Entire Net-Work Administration

Installation

Version 6.2.2

May 2011

Entire Net-Work

This document applies to Entire Net-Work Administration Version 6.2.2.

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

Table of Contents

1 Entire Net-Work Installation

This document explains how to install Entire Net-Work on supported mainframe platforms.

The Entire Net-Work Installation document is organized as follows:

●	*Installation Overview*	Contains general information that applies to all Entire Net-Work installations.
●	*Software AG Mainframe Product Licensing*	Describes Software AG's mainframe product license support and processing.
●	*z/OS Environments*	Provides information about installing and running Entire Net-Work in the z/OS operating system environment.
●	*z/VSE Environments*	Provides information about installing and running Entire Net-Work in the z/VSE operating system environment.
●	*BS2000/OSD Environments*	Provides information about installing and running Entire Net-Work in the BS2000/OSD operating system environment.
●	*Zap Naming Convention*	Describes the naming convention used for Entire Net-Work zaps.

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 Installation Overview

This section contains general information that applies to all Entire Net-Work installations.

Installation Checklist

The following checklist outlines the steps for installing Entire Net-Work:

1. Prepare the environment as described in the section specific to the operating system and the Simple Connection Line Driver.
2. Unload the installation libraries from the installation tape. Refer to the *Report of Tape Creation* for specific data set sequence numbers.
3. Prepare and install the product license file.
4. Prepare Entire Net-Work JCL and parameter statements.
5. Install the Adabas router, if necessary.
6. Complete any access method-related work.
7. Start Entire Net-Work and establish connectivity with the partner nodes.

Using System Maintenance Aid

If you use Software AG's System Maintenance Aid (SMA), refer to the System Maintenance Aid manual for information about the installation process.

The installation procedures described in this section correspond to the jobs that SMA creates to install the product.

If you do not use SMA, you can modify and use the sample JCL to unload the Entire Net-Work libraries from the installation tape. Sample JCL is provided in each of the platform-specific installation procedure sections.

The Entire Net-Work Installation Tape

The Entire Net-Work installation tape is a standard label tape. Refer to the *Report of Tape Creation* that accompanies the tape for the volume serial number, density, media type, data set names, and data set sequence numbers.

The tape contains the installation data sets, a data set required by SMA, and one or more data sets containing maintenance fixes. Only the load modules required to run the Entire Net-Work line drivers purchased for your environment are included on the tape.

Contents of the Release Tape

The following table describes most of the libraries included on the release tape. Once you have unloaded the libraries from the tape, you can change these names as required by your site, but the following lists the names that are delivered when you purchase Entire Net-Work.

Note: Some of the libraries listed below may not appear on your release tape. If this is the case, it is likely that an update to that library was not necessary for the release.

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
BTE*vrs*.LD*nn*	A Software AG internal library. 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
MLC*vrs*.JOBS	The sample job library for Software AG's common mainframe license check software. The *vrs* in the library name represents the *version* of the license check software, which is not necessarily the same as the version of Entire Net-Work.
MLC*vrs*.LOAD	The load library for Software AG's common mainframe license check software. The *vrs* in the library name represents the *version* of the license check software, which is not necessarily the same as the version of Entire Net-Work.
WAF*vrs*.JOBS	The sample z/OS job library for Network SAF Security. The *vrs* in the library name represents the *version* of Network SAF Security.
WAF*vrs*.LOAD	The z/OS load library for Network SAF Security. The *vrs* in the library name represents the *version* of Network SAF Security.
WAF*vrs*.SRCE	The z/OS source library for Network SAF Security. The *vrs* in the library name represents the *version* of Network SAF Security.
WAL*vrs*.JOBS	The sample z/OS job library for Adabas Limited Load Library, containing code for Adabas components shared by Adabas and other Software AG products. The *vrs* in the library name represents the *version* of Adabas Limited Load Library.
WAL*vrs*.LOAD	The z/OS load library for Adabas Limited Load Library, containing code for Adabas components shared by Adabas and other Software AG products. The *vrs* in the library name represents the *version* of Adabas Limited Load Library.
WAL*vrs*.SRCE	The z/OS source library for Adabas Limited Load Library, containing code for Adabas components shared by Adabas and other Software AG products. The *vrs* in the library name represents the *version* of Adabas Limited Load Library.
WAT*vrs*.LOAD	The z/OS load library for Entire Net-Work Adabas TCP/IP Option. The *vrs* in the library name represents the *version* of Entire Net-Work Adabas TCP/IP Option.
WCA*vrs*.LOAD	The z/OS load library for Entire Net-Work Administration, used by some of the Adabas add-on products. The *vrs* in the library name represents the *version* of Entire Net-Work Administration.

Library Name	Description
WCA*vrs*.SRCE	The z/OS source library for Entire Net-Work Administration, used by some of the Adabas add-on products. The *vrs* in the library name represents the *version* of Entire Net-Work Administration.
WCC*vrs*.LOAD	The z/OS load library for Entire Net-Work CTC Option. The *vrs* in the library name represents the *version* of Entire Net-Work CTC Option.
WCP*vrs*.LOAD	The z/OS load library for Entire Net-Work. The *vrs* in the library name represents the *version* of Entire Net-Work.
WCP*vrs*.SRCE	The z/OS source library for Entire Net-Work. The *vrs* in the library name represents the *version* of Entire Net-Work.
WDX*vrs*.LOAD	The z/OS load library for Entire Net-Work Limited Load. The *vrs* in the library name represents the *version* of Entire Net-Work Limited Load.
WSL*vrs*.LIBR	The z/VSE 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.
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.
WTC*vrs*.LIBR	The z/VSE load library for Entire Net-Work TCP/IP Option. The *vrs* in the library name represents the *version* of Entire Net-Work TCP/IP Option.
WTC*vrs*.LOAD	The z/OS load library for Entire Net-Work TCP/IP Option. The *vrs* in the library name represents the *version* of Entire Net-Work TCP/IP Option.
WTC*vrs*.MOD	The BS2000 load library for Entire Net-Work TCP/IP Option. The *vrs* in the library name represents the *version* of Entire Net-Work TCP/IP Option.
WXF*vrs*.LOAD	The z/OS load library for Entire Net-Work XCF Option. The *vrs* in the library name represents the *version* of Entire Net-Work XCF Option.

Adabas Requirements

This release of Entire Net-Work requires the Adabas 8 components supplied in Adabas Limited Load Library (WAL) and the SVC from Adabas 8.2.2 or later. These are distributed on the installation tape.

Important: You should always use the most current Adabas load library 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. Entire Net-Work Version 6.2.2 requires Adabas mainframe version 8.2.3 or later or WAL 8.2.3 or later. In situations where Adabas 8.2.2 or the WAL 8.2.2 library must be used, you must apply zap AU822061; if you do not, errors will result.

Software AG recommends that you concatenate the Entire Net-Work and Adabas 8 WAL with the Entire Net-Work library first in the sequence. The Adabas 8 router (that is, ADASVC in z/OS or z/VSE) must be installed on any node where Entire Net-Work is to be run, even if no servers are running on that node. Refer to the section *Installation Tape Information* in the *Entire Net-Work Release Notes* and the *Report of Tape Creation* provided with the Entire Net-Work installation tape for information about the WAL library modules and the version of Adabas required. Refer to the *Adabas Installation* documentation for information about installing the Adabas router in your environment.

Adabas client application programs need an Adabas link routine module suitable for the environment in which they run. For example, batch programs should be linked with ADAUSER, which loads current versions of ADALNK and supporting modules from the Adabas load library. TP environments usually require specialized link routine modules, such as ADALNC for CICS, which may require an additional license. For complete information about the Adabas link routine modules, refer to the *Adabas Installation* documentation.

Adabas SVC

The Adabas SVC is required in z/OS to provide communications between user programs, Entire Net-Work, and Adabas or other servers. If already installed for Adabas or any other Software AG server, it can and should also be used for Entire Net-Work. The Adabas SVC number is specified in the ADARUN SVC=nnn parameter.

