z/OS Environments

This section provides information about installing and running Entire Net-Work in the z/OS operating system environment. It covers the following topics:

- Library Requirements

- Installation Procedure

- APF Authorization

- Execution Job Example

- Entering Operator Commands

Library Requirements

The source library requires one cylinder of 3390 DASD space. The load library requires four cylinders. See the section *Installation Tape Information* and the *Report of Tape Creation* for additional information.

Installation Procedure

To install Entire Net-Work on a z/OS system, perform the following steps for each machine:

- Step 1: Copy the Tape Contents

- Step 2: Install the Adabas SVC

- Step 3: Relink ADARUN for APF Authorization

- Step 4: Prepare and Install the Product License File

- Step 5: Customize the Entire Net-Work Startup Jobs

- Step 6: Modify the Entire Net-Work Parameters

- Step 7: Install Appropriate Line Drivers

- Step 8: Prepare Access Method-Specific Definitions

- Step 9: Apply Corrective Maintenance

Step 1: Copy the Tape Contents

Note:
If you are using SMA, refer to the *System Maintenance Aid* documentation. If you are not using SMA, follow the instructions below.

This section explains how to copy all data sets from tape to disk. You will then need to perform the individual installation procedure for each component to be installed.

- Step 1: Copy Data Set COPY.JOB from Tape to Disk
- Step 2: Modify COPY.JOB on Your Disk
- Step 3: Submit COPY.JOB

Step 1: Copy Data Set COPY.JOB from Tape to Disk

The data set COPY.JOB contains the JCL required to copy all data sets from tape to disk. If the data sets for more than one product are delivered on the tape, the data set COPY.JOB contains the JCL to unload the data sets for all delivered products from the tape to your disk.

Copy COPY.JOB to your disk using the following sample JCL:

```
//SAGTAPE JOB SAG,CLASS=1,MSGCLASS=X
//* -------------------------------
//COPY EXEC PGM=IEBGENER
//SYSUT1 DD DSN=COPY.JOB,
// DISP=(OLD,PASS),
// UNIT=(CASS,,DEFER),
// VOL=(,RETAIN,SER=tape-volume),
// LABEL=(2,SL)
//SYSUT2 DD DSN=hilev.COPY.JOB,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=volume,
// SPACE=(TRK,(1,1),RLSE),
// DCB=*.SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//
```

where:

hilev is a valid high-level qualifier
tape-volume is the tape volume name, for example: T12345
volume is the disk volume name

Step 2: Modify COPY.JOB on Your Disk

Modify COPY.JOB according to your local naming conventions and set the following disk space parameters:

- Set HILEV to a valid high-level qualifier.

- Set LOCATION to a storage location.

- Set EXPDT to a valid expiration date.

Step 3: Submit COPY.JOB

Submit COPY.JOB to copy all data sets from tape to your disk.

Step 2: Install the Adabas SVC

In systems where Adabas is not already installed, install the Adabas SVC.

Refer to the *Adabas Installation Manual* for specific details.

Step 3: Relink ADARUN for APF Authorization

Relink ADARUN for APF authorization, if required.

Step 4: Prepare and Install the Product License File

You must install a valid license file on all mainframe platforms in which your Software AG mainframe product is installed. The license file is provided as an XML document (encoding is US-ASCII) and must remain in that format -- even on the mainframe. It must not be modified. Any modification of the license file will invalidate the digital signature and the license check will fail. If the check fails, you will not be able to run the product. In the event of a check failure, please contact your Software AG technical support representative.

Note:
Thirty days before the license expires, license check failure messages are produced. Your software product will still function, but these messages warn you that it is time to obtain a new license.

In this step, you will prepare the license file (obtain it from e-mail or the installation tape and store it on your z/OS system) and then install it:

- Preparing the Product License File
- Installing the Product License File

Preparing the Product License File

The product license file is supplied on the individual customer installation tape or separately via an e-mail attachment. Before you can install the license, you must transfer it from e-mail or the installation tape and store it on a z/OS system. This section describes how to do this for a license distributed either by e-mail or on the installation tape.

