW,'/SAGLIB/WSLnnn/pem-passphrase-member'
   // DLBL NETPC,'/SAGLIB/WSLnnn/public-key-certificate-member'
   // DLBL NETPK,'/SAGLIB/WSLnnn/private-key-member'
   // DLBL NETCAF,'/SAGLIB/WSLnnn/CA-certification-member'
   // DLBL NETRND,'/SAGLIB/WSLnnn/random-member'
   ```

   In z/VSE, you would add these statements if the certificate files were stored in sequential files:

   ```
   // DLBL NETPSW,'netpsw,seq,file',xxxx/yy,SD
   // EXTENT SYS004,VSEzzz,1,0,40000,160
   // ASSGN SYS004,DISK,VOL=VSEzzz,SHR
   // DLBL NETPC,'netpc,seq,file',xxxx/yy,SD
   // EXTENT SYS004,VSEzzz,1,0,40000,160
   // ASSGN SYS004,DISK,VOL=VSEzzz,SHR
   // DLBL NETPK,'netpk,seq,file',xxxx/yy,SD
   // EXTENT SYS004,VSEzzz,1,0,40000,160
   // ASSGN SYS004,DISK,VOL=VSEzzz,SHR
   // DLBL NETCAF,'netcaf,seq,file',xxxx/yy,SD
   // EXTENT SYS004,VSEzzz,1,0,40000,160
   // ASSGN SYS004,DISK,VOL=VSEzzz,SHR
   // DLBL NETRND,'netrnd,seq,file',xxxx/yy,SD
   // EXTENT SYS004,VSEzzz,1,0,40000,160
   // ASSGN SYS004,DISK,VOL=VSEzzz,SHR
   ```

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. (z/OS only)
APSTRCF	The SYSOUT specification for APS (Software AG internal library) traces. (z/OS only)
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 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.

Symbolic Name	References
NETCAF	A single sequential data set or a member of a partitioned data set containing the certificate authority's public key , and the signature of the certificate authority. This is the NETWRK.*vrs*.SAGSSL.CERTS data set defined in Step 2. The certificate authority's public key file was deployed into this data set in Step 3. This JCL statement is required if SSLCAFIL=Y is specified as an SSL DRIVER statement parameter. It is not necessary in every secured transmission scenario, but it is always necessary when you are performing client or client/server authentication.

Step 7. Add SSL DRIVER and LINK Statements

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

Step 8. 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*.

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*.

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*.

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*.

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*.