Adabas 7 Data Conversion

Entire Net-Work Administration requires UES-enabled databases, which enable it to perform data conversion instead of Entire Net-Work.

Running Entire Net-Work

Starting the Entire Net-Work Job/Task

When execution JCL, including the ADARUN and Entire Net-Work parameter statements, is created for each node, and all access method-specific definitions and installation steps are completed, Entire Net-Work can be started.

The execution job or procedure for Entire Net-Work must be started on each of the participating nodes. Once started, Entire Net-Work determines the interregion communications environment on its node, connects to neighboring nodes through the defined links, and exchanges all relevant information with the other nodes during the "handshaking" process following link connection.

While the Entire Net-Work tasks are active and connected, user programs on any node can access all active targets on any other node, regardless of the targets' locations.

If a link fails or cannot be connected properly, Entire Net-Work searches for possible alternate routes to complete the call. Once a disconnected link becomes available again, it is automatically considered for traffic.

Specifying the Control Statements

There are five types of Entire Net-Work control statements:

- ADARUN control statements define the interregion communications environment. Some ADARUN parameters are the same as for Adabas; others are similar, but are designed specifically for Entire Net-Work. For more information, see the section *ADARUN Control Statements* in the *Entire Net-Work Reference Guide*
- NODE statements define the node's name and operating characteristics. For more information, see the section *Entire Net-Work NODE Statements* in the *Entire Net-Work Reference Guide*.
- DRIVER statements define the line driver type to be loaded. For more information, see the section *Entire Net-Work DRIVER Statements* in the *Entire Net-Work Reference Guide*.
- LINK statements define the links to other nodes. For more information, see the section *Entire Net-Work LINK Statements* in the *Entire Net-Work Reference Guide*

3 Software AG Mainframe Product Licensing

This document describes Software AG"s mainframe product licensing procedures, utilities, and license file.

Distributed License Libraries

All Software AG mainframe products are distributed with licensing libraries that contain the files necessary to license your Software AG mainframe software. This section describes these data sets. All licensing data sets have names starting with the characters "MLC".

z/OS Licensing Data Sets

The following licensing data sets are provided in z/OS environments:

Data Set Name	Description
MLC*vrs*.LOAD	Mainframe license check load library.
MLC*vrs*.JOBS	Mainframe license check example jobs.

z/VSE Licensing Data Sets

The following licensing data sets are provided in z/VSE environments:

Data Set Name	Description
MLC*vrs*.LIBR	Mainframe license check load library.
MLC*vrs*.LIBJ	Mainframe license check example jobs.

z/VM Licensing Data Sets

The following licensing data set is provided in z/VM environments:

Data Set Name	Description
MLC*vrs*.TAPE	Mainframe license check library.

Note: Software AG licensing software is not provided for VM/GCS environments.

BS2000 Licensing Data Sets

The following licensing data sets are provided in BS2000 environments:

Data Set Name	Description
MLC*vrs*.MOD	Mainframe license check load library.
MLC*vrs*.JOBS	Mainframe license check example jobs.

License File

You must install a valid license file on all mainframe platforms in which your Software AG mainframe product is installed. Your individual product license is shipped on the installation tape. It can also be shipped by e-mail, if desired. License files have file names in the format "*pppvr*.xml", where *ppp* is the three-character product code of the Software AG product (for example, ADA for Adabas, NAT for Natural, or WCP for Entire Net-Work) and *vr* represent the *major and minor version numbers* (respectively) of the product.

Once the license file is received, you must use native FTP commands to transfer it to your mainframe host before using it during installation.

Caution: Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

The license file is provided as an XML document (encoding is US-ASCII). This document can be viewed using a browsing tool or text editor on a PC. It contains text, which represents the licensing information and a digital signature, the license key. Among other things, it displays Software AG legal notices and copyright information. The product license is comprised of environmental information.

Important: Please note that the license file must remain in ASCII 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.

Transferring a License File from PC to a z/OS Host Using FTP

If a license file is supplied as an e-mail attachment, you must transfer the attached license file from the PC to the mainframe using native FTP commands.

Caution: Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

To transfer a license file from the PC to a mainframe z/OS host, perform the following steps:

1. Save the product license file e-mail attachment on your PC's hard disk.
2. Open a command prompt window. In the command prompt window, change to the directory where you saved the license file.
3. Start an FTP session for communication with the z/OS host using the following FTP command:

   ```
   ftp host-name
   ```

   where *host-name* is the name of the z/OS host.
4. Enter your z/OS host login ID and password.
5. Once the FTP session has been established, you can optionally delete the default working prefix. To determine what the default working prefix is, enter the following command:

   ```
   ftp>pwd
   ```

   To delete the default working prefix, enter the following command:

   ```
   ftp>cd ..
   ```

6. Specify the z/OS file system and data set prefix (for example, "ADABAS", "NATURAL", "NETWORK", or any other convenient prefix) for the license file. The following example sets the data set prefix to "ADABAS":

   ```
   ftp>cd 'ADABAS'
   ```

7. Switch to binary data mode (the license file must retain its format as ASCII during the transfer):

   ```
   ftp>binary
   ```

8 Specify that the data set for the license file must be written with RECFM=FB and LRECL=80.

```
ftp>quote site RECFM=FB LRECL=80 BLKSIZE=4000
```

RECFM=F is also supported for product license files.

9 Write the license as a data set on the z/OS system. For example, if the license file name is *ppp81.xml*, you might enter:

```
ftp>put ppp81.xml
```

This command will create a data set called "*`prefix`*.ADA81.XML", where *`prefix`* is the data set prefix you specified in Step 6. For example, if you requested a prefix of "ADABAS" in Step 6, the data set name would be "ADABAS.ADA81.XML". The data set will have RECFM=FB, LRECL=80, and the license information stored in the data set will be in ASCII format.

10 Stop your FTP session.

```
ftp>quit
```

Transferring a License File from PC to a z/VSE Host Using FTP

If a license file is supplied as an e-mail attachment, you must transfer the attached license file from the PC to the mainframe using native FTP commands.

Caution: Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

The FTP sample session described in these steps loads the license file onto a member of a z/VSE library. It is possible to load the license file onto a sequential z/VSE data set (with RECFM=F or FB and LRECL=80) instead. For more information, refer to your FTP documentation.

To transfer a license file from the PC to a mainframe z/VSE host, perform the following steps:

1 Save the product license file e-mail attachment on your PC's hard disk.

2 Open a command prompt window. In the command prompt window, change to the directory where you saved the license file.

3 Start an FTP session for communication with the z/VSE host using the following FTP command:

```
ftp host-name
```

where *`host-name`* is the name of the z/VSE host.

4 Enter your z/VSE host login ID and password.

5. Once the FTP session has been established, you can optionally delete the default working prefix. To determine what the default working prefix is, enter the following command:

   ```
   ftp>pwd
   ```

   To delete the default working prefix, enter the following command:

   ```
   ftp>cd ..
   ```

6. Point to the z/VSE library for the license file, (for example, "USERLIB"):

   ```
   ftp>cd USERLIB
   ```

7. Point to the z/VSE sublibrary for the license file, (for example, "MYSUBLIB"):

   ```
   ftp>cd MYSUBLIB
   ```

8. Switch to binary data mode (the license file must retain its format as ASCII during the transfer):

   ```
   ftp>quote type I
   ```

9. Specify that the data set for the license file must be written with RECFM=FB and LRECL=80.

   ```
   ftp>quote site recfm fb 80
   ```

   RECFM=F is also supported for product license files.

10. Write the license as a member in the libraries on the z/VSE system you identified in Steps 6 and 7. For example, if the license file name is ppp81.xml, you might enter:

    ```
    ftp>put ppp81.xml LICENSE.XML
    ```

    This command will write the license file ada81.xml to a member called "LICENSE.XML". If you requested a library and sublibrary of "USERLIB" and "MYSUBLIB" in Steps 6 and 7, the LICENSE.XML file would reside in the USERLIB.MYSUBLIB library. The data set will have RECFM=FB, LRECL=80, and the license information stored in the member will be in ASCII format.