▶ **To prepare the license file from an e-mail attachment, complete the following steps:**

1. Transfer the license to z/OS, as described in *Transferring a License File from PC to a z/OS Host Using FTP*.

2. Verify that the transferred license file is stored in an Entire Net-Work source library (with RECFM=F or FB and LRECL=80), taking care to preserve its format as ASCII.

▶ **To prepare the license file from the installation tape, complete the following steps:**

- Verify that the license file is stored from the tape into an Entire Net-Work source library (with RECFM=F or FB and LRECL=80), taking care to preserve its format as ASCII.

Installing the Product License File

Once the license file has been prepared, you can install it in one of two ways:

- You can convert the license to a load module (NETLIC) that is then loaded by Entire Net-Work.

- You can reference the license file in the Entire Net-Work startup job by DD statement.

This section describes both methods.

▶ **To convert the license file to a load module, complete the following steps:**

1. Review and modify sample job LICASM, as follows:

   - Change the STEPLIB DD statement to point to your license load library (MLC*vrs*.LOAD).

   - Change the SYSUT1 DD statement to point to the data set containing the Entire Net-Work license file you transferred to z/OS earlier.

   - Specify an appropriate user load library for the L.SYSLMOD DD statement.

     Note:
     This user load library must also be included in the STEPLIB concatenation for the Entire Net-Work.

2. Submit sample job LICASM. This job runs the MAKE function of the LICUTIL utility to convert the license text file to an assembler source module. LICASM then links and assembles the assembler source to generate a load module called NETLIC, which is stored in the specified user load library (L.SYSLMOD DD statement). For more information about the LICUTIL utility, read *Using The License Utility: LICUTIL*.

3. Update your Entire Net-Work startup jobs to reference the user load library so NETLIC will be loaded by Entire Net-Work, as described in *Step 5: Customize the Entire Net-Work Startup Jobs*.

▶ **To reference the license file in the Entire Net-Work startup job, complete the following steps:**

1. Make sure any previously-created NETLIC load module is inaccessible to the Entire Net-Work load library being used in your Entire Net-Work startup job. Entire Net-Work first tries to load NETLIC and, if unsuccessful, it reads from a DDLIC data set referenced in the Entire Net-Work startup job.

2. Update your Entire Net-Work startup jobs to reference the license, as described in *Step 5: Customize the Entire Net-Work Startup Jobs*.

Step 5: Customize the Entire Net-Work Startup Jobs

Customize the Entire Net-Work startup jobs. The following modifications must be made:

1. Software AG licensing requires that the modules LICMAIN and LICUTIL be loaded when Entire Net-Work starts up. These modules are distributed in the MLC*vrs*.LOAD library. You must either:

   - Copy LICMAIN and LICUTIL into NETWRK.*vrs*.LOAD; or

- Concatenate MLC*vrs*.MVSLOAD with NETWRK.*vrs*.LOAD.

2. Verify that the license file is correctly referenced in each startup job. Do either of the following:

   - Verify that the NETLIC load module, installed in *Step 4: Prepare and Install the Product License File*, is stored in a load library that is accessible to the Entire Net-Work load library. Add the user load library in which NETLIC resides to the STEPLIB concatenation of the Entire Net-Work startup job or copy the NETLIC library into NETWRK.*vrs*.LOAD.

   - Verify that there is no NETLIC load module accessible to the Entire Net-Work load library and that the following DD statement is included in the ADANUC job:

     ```
     //DDLIC     DD  DISP=SHR,DSN=dsn
     ```

     where *dsn* is the data set name of the license file loaded from the tape (in ASCII format). Note that *dsn* could reference a member in a partitioned data set.

     Note:
     Entire Net-Work first tries to load NETLIC and, if unsuccessful, it reads from the DDLIC data set.

3. Data set names for the database and libraries must be customized for your installation.

   Note:
   Be sure to include appropriate user load libraries.

Step 6: Modify the Entire Net-Work Parameters

Modify the Entire Net-Work parameters.

See the section *Entire Net-Work Parameter Statements*.