11. Stop your FTP session.

    ```
    ftp>quit
    ```

Transferring a License File to a z/VM Host

If a license file is supplied as an e-mail attachment or on tape, you must transfer it to the mainframe. This section describes how to transfer the license file from tape and from e-mail (using FTP).

Note: Software AG licensing software is not provided for VM/GCS environments.

To transfer the product license file from tape to a source library, complete the following steps:

1 Refer to the *Report of Tape Creation* to accurately locate the license file on the tape.

2 If the tape is positioned somewhere other than the beginning of the tape, rewind it:

```
TAPE REW
```

3 Position the tape to the license file location, as shown in the *Report of Tape Creation*. For example, if the sequence number of ADA*`vr`*ZVM file, as shown by the *Report of Tape Creation*, is *`n`*, you must position over 3 n - 2 tape marks (that is, FSF 1 for the first data set, FSF 4 for the second, etc.).

4 Run the following z/VM commands in sequence to transfer the file off of the tape.

```
FILEDEF IN TAP1 (RECFM FB LRECL 80 BLKSIZE 3120
FILEDEF OUT DISK fn ft A (RECFM F LRECL 80 BLKSIZE 80
MOVEFILE IN OUT
```

where *`fn`* is the file name of the license file and *`ft`* is the file type of the license file.

To transfer the product license file e-mail attachment from Windows to a z/VM host, complete the following steps (make sure to switch to binary transfer and verify that the resulting data set has RECFM=F or FB and LRECL=80):

1 Save the product license file e-mail attachment on your PC's hard disk.

2 Open a command prompt window. In the command prompt window, change to the directory where you saved the license file.

```
cd dir
```

3 Start an FTP session with the z/VM host:

```
ftp host-name
```

where *`host-name`* is the name of the z/VM host.

4 Enter your FTP logon userid and password, as appropriate.

5 Switch the FTP data mode to binary:

```
binary
```

6 Request that the license data set to be written has RECFM=FB and LRECL=80:

```
quote site fix 80
```

7 Write the license file to the z/VM host. For example:

```
put fn.ft
```

where `fn` is the file name of the license file and `ft` is the file type of the license file.

The license file is written to your VM host.

8 Stop your FTP session:

```
quit
```

Caution: Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

Transferring a License File from PC to a BS2000 Host Using FTP

If a license file is supplied as an e-mail attachment, you must transfer the attached license file from the PC to the mainframe using native FTP commands.

Caution: Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

▶ **To transfer a license file from the PC to a BS2000 host, perform the following steps:**

Make sure to switch to binary transfer and verify that the resulting data set has file type SAM.

1 Save the product license file e-mail attachment on your PC's hard disk.

2 Open a command prompt window. In the command prompt window, change to the directory where you saved the license file.

```
cd dir
```

3 Start an FTP session for communication with the BS2000 host using the following FTP command:

```
ftp host-name
```

where `host-name` is the name of the BS2000 host.

4 Enter your FTP login ID and password.

5 Switch the FTP data mode to binary:

```
binary
```

6 Request that the license data set has a file type of SAM:

```
quote file *,fcbtype=sam
```

7 Write the license file to the BS2000 host. For example, to write the *ppp81.xml* license file, you would enter:

```
put ppp81.xml
```

The license file is written to your BS2000 host.

8 Stop your FTP session:

```
quit
```

License File Installation

During the mainframe product installation process, the license file is loaded and processed as required by your Software AG mainframe product. Refer to the appropriate installation procedure for your Software AG mainframe product for further information.

Product License Check FAQs

1. **Why has Software AG introduced a license check for products on the mainframe?**
   Software AG wants to ensure that customers run our software products only on mainframe machines for which they have valid and sufficient product license. This ensures that we have better control of the use of our software products. On other platforms (UNIX, Windows), the product licensing has been established for many years.
2. **What information is contained in a product license?**
   A product license is a sequential file with US-ASCII text in XML format containing the following items:
   - Software AG header
   - customer information (Name, ID)

- encrypted license key
- license expiration date (or unlimited)
- product information (product code, version, name)
- environment information (operating system type, CPU ID, system name, capacity)

Caution: Any modification of the license file will invalidate the digital signature and the license key 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.

3. **What exactly is meant by the CPU ID?**
IBM defines the CPU ID as the "central processing complex node descriptor sequence number". It is the unique hexadecimal machine serial number without the machine model number.

z/OS

On z/OS systems, information about the CPU can be displayed on the operator console by entering the command:

```
D M=CPU
```

For example, suppose the console output from this command contains these lines:

```
EE174I 16.38.50 DISPLAY M 951
PROCESSOR STATUS
ID  CPU                  SERIAL
00  +                     0FA10E2096
01  +                     0FA10E2096
CPC ND = 002096.S07.IBM.83.00000007A10E
CPC SI = 2096.V03.IBM.83.000000000007A10E
....
```

In this example, the CPU ID is "7A10E", as highlighted in the example above.

z/VSE

On z/VSE systems, information about the CPU can be displayed on the operator console by entering the operator command:

```
sir
```

For example, suppose the output from this command contains this line:

```
...
AR 0015 PROCESSOR = IBM 2096-V03 83 (7A10E83)  LPAR = DAEX      No. = 0007
....
```

The CPU ID is highlighted in the example.

BS2000

On BS2000/OSD systems, the CPU ID (8-byte hexadecimal) consists of the machine serial number, the processor ID and the machine model number. The processor ID in the second byte of the CPU ID is ignored for the license check. Information about the CPU can be displayed by entering the following BS2000/OSD command:

```
/SHOW-SYSTEM-INFORMATION INFORMATION=*CPU-ID-LIST
```

For example, suppose the output from this command contains these lines:

```
...
%CONFIGURATION              = 7.500- S140-20A      «
%CPU-ID-LIST :  ADR   0     = 1D02301375000000 «
...
```

The most relevant information is shown in these lines. The first value identifies the machine type as "7.500- S140-20A", the second value identifies the physical CPU ID as "1D02301375" (trailing zeros can be omitted).

z/VM

On z/VM systems, the CPU ID can be determined entering the command:

```
q cpu
```

Suppose the output from this command is:

```
CPUID = FF07A10E20968000
```

The first byte will always be "FF", followed by the CPU-ID (highlighted in the example).

4. **Which mainframe products use the Software AG common mainframe license routines?**
   - Adabas (version 8.1 SP4 or later)
   - Com-plete (planned)
   - Entire Net-Work (version 6.2 SP1 or later)
   - Entire System Services (planned)

- Natural (planned)
- webMethods EntireX (planned)

5. **How is the product license file installed?**
   The product license file is supplied on the individual customer installation tape or separately via an e-mail attachment. For information on its installation, refer to the installation documentation for your product. For information on transferring the license file from a PC to a z/OS, z/VSE, z/VM, or BS2000 host, read the various *Transferring a License File from PC to a xxx Host* sections , elsewhere in this section.

   In addition, steps for actually installing the license file are provided in the installation steps of your software product.
6. **How can I read the product license file?**
   - The product license file can be read on a PC using an XML editor (file type *.xml*) or any PC-based text editor.
   - The license utility, LICUTIL, provided with the Software AG mainframe license check software includes a function that allows you to display the license (Display License function). For more information about the license utility, read ***The License Utility: LICUTIL***, elsewhere in this section.
   - When Natural is installed and runs, the license information can be displayed using function "L" of the Natural SYSTP utility.
   - When Adabas, Com-plete, Entire Net-Work, Entire System Services, and webMethods EntireX are installed and run, they print a formatted license file into their logging files.
7. **How can I display machine-specific data?**
   Use the license utility, LICUTIL, to display machine-specific data. For more information about the license utility, read ***The License Utility: LICUTIL***, elsewhere in this section.
8. **When is the product license checked?**
   The product license is checked during every time the product is initialized. In addition, the product license is checked once a day.

   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.
9. **Which items of a product license are checked?**
   Product license items are divided into three logical license groups (License Key, Product-Specific Information, and Machine-Specific Information) and checked in that order. If an inconsistency in any one of these groups is detected by the license checker, the succeeding items of that logical license group are also checked. Then the check terminates with corresponding error messages. For example, if the license key expires, the license key check terminates with an error before the license checker checks any product-specific or machine-specific information.

   The following table identifies the product license items comprising each logical group:

Checking Order	Logical License Group Name	Product License Items Included
1	License Key	The encrypted license key
2	Product-Specific Information	■ The expiration date (if any) ■ The operating system (z/OS, z/VSE, z/VM, or BS2000/OSD) ■ The product code (for example, ADA, COm, EXX, NAT, or WCP) ■ The product version, in either "`v.r`" or "`v.r.s`" format. For further information on product versions, refer to the description of "`v.r`" or "`v.r.s`" in the product documentation Glossary.
3	Machine-Specific Information	■ The machine CPU ID ■ The machine capacity (only in z/OS, z/VSE, and z/VM environments) ■ The logical partition

10. **Does the Software AG mainframe license checker software support a disaster recovery scenario?**
    Yes. Please contact your Software AG sales representative to obtain a license file which is able to support disaster recovery scenarios. The parameter `DisasterRecoveryCPUIDs` in this license file allows you to specify one or more CPU IDs of the machines to be used when your production machine fails. If you do not have these CPU IDs available, it is possible to specify the value "Unknown". In this case, the license checker issues a warning message when a disaster recovery scenario is assumed.

    In a disaster recover scenario, no machine data besides the CPU ID is checked.
11. **What happens if the product license expires or is incorrect, insufficient, or not installed?**
    The reaction varies by Software AG mainframe product. For more information, refer to the documentation for the product.

    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.
12. **How can I get a new product license file if the delivered license file is insufficient for my environment?**
    Contact your Software AG sales representative to get a correct product license file according to your contract. It can be sent by e-mail either in ASCII format or in the converted assembler input format.
13. **Do I need separate product licenses for different machines?**
    No, it is possible to have multiple CPU IDs defined in one license file.

Using The License Utility: LICUTIL

The license utility, LICUTIL, is a utility provided with Software AG's mainframe license checking software. It can perform the following functions:

- Check the license file. LICUTIL is mainly intended to be used to check the license file during the installation of the product. The original license file must be provided as input along with the product code and version. If the license file is not valid, appropriate error messages are returned.
- Display the license file and machine data. LICUTIL is able to display the contents of the license file and machine data, such as the CPU ID, LPAR, or MSU information.
- Convert the license file into an assembler source module. LICUTIL can be used to convert the license file into an assembler source module that can then be used to assemble and link a license module. For more information about using license key modules, refer to the documentation provided with your Software AG product.

The following table briefly describes the functions and parameters in the license utility. More detailed information of each function is provided later in this section:

Function	Parameters	Description
CHECK	---	Check the product license.
DISPLAY	`ALL`	Display the product license items and machine data.
	`LICENSE`	Display only the product license items.
	`MACHINE`	Display only the machine data.
END | EXIT	---	Stop the license utility.
HELP	---	Produce a list of available LICUTIL functions and the syntax of the functions.
MAKE	---	Create an assembler source module from the license file.

SET	`DATE=`*`yyyy-mm-dd`*
	or
	`DATE=` *`yyyymmdd`*
		Set the date for a license check. The default is today.

	`FORMAT= {FREE | LEFT | [STRUCTURED]`	Produce output without XML tags (FREE), with each XML tag in a new line (LEFT), or with each XML tag indented (STRUCTURED). The default is FREE.
	`PRODVER=`*`pppv.r.s`*	Set the product code and version, where *`ppp`* is the Software AG 3-character product code and *`v.r.s`* is the three-digit version number.

Error messages for the license utility all begin with the prefix MLC. They are documented with the rest of your Software AG product messages.

This section covers the following topics:

- CHECK Function: Check the Product License
- DISPLAY Function: Display License Items and Machine Data
- END/EXIT Functions: Terminate the License Utility
- HELP Function: View LICUTIL Help
- MAKE Function: Convert the License File to Assembler Source
- SET Function: Set LICUTIL Options
- JCL Examples and Requirements

CHECK Function: Check the Product License

The CHECK function can be used to check the license file against the product's execution environment. There are no parameters, however to run the CHECK function successfully, you must have previously specified the **SET PRODVER** function to provide the product code and version number.

The following items are checked, in the following order:

1. License key
2. Product-specific data, including the product code, version, operating system, and expiration date.
3. Machine-specific data, including the CPU IDs, the capacity (on only z/OS, z/VSE and z/VM systems), and the logical partition.

The syntax of the CHECK function is:

Sample JCL or JCS is provided for the CHECK function in member LICCHECK of the Software AG mainframe license JOBS data set (MLC*vrs*.JOBS).

Example:

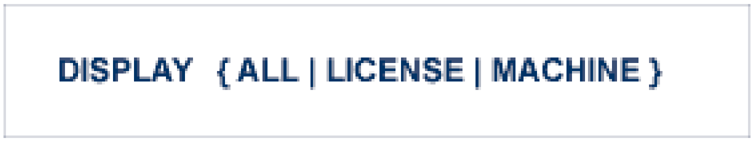
In the following example, the product code and version are set to ADA8.1.4 (Adabas 8.1.4) and then the product license is checked.

```
SET PRODVER=ADA8.1.4
CHECK
```

DISPLAY Function: Display License Items and Machine Data

Use the DISPLAY function to display the contents of the product license items (license file, product-specific information, and machine-specific information) and to display machine data (CPU ID, machine type, model ID, LPAR, MSU) about the machine on which the utility is being run.

The syntax of the DISPLAY function is:



```
DISPLAY { ALL | LICENSE | MACHINE }
```

The parameters are described in the following table. At least one parameter is required.

Parameters	Description
`ALL`	Display the product license items and machine data.
`LICENSE`	Display only the product license items.
`MACHINE`	Display only the machine data.

Sample JCL or JCS is provided for the DISPLAY function in member LICMDATA of the Software AG mainframe license JOBS data set (MLC*vrs*.JOBS).

Examples

In the following example, both license and machine data is displayed:

```
DISPLAY ALL
```

In the following example, only license data is displayed:

```
DISPLAY LICENSE
```

In the following example, the license data is displayed in structured format:

```
SET FORMAT=STRUCTURED
DISPLAY LICENSE
```

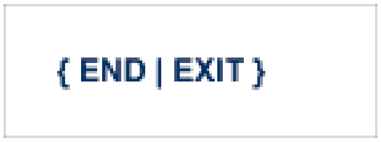
In the following example, only machine data is displayed:

```
DISPLAY MACHINE
```

END/EXIT Functions: Terminate the License Utility

Use the END or EXIT functions to terminate the license utility. These functions should be used if LICUTIL is running interactive in environments other than batch (for example TSO under z/OS).

The syntax of the END and EXIT functions is:

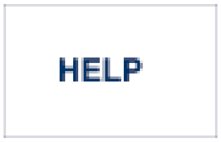


```
{ END | EXIT }
```

HELP Function: View LICUTIL Help

Use the HELP function to review a list of the available LICUTIL functions and their syntax.

The syntax of the HELP function is:



```
HELP
```

Sample Output

Here is some sample output from this function:

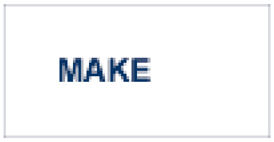
```
>>>HELP
LICUTIL reads a Software AG product license from SYSUT1.
It has the following functions:
    CHECK   : Check license read from SYSUT1
    DISPLAY : Display license and machine
    END     : Terminate LICUTIL
    MAKE    : write license for assembly to SYSUT2
    SET     : sets execution defaults

SET keyword=value accepts the following keyword(s):
    DATE=YYYY-MM-DD or YYYYMMMDD - Set date for license check (default TODAY)
    PRODVER=zzzv.r.s             - Set product code zzz and version release
                                   SM-level v.r.s for license check
    FORMAT=LEFT/STRUCTURED/FREE  - Set format of license display
```

MAKE Function: Convert the License File to Assembler Source

The MAKE function can be used to convert the license file into an assembler source. After assembling this source (and linking it), the resulting license module can be used by defined Software AG products (for example Adabas or Natural) instead of the license files. For more information, please read the documentation for your specific Software AG mainframe product.

The syntax of the MAKE function is:



```
MAKE
```

Sample JCL or JCS is provided for the CHECK function in member LICMAKE of the Software AG mainframe license JOBS data set (MLC*vrs*.JOBS).

Sample Output

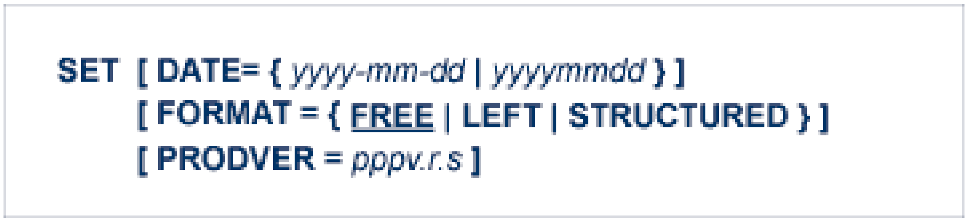
Here is some sample output from this function when run on a z/OS system:

```
MLC9006 License converted to assembler output. 042 records written
        to  SYSUT2.
```

SET Function: Set LICUTIL Options

Use the SET function to set options for the license utility.

The syntax of the SET function is:



```
SET  [ DATE= { yyyy-mm-dd | yyyymmdd } ]
     [ FORMAT = { FREE | LEFT | STRUCTURED } ]
     [ PRODVER = pppv.r.s ]
```

The parameters are described in the following table. At least one parameter is required.

Parameters	Description
`DATE`	Sets the date for the license check. The default is today. The format of the date can be either *yyyy-mm-dd* or *yyyymondd*, where *yyyy* is the four-character year, *mm* is the two-digit month, *mon* is the three-character month abbreviation, and *dd* is the two-digit day.
`FORMAT`	Sets the display format for the **DISPLAY** function. There are three different formats: ■ Specify FREE (the default) to produce output with XML tags. ■ Specify LEFT to produce output with XML tags ■ Specify STRUCTURED to produce output with each XML tag indented.
`PRODVER`	Sets the product code and version for the **CHECK** function. The format in which the product code and version should be specified is *pppv.r.s*, where *ppp* is the three-character Software AG product code (such as ADA, NAT or WCP) and *v.r.s* is the three-digit version number.

Sample JCL or JCS is provided for the DISPLAY function in member LICCHECK of the Software AG mainframe license JOBS data set (MLC*vrs*.JOBS).

Example

In the following example, the format of the DISPLAY output will not include XML tags and the product and version are set as Adabas 8.1.4:

```
SET FORMAT=FREE
SET PRODVER=ADA8.1.4
```

JCL Examples and Requirements

Sample JCL of the license utility functions can be found in the following members of the Software AG mainframe license example libraries (MLC*vrs*.JOBS in z/OS and BS2000 environments, MLC*vrs*.LIBJ in z/VSE environments, and MLC*vrs*.TAPE in VM environments).

Member	Functions Included	Demonstrates how to...

LICCHECK	SET
	PRODVER=PPP*v.r.s*
	DISPLAY
	CHECK
		Check a license file with LICUTIL.

LICMAKE	MAKE	Convert the license file into an assembler source.
LICMDATA	DISPLAY MACHINE	Display machine data. This function can be used to send Software AG the required data for a license file.

This section covers the following topics:

- z/OS and z/VM JCL Requirements
- z/VSE JCL Requirements

- BS2000/OSD JCL Requirements

z/OS and z/VM JCL Requirements

The following file assignments are used by the license utility in z/OS and z/VM environments:

DD Name	Description
SYSIN	Input of LICUTIL control parameters.
SYSPRINT	Output of LICUTIL run.
SYSUT1	Input of original license file.
SYSUT2	Output of license converted to assembler source.

The following table shows which file assignments are required in the JCL by each LICUTIL function (an X in a table cell indicates that the file assignment is required; a blank cell indicates that file assignment is not required):

License Utility Function	DD Name			
	SYSIN	SYSPRINT	SYSUT1	SYSUT2
CHECK	X	X	X	
DISPLAY ALL	X	X	X	
DISPLAY LICENSE	X	X	X	
DISPLAY MACHINE	X	X		
MAKE	X	X	X	X
HELP	X	X		

z/VSE JCL Requirements

The following file assignments are used by the license utility in z/VSE environments:

File Assignment	Description
SYSIPT	Input of LICUTIL control parameters and the input of original license file. **Note:** The license file has to be stored in a sublibrary.
SYSLIST	Output of LICUTIL run.
SYSPCH	Output of license converted to assembler source.

The following table shows which file assignments are required in the JCL by each LICUTIL function (an X in a table cell indicates that the file assignment is required; a blank cell indicates that file assignment is not required):

License Utility Function	File Assignments		
	SYSIPT	SYSLIST	SYSPCH
CHECK	X	X	
DISPLAY ALL	X	X	
DISPLAY LICENSE	X	X	
DISPLAY MACHINE	X	X	
MAKE	X	X	X
HELP	X	X	

BS2000/OSD JCL Requirements

The following file assignments are used by the license utility in BS2000 environments:

System File	Description
SYSDTA	Input of LICUTIL control parameters.
SYSOUT	Output of LICUTIL run.

Link Name	Description
SYSUT1	Input of original license file.
SYSUT2	Output of license converted to assembler source.

The following table shows which file assignments are required in the JCL by each LICUTIL function (an X in a table cell indicates that the file assignment is required; a blank cell indicates that file assignment is not required):

License Utility Function	File Assignment			
	SYSDTA	SYSOUT	SYSUT1	SYSUT2
CHECK	X	X	X	
DISPLAY ALL	X	X	X	
DISPLAY LICENSE	X	X	X	
DISPLAY MACHINE	X	X		
MAKE	X	X	X	X
HELP	X	X		

4 z/OS Environments

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

Library Requirements

The source library requires one cylinder of 3390 DASD space. The load library requires four cylinders. See the section *Installation Tape Information* in the *Entire Net-Work Release Notes* 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***, elsewhere in this guide.

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***, elsewhere in this guide.
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* in the *Entire Net-Work Reference Guide*.

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.

Caution: 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* in the *Entire Net-Work Reference Guide*.
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* in the *Entire Net-Work Reference Guide*.

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* in the *Entire Net-Work Reference Guide*.

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
```

5 z/VSE Environments

This section provides information about installing and running Entire Net-Work in the z/VSE operating system environment.

Installation Procedure

To install Entire Net-Work on a z/VSE system, perform the following steps for each machine. Sample JCS can be found in the Entire Net-Work source library.

- Step 1: Define an Installation Library
- Step 2: Copy the Tape Contents to Disk
- Step 3: Specify a z/VSE Partition to Run Entire Net-Work
- Step 4: Prepare and Install the Product License File
- Step 5: Install the Adabas SVC (optional)
- Step 6: Customize the NETnnLIB Procedure
- Step 7: Modify the Entire Net-Work Startup Parameters
- Step 8: Perform the Line Driver-Dependent Installations
- Step 9: Prepare Access Method-Specific Definitions
- Step 10: Apply Corrective Maintenance

Step 1: Define an Installation Library

Define an Entire Net-Work installation library. Allocate DASD space; then define and restore the Entire Net-Work libraries from the installation tape.

If all of the sublibraries are restored, approximately five cylinders of 3390 DASD space is required. Refer to the *Report of Tape Creation* .

Step 2: Copy the Tape Contents to Disk

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 the data sets `.LIBJ`, `.LIBR` and `.LICS` from tape to disk. All other data sets can be installed directly from the tape.

You will then need to perform the individual installation procedure for each component to be installed.

- Step 1: Copy Data Set COPYTAPE.JOB from Tape to Disk
- Step 2: Modify COPYTAPE.JOB

- Step 3: Submit COPYTAPE.JOB

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

The data set `COPYTAPE.JOB` contains the JCL required to copy the data sets `.LIBJ`, `.LIBR` and `.LICS` from tape to disk. Copy `COPYTAPE.JOB` to your disk by using the following sample JCL:

```
* $$ JOB JNM=LIBRCAT,CLASS=0,                                          +
* $$ DISP=D,LDEST=(*,UID),SYSID=1
* $$ LST CLASS=A,DISP=D
// JOB LIBRCAT
* *****************************************
*     STORE COPYTAPE.JOB IN LIBRARY
* *****************************************
// ASSGN SYS004,nnn
// MTC REW,SYS004
// MTC FSF,SYS004,4
ASSGN SYSIPT,SYS004
// TLBL IJSYSIN,'COPYTAPE.JOB'
// EXEC LIBR,PARM='MSHP; ACC S=lib.sublib'
/*
// MTC REW,SYS004

ASSGN SYSIPT,FEC
/*
/&
* $$ EOJ
```

where:

`nnn` is the tape address
`lib.sublib` is the library and sublibrary in which `COPYTAPE.JOB` is to be stored

Step 2: Modify COPYTAPE.JOB

Modify `COPYTAPE.JOB` according to your local naming conventions and set the disk space parameters.

Step 3: Submit COPYTAPE.JOB

Submit `COPYTAPE.JOB` to copy the data sets `.LIBJ`, `.LIBR` and `.LICS` from tape to your disk.

Step 3: Specify a z/VSE Partition to Run Entire Net-Work

Specify a z/VSE partition to run Entire Net-Work.

The minimum space requirement is 128KB (real) in the ALLOCR for this Entire Net-Work partition. See the section ***Partition/Address Space Requirements*** .

Note: In the case of z/VSE systems, the real storage should be specified by the following job control statement:

```
// SETPFIX LIMIT=nnnk
```

Step 4: Prepare and Install the Product License File

Prepare and install the product license file as described in ***Prepare and Install the License File***, elsewhere in this section.

Step 5: Install the Adabas SVC (optional)

In systems where Adabas is not already installed, install the Adabas SVC using the ADASIP program. For more information, see the section ***Adabas SVC* in this section.**.

For information about the Adabas z/VSE communication environment, refer to the *Adabas Installation Manual*.

Step 6: Customize the NETnnLIB Procedure

Customize the procedure NET*nn*LIB before cataloging it in the procedure library.

The following specific items must be customized:

- File IDs for the libraries
- Volumes for the libraries
- Be sure to include appropriate LIBDEF references for user sublibraries, especially the library containing the NETLIC license file. The licensing component MLC*vrs* must also be added to the LIBDEF SEARCH chain for phases.

Step 7: Modify the Entire Net-Work Startup Parameters

Modify the Entire Net-Work startup parameters. See the section *Entire Net-Work Parameter Statements* in the *Entire Net-Work Reference Guide*.

Step 8: Perform the Line Driver-Dependent Installations

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

Step 9: Prepare Access Method-Specific Definitions

Prepare the required access method-specific definitions.

Step 10: Apply Corrective Maintenance

Apply corrective maintenance if necessary (refer to the *Report of Tape Creation*).

Caution: 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.

Preparing and Installing the License File

The product license file is supplied on the individual customer installation tape or separately via an e-mail attachment. If the license file is provided on an installation tape, you can follow the instructions in this step to install the license file. If the license file is supplied via an e-mail attachment, you must first transfer the license to z/VSE, as described in ***Transferring a License File from PC to a z/VSE Host Using FTP***, elsewhere in this guide and then you can install it, as described in this step.

Installing the license file.

In z/VSE environments, the product license file can be installed either as a phase or as a library member.

To install the product license file as a phase, complete the following steps:

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

2 If you loaded your Entire Net-Work license file to a library, review and modify the following JCS, adjusting the library and volume specifications as appropriate for your site.

Note: In the sample JCS shown in this step, the standard label area is assumed to contain label information for library USERLIB. You can change this as appropriate for your library.

```
* ********************************************************************* 000100
*                                                                     *
*  This job assembles/link-edits your NET-WORK License NETLIC         *
*                                                                     *
*  Note: The Net-Work license file must have RECFM FB and LRECL=80    *
*        in order to be processed by NETLICAV.                        *
*                                                                     *
*        If received separately and moved to the machine with FTP     *
*        use the binary option such that the file is not converted    *
*        to EBCDIC                                                    *
*                                                                     *
*  This sample reads the Net-Work license file from a VSE library     *
*  member.                        *
*                                                                     *
*  In the example below, the standard label area is assumed to        *
*  contain label information for library USERLIB.                     *
*                                                                     *
*  The first step produces an assembler input for the Net-Work license*
*  The second step assembles the license and creates a load module    *
*  NETLIC.                                                            *
*                                                                     *
* ********************************************************************* 000100
* Replace the following symbols in this manner through the whole JCL: * 000200
*                                                                     * 000300
*     Before:             After:                                      * 000400
*                                                                     * 000500
*         $+      ------>     /+                                      * 000600
*         $*      ------>     /*                                      * 000600
*         $&      ------>     /&                                      * 000700
*         X $$    ------>     * $$                                    * 000800
*                                                                     * 000900
* ********************************************************************* 001000
*                                                                     * 000900
* Edit the following sample JCS to your own library and file          * 001400
* specifications and run the job.                                       001400
*                                                                     * 000900
* ********************************************************************* 001000
X $$ JOB JNM=NETLICAV,DISP=D,CLASS=0
```

```
X $$ LST DISP=D,CLASS=A
// JOB NETLICAV  GENERATE AN Net-Work LICENSE KEY
// DLBL SAGLIC,'SAG.MLCvrs.VSE.LIBRARY'
// EXTENT SYS018
// ASSGN SYS018,DISK,VOL=vollib,SHR
// LIBDEF *,SEARCH=(SAGLIC.MLCvrs),TEMP
// DLBL IJSYSPH,'WORK.FILE.SYSPCH',0
// EXTENT SYSPCH,volpch,1,0,pchsta,pchtrk
ASSGN SYSPCH,DISK,VOL=volpch,SHR
// EXEC LICUTIL,SIZE=LICUTIL
MAKE
X $$ SLI MEM=LICENSE.XML,S=USERLIB.sublib
$+
END
$*
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'WORK.FILE.SYSPCH',0
// EXTENT SYSIPT,volpch,1,0,pchsta,pchtrk
ASSGN SYSIPT,DISK,VOL=volpch,SHR
// DLBL IJSYSPH,'WORK.FILE.SYSPCH2',0
// EXTENT SYSPCH,volpc2,1,0,pc2sta,pc2trk
ASSGN SYSPCH,DISK,VOL=volpc2,SHR
// OPTION DECK
// EXEC ASMA90,SIZE=1024K
$*
CLOSE SYSIPT,SYSRDR
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'WORK.FILE.SYSPCH2',0
// EXTENT SYSIPT,volpc2,1,0,pc2sta,pc2trk
ASSGN SYSIPT,DISK,VOL=volpc2,SHR
// DLBL SAGLIB,'WCP621.Vvrs.VSE.LIBRARY'
// EXTENT SYS018
// ASSGN SYS018,DISK,VOL=vollib,SHR
// EXEC LIBR,PARM='ACC S=SAGLIB.usrlib; CATALOG NETLIC.OBJ  R=Y'
$*
CLOSE SYSIPT,READER
// LIBDEF OBJ,SEARCH=(SAGLIB.usrlib),TEMP
// LIBDEF PHASE,CATALOG=SAGLIB.usrlib
$*
// OPTION CATAL
 PHASE NETLIC,*
 MODE AMODE(24) RMODE(24)
 INCLUDE NETLIC
 ENTRY NETLIC
// EXEC LNKEDT
$*
// EXEC LISTLOG
$*
// JOB RESET
* ********************************************************
*      RESET SYSIPT AND SYSPCH
* ********************************************************
```

```
ASSGN SYSIPT,READER
ASSGN SYSPCH,PUNCH
$*
$&
X $$ EOJ
```

If you loaded your Entire Net-Work license file to a data set, modify the following sample JCS instead.

```
* *********************************************************************
*                                                                     *
*  This job assembles/link-edits your NET-WORK License NETLIC         *
*                                                                     *
*  Note: The Net-Work license file must have RECFM FB and LRECL=80    *
*        in order to be processed by WCPLICAV.                        *
*                                                                     *
*        If received separately and moved to the machine with FTP     *
*        use the binary option such that the file is not converted    *
*        to EBCDIC                                                    *
*                                                                     *
*  This sample reads the Net-Work license file from a sequential disk *
*  data set.
*                                                                     *
*  In the example below, the standard label area is assumed to        *
*  contain label information for library USERLIB.                    *
*                                                                     *
*  The first step produces an assembler input for the Net-Work license*
*  The second step assembles the license and creates a load module    *
*  NETLIC.                                                            *
*                                                                     *
* *********************************************************************
* Replace the following symbols in this manner through the whole JCS: *
*                                                                     *
*      Before:             After:                                     *
*                                                                     *
*         $*      ------>    /*                                       *
*         $&      ------>    /&                                       *
*         X $$    ------>    * $$                                     *
*                                                                     *
* *********************************************************************
*                                                                     *
* Edit the following sample JCS to your own library and file          *
* specifications and run the job.                                     *
*                                                                     *
* *********************************************************************
X $$ JOB JNM=NETLICAV,DISP=D,CLASS=0
X $$ LST DISP=D,CLASS=A
// JOB NETLICAV  GENERATE AN NET-WORK LICENSE KEY
// UPSI 1
// DLBL INPSEQ,'WCP621.Vvrs.LICENSE'
// EXTENT SYS018
// ASSGN SYS018,DISK,VOL=vollic,SHR
// EXEC DITTO
$$DITTO SL FILEIN=INPSEQ,
$$DITTO    LIBOUT=USERLIB.usrsublb,MEMBEROUT=LICENSE.WORKFILE
/*
// UPSI 0
// DLBL SAGLIC,'SAG.MLCvrs.VSE.LIBRARY'
// EXTENT SYS018
// ASSGN SYS018,DISK,VOL=vollib,SHR
// LIBDEF *,SEARCH=(SAGLIC.MLCvrs),TEMP
// DLBL IJSYSPH,'WORK.FILE.SYSPCH',0
// EXTENT SYSPCH,volpch,1,0,pchsta,pchtrk
ASSGN SYSPCH,DISK,VOL=volpch,SHR
// EXEC LICUTIL,SIZE=LICUTIL
MAKE
X $$ SLI MEM=LICENSE.WORKFILE,S=USERLIB.usrsublb
$+
END
$*
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'WORK.FILE.SYSPCH',0
// EXTENT SYSIPT,volpch,1,0,pchsta,pchtrk
ASSGN SYSIPT,DISK,VOL=volpch,SHR
// DLBL IJSYSPH,'WORK.FILE.SYSPCH2',0
// EXTENT SYSPCH,volpc2,1,0,pc2sta,pc2trk
ASSGN SYSPCH,DISK,VOL=volpc2,SHR
// OPTION DECK
// EXEC ASMA90,SIZE=1024K
$*
CLOSE SYSIPT,SYSRDR
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'WORK.FILE.SYSPCH2',0
// EXTENT SYSIPT,volpc2,1,0,pc2sta,pc2trk
ASSGN SYSIPT,DISK,VOL=volpc2,SHR
// DLBL SAGLIB,'WCP621.Vvrs.VSE.LIBRARY'
// EXTENT SYS018
// ASSGN SYS018,DISK,VOL=vollib,SHR
// EXEC LIBR,PARM='ACC S=SAGLIB.usrlib; CATALOG NETLIC.OBJ  R=Y'
$*
CLOSE SYSIPT,READER
// LIBDEF OBJ,SEARCH=(SAGLIB.usrlib),TEMP
// LIBDEF PHASE,CATALOG=SAGLIB.usrlib
$*
// OPTION CATAL
 PHASE NETLIC,*
 MODE AMODE(24) RMODE(24)
 INCLUDE NETLIC
 ENTRY NETLIC
// EXEC LNKEDT
$*
// EXEC LISTLOG
$*
```

```
// JOB RESET
* *****************************************************
*      RESET SYSIPT AND SYSPCH
* *****************************************************
ASSGN SYSIPT,READER
ASSGN SYSPCH,PUNCH
$*
$&
X $$ EOJ
```

3. Submit the job you modified in the previous step.

   The job generates your Entire Net-Work license in NETLIC.PHASE. It assumes that NETLIC.PHASE will be in a user sublibrary. If a user sublibrary is chosen for NETLIC.PHASE, this sublibrary must be included in the LIBDEF search chain in your Entire Net-Work startup JCS. You may find it more convenient to place NETLIC.PHASE directly into the Entire Net-Work WCP*vrs* sublibrary, to avoid the need to define additional libraries. During initial testing, Software AG recommends using a user sublibrary.

To install the product license file as a library member, complete the following steps:

1. Verify that the 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.
2. Make sure any previously-created NETLIC phase is inaccessible in the Entire Net-Work load library being used by the Entire Net-Work startup jobs. Entire Net-Work first tries to load NETLIC and if unsuccessful it reads from DDLIC.
3. Provide all Entire Net-Work startup jobs with a DLBL statement in the following format:

   ```
   // DLBL DDLIC,'/libname/sublb/memname.memtype'
   ```

   where *libname* is the Librarian name of the library, *sublib* is the name of the sublibrary, *memname* is the license member name, and *memtype* is the license member type.

To install the product license file as a sequential data set, complete the following steps:

1. Verify that the license file is stored in a sequential file (with RECFM=F or FB and LRECL=80), taking care to preserve its format as ASCII.
2. Make sure any previously created NETLIC phase is inaccessible in the Entire Net-Work load library being used by the Entire Net-Work startup jobs. Entire Net-Work first tries to load NETLIC and, if unsuccessful, it reads from DDLIC.
3. Provide all Entire Net-Work startup jobs with DLBL, EXTENT and ASSGN statements in the following format:

```
// DLBL DDLIC,'network.license.file'
// EXTENT SYSnnn
// ASSGN SYSnnn,DISK,VOL=volser,SHR
```

where `network.license.file` is the physical file name, `nnn` is an unused logical unit, and `volser` is the volume serial on which the license file resides.

Sample JCS

This section provides sample JCS for:

- defining the Entire Net-Work library as a non-VSAM library
- defining the Entire Net-Work library as a VSAM library
- restoring the Entire Net-Work installation tape
- defining the Entire Net-Work catalog procedure
- installing the Adabas SVC
- unloading the ZAPS data sets

User Substitution Variables

The sample JCS in this section contains the following user substitution variables:

Variable	Description
vrs	The product version, revision, and system maintenance (SM) level
vvvvvv	Specified volume for the Entire Net-Work library
ssss	Starting track number
nnnn	Number of tracks or cylinders for the Entire Net-Work library
cuu	Channel unit address
ccccccc	VSAM catalog name
tt	Number of tape marks to space forward

Non-VSAM Library

The following sample JCS can be used to define the Entire Net-Work library as a non-VSAM library:

```
* $$ JOB JNM=LIBRDEF,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB LIBRDEF
// DLBL SAGLIB,'NETWRK.vrs.LIBRARY',99/365,SD
// EXTENT SYS010,vvvvvv,1,0,ssss,nnnn
// ASSGN SYS010,DISK,VOL=vvvvvv,SHR
// EXEC LIBR
  DEF LIB=SAGLIB   R=Y
  DEF S=SAGLIB.WCPvrs R=Y
  DEF S=SAGLIB.WCPvrsZ R=Y
  DEF S=SAGLIB.WTCvrs R=Y
  DEF S=SAGLIB.WALvrs R=Y
/*
/&
* $$ EOJ
```

Note: If you use System Maintenance Aid (SMA) for the installation, you can change the library name to SAG.PRODUCT.SMALIB to contain all of Software AG's software by product.

VSAM Library

The following sample JCS can be used to define the Entire Net-Work library as a VSAM library:

```
* $$ JOB JNM=LIBRDEF,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB LIBRDEF
// EXEC IDCAMS,SIZE=AUTO
   DEFINE CLUSTER                          -
   (NAME(NETWORK.vrs.LIBRARY)            -
   VOLUME(vvvvvv)                         -
   NONINDEXED                             -
   RECORDFORMAT(NOCIFORMAT)               -
   SHR(3)                                 -
   CYLINDERS (nnnn 1)                     -
   DATA (NAME(NETWORK.vrs.LIBRARY.DATA))
/*
// DLBL SAGLIB,'NETWRK.vrs.LIBRARY',,VSAM,CAT=ccccccc
// EXEC LIBR
   DEF LIB=SAGLIB   R=Y
   DEF S=SAGLIB.WCPvrs R=Y
   DEF SAGLIB.WCPvrsZ R=Y
   DEF S=SAGLIB.WTCvrs R=Y
   DEF S=SAGLIB.WALvrs R=Y
/*
```

```
/&
* $$ EOJ
```

Installation Tape Restore

The following sample JCS can be used to restore the Entire Net-Work installation tape:

```
* $$ JOB JNM=RESNET,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB RESNET
// ASSGN SYS005,IGN
// ASSGN SYS006,cuu
// MTC REW,SYS006
// PAUSE MOUNT THE NETWORK 5 INSTALLATION TAPE
// MTC FSF,SYS006,tt
// ASSGN SYS001,DISK,VOL=vvvvvv,SHR
// DLBL SAGLIB,`NETWRK.vrs.LIBRARY',99/365,SD
// EXTENT SYS001,vvvvvv,1,0,ssss,nnnn
// EXEC LIBR
  RESTORE SAGLIB.WCPvrs TAPE=SYS006 REPLACE=YES
  RESTORE SAGLIB.WALvrs TAPE=SYS006 REPLACE=YES
/*
// MTC REW,SYS006
// ASSGN SYS006,UA
/*
/&
* $$ EOJ
```

Note: If a VSAM library is being used, the ASSGN and the EXTENT for SYS001 are not needed; the DLBL for SAGLIB might look like the following:

```
// DLBL SAGLIB,'NETWRK.vrs.LIBRARY',,VSAM,CAT=ccccccc
```

Note: If installing the z/VSE TCP/IP, include an additional step with the following:

```
// MTC REW,SYS006
// MTC FSF,SYS006,UU
   ...
   ...
   ...
// EXEC LIBR
  RESTORE SAGLIB.WTCvrs TAPE=SYS006 REPLACE=YES
```

Catalog Procedure Definition

The following sample JCS can be used to define the Entire Net-Work catalog procedure:

```
* $$  JOB JNM=NETPROCS,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB NETPROCS
// OPTION LOG
// EXEC LIBR
 ACCESS SUB=IJSYSRS.SYSLIB
 CATALOG NETnnLIB.PROC  REPLACE=YES
*
*    ENTIRE NETWORK 5 LIBRARY DEFINITIONS
*
***  EITHER FOR NON VSAM OWNED LIBRARY
// DLBL SAGLIB,`NETWRK.vrs.LIBRARY',99/365,SD
// EXTENT SYS010,vvvvvv
// ASSGN SYS010,DISK,VOL=vvvvvv,SHR
***  OR FOR VSAM OWNED LIBRARY
// DLBL SAGLIB,`NETWRK.vrs.LIBRARY',,VSAM,CAT=ccccccc
*
*    THE LIBDEFS FOR BOTH VSAM AND NON VSAM LIBRARIES
*
// LIBDEF PHASE,SEARCH=(SAGLIB.WCPvrsZ,SAGLIB.WCPvrs,SAGLIB.WTCvrs,SAGLIB.WALvrs),TEMP
// LIBDEF PHASE,CATALOG=SAGLIB.WCPvrsZ,TEMP
// LIBDEF OBJ,SEARCH=(SAGLIB.WCPvrsZ,SAGLIB.WCPvrs,SAGLIB.WTCvrs,SAGLIB.WALvrs),TEMP
// LIBDEF SOURCE,SEARCH=(SAGLIB.WCPvrs,SAGLIB.WTCvrs,SAGLIB.WALvrs),TEMP
/+
/*
/&
* $$ EOJ
```

Also see the sample member NET*nn*LIB.X in the Entire Net-Work source library.

Adabas SVC

Sample job ADASIP can be used to install the Adabas SVC. See Step 3 of the ***Installation Procedure***.

Note: Note: This job must run in the BG partition.

```
* $$ JOB JNM=ADASIP,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ADASIP
// OPTION LOG
// EXEC PROC=NETnnLIB
SET SDL ADASVC81,SVA
/*
// UPSI 00000000
// OPTION SYSPARM='xxx'
// EXEC ADASIP
```

```
/*
/&
* $$ EOJ
```

Unload ZAPS Data Sets

Refer to the *Report of Tape Creation* to determine whether any *pppvrs*.ZAPS data sets have been distributed on the installation tape. If so, the following sample JCS can be used to unload the ZAPS data sets.

```
* $$ JOB JNM=LIBRIPT,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB LIBRIPT
// ASSGN SYS006,cuu
// MTC REW,SYS006
// MTC FSF,SYS006,tt
// ASSGN SYSIPT,SYS006
// DLBL SAGLIB,`NETWRK.vrs.LIBRARY'
// EXTENT ,vvvvvv
// EXEC LIBR,PARM=`ACCESS S=SAGLIB.pppvrs'
/*
// ASSGN SYSIPT,SYSRDR
// DLBL SAGLIB,`NETWRK.vrs.LIBRARY'
// EXTENT ,vvvvvv
// EXEC LIBR
 ACCESS S=SAGLIB.pppvrs
 LIST $READVSE.A
/*
/&
* $$ EOJ
```

This job LIBR catalogs the distributed maintenance for Entire Net-Work into the pppvrs sublibrary and prints the $READVSE.A member, which contains the necessary documentation for applying distributed maintenance.

The WALvrs Sublibrary

The Adabas limited sublibrary WAL*vrs* is provided on the installation tape for systems where Adabas is not installed. WAL*vrs* contains the Adabas code required by Entire Net-Work.

To apply maintenance to the WAL components, refer to the instructions accompanying the fix. If additional information or explanation is needed, contact your Software AG technical support representative.

The following sections describe the functionality of certain members in the WAL*vrs* library:

WTOVSE

The WTOVSE phase is loaded by ADAUSER when an error is encountered. It issues an appropriate error message and abends with a dump, unless an optional zap is applied to eliminate the dump. WTOVSE is required to accommodate the IBM z/VSE MACRO execution-mode restrictions for z/VSE 1.3 and above.

SAGUSER Job Exit Facility

The phases SAGJBXT, SAGIPT, and SAGINST provide a job exit facility that can be used to scan a job stream for input-override statements; these statements indicate that card input for a job step is to come from library members rather than from SYSRDR or SYSIPT. The exit has a default capacity to store a maximum of 2000 input-override cards simultaneously throughout the system. Adabas and Entire Net-Work can use this facility in the processing of CARD and KARTE parameters.

You can use this feature by adding SAGUSER control statements to the job stream for running the Entire Net-Work startup job JCLNET. The SAGUSER control statements replace the CARD and KARTE input statements. See the sample job in the section ***Running the Startup Job with the SAGUSER Job Exit*** .

For example, the SAGUSER control statement

```
* SAGUSER FILE=CARD,MEMBER=ADARUN
```

searches the current LIBDEF chain for the member ADARUN. If found, it is used for the ADARUN control statements in place of SYSIPT.

To permit flexible startup processing, multiple SAGUSER statements may