Step 7: Install Appropriate Line Drivers

Perform the line driver-dependent installation procedure and prepare the DRIVER and LINK statementsfor the TCPX line driver.

Step 8: Prepare Access Method-Specific Definitions

Prepare the required access method-specific definitions.

Step 9: Apply Corrective Maintenance

Apply corrective maintenance.

Refer to the *Report of Tape Creation* to determine whether any files containing corrective maintenance (i.e., data sets named *pppvrs*.ZAPS) are supplied on the installation tape. If so, restore the data sets using IEBCOPY, and then follow the corrective maintenance instructions in the $READMVS member.

⚠️   **Warning:**
     Zap WM621028 must be applied in Entire Net-Work 6.2 SP1
     installations. This zap must be applied to all nodes that will connect
     together, regardless of the operating system or whether Adabas
     Cluster Services exists on the node. If an Entire Net-Work 6.2 SP1
     node without this zap attempts to connect to a node that has it
     applied, message NETT029I will be issued and the connection will
     fail. If a node without zap WM621028 attempts to connect via IUCV
     to a node that has the zap applied, one of the nodes will get message
     NET0151I and the connection will fail.

APF Authorization

Certain line drivers require that Entire Net-Work run in supervisor state with AMODE (31). For other line drivers supervisor state is not required but does provide significant performance advantages. Using AMODE (31) allows buffers to be above the 16 MB line.

To run in supervisor state, the Entire Net-Work load library and all other load libraries in the STEPLIB concatenation must be APF-authorized. Also, ADARUN must be linked into one of these APF-authorized libraries with SETCODE AC(1). ADARUN can be linked as shown in the following example:

```
//LINKRUN EXEC PGM=IEWL
//SYSPRINT   DD    SYSOUT=*
//SYSUT1     DD    UNIT=SYSDA,SPACE=(CYL,(1,1))
//ADALIB     DD    DSN=adabas.loadlib,DISP=SHR
//SYSLMOD    DD    DSN=network.loadlib,DISP=SHR   <=== APF-AUTHORIZED
//SYSLIN     DD    *
 MODE      AMODE(31),RMODE(24)
 INCLUDE   ADALIB(ADARUN)
 SETCODE   AC(1)
 NAME   ADARUN(R)
```

Execution Job Example

The following is an example of an Entire Net-Work z/OS execution job. (See the sample source member JCLNET in the source library for an alternate example.) More JCL may be needed, depending on the node configuration (types of line drivers, number of links, and so on). For more information, refer to the Simple Connection Line Driver documentation.

```
//*--------------------------------------------------------------*
//*                     SAMPLE   JCL                             *
//*--------------------------------------------------------------*
//STEP1 EXEC PGM=ADARUN,REGION=2048K,TIME=1440       <--- see Note 1
//STEPLIB  DD DISP=SHR,DSN=WCAvrs.LOAD,DCB=BLKSIZE=32760
//         DD DSN=WAL.vrs.LOAD,DISP=SHR              <--- see Note 2
//DDLIC DD DSN=NETWRK.vrs.LICENSE,DCB=(RECFM=FB,LRECL=80,BLKSIZE=80*n) <---See Note 3
//DDPRINT  DD SYSOUT=*                               <--- see Note 4
//NETPRNT  DD DISP=SHR,DSN=WCAvrs.NETPRNT            <--- see Note 5
//MPMDUMP  DD SYSOUT=*                               <--- see Note 6
//SYSUDUMP DD SYSOUT=*
//DDCARD   DD DISP=SHR,DSN=WCAvrs.SRCE(ADARUN)       <--- see Note 7
//         DD DISP=SHR,DSN=WCAvrs.SRCE(FORCEN)
//DDKARTE  DD DISP=SHR,DSN=WCAvrs.SRCE(NWKWCA)       <--- see Note 8
```

Here is a sample of the contents of library member NWKWCA, listing the Entire Net-Work parameter statements for this Entire Net-Work Administration execution:

```
*                     NODE STATEMENT                                    *
*-----------------------------------------------------------------------*
NODE NODENAME CQTIMER=60,                    -
              LOG=NO,                        -
              MAXPATH=10,                    -
              NTRACE=1000,                   -
              REPLYTIM=60,                   -
              TIMER=20
*
*-----------------------------------------------------------------------
*                                                                       *
*     DRIVER TCPX                                                       *
*                                                                       *
*-----------------------------------------------------------------------
DRIVER  TCPX    API=OES,                                     -
                SERVERID=1996,                               -
                USERID=CCCC
*-----------------------------------------------------------------------
*                E N D   O F   P A R A M E T E R S ..         *
*-----------------------------------------------------------------------
```

Notes:

1. The region size required varies with the number and type of links, as well as other operating parameters.
2. The second data set of the STEPLIB concatenation should always be your most current Adabas load library (Entire Net-Work Administration Version 6.2 requires Adabas mainframe Version 8 or above) or the Adabas limited library (WAL), as provided on the Entire Net-Work installation tape, unless you have been specifically instructed otherwise by Software AG.
3. DDLIC contains the binary license key. It is only required if a license module is not present in the STEPLIB.
4. All Entire Net-Work Administration print output is written to DDPRINT.
5. All diagnostic information from tracing, logging, and abends is written to the NETPRNT file if it is open (otherwise it is written to the DDPRINT file). NETPRNT can be allocated to a large data set that can be copied when closed. The data set should be created with the DCB attributes RECFM=FBA and LRECL=121. To do this, allocate the file SHR. This causes the data set to be erased at the time the file is opened. Be aware that the diagnostic information is very large and will fill a data set quickly. When this happens, the file is closed and all additional output is sent to DDPRINT. This diagnostic information is created by Entire Net-Work and does not include the operating system dump information written to SYSUDUMP.
6. If MPMDUMP is defined, a snap dump is produced during any abnormal termination. In some error situations, the MPMDUMP dump may contain more pertinent information than the SYSUDUMP dump.
7. DDCARD contains the ADARUN control statements or it identifies the library member in which the ADARUN control statements are defined. These ADARUN control statements define the interregion communications parameters for Entire Net-Work. See the section *ADARUN Control Statements*.
8. DDKARTE contains Entire Net-Work parameter statements or identifies the library member in which the parameter statements are defined. These parameter statements describe the local environment and the network connections for this node. See the section *Entire Net-Work Parameter Statements*.

Entire Net-Work can also be installed as a started task; no special considerations apply.

Entire Net-Work uses cross-memory services similar to Adabas in z/OS systems. As a result, z/OS removes the address space and initiator when Entire Net-Work terminates operation. This is normal and should not be regarded as an error.

Entering Operator Commands

During execution, a number of operator commands may be issued to Entire Net-Work to display or modify the system status. These commands are essentially identical for all operating environments. See the section *Entire Net-Work Operator Commands*.

The way in which operator commands are presented to Entire Net-Work depends on the operating system and is identical to the way operator commands are presented to Adabas.

In z/OS environments, the following operator commands are used:

```
MODIFY (abbreviation = F)
STOP (abbreviation = P)
```

The STOP command serves as an alternative to the NETEND command and terminates Entire Net-Work. Its synonyms are described as part of the following example.

Example:

Entering the following long form MODIFY command results in the following status displays (assuming that NETWK is the name of the started task running Entire Net-Work):

```
MODIFY NETWK,D STATS
NET0090I: BUFFER USAGE STATISTICS:
NET0091I: ASYNCH. BUFFERS:      000016 (= 24.2 %) OF 000064 K USED
NET0091I: LONG TERM BUFFERS:    000000 (=  0.4 %) OF 000064 K USED
NET0091I: SHORT TERM BUFFERS:   000000 (=  6.1 %) OF 001025 K USED
NET0091I: ATTACHED BUFFERS:     000000 (= 11.9 %) OF 000080 K USED
NET0091I: REQUEST QUEUE:        000000 (=  6.0 %) OF 000050 RQES USED
NET0087I: 0000010847 REQUESTS FROM LOCAL RQ
```

The following two commands are equivalent ways to terminate the Entire Net-Work session:

```
F NETWK,NETEND
P NETWK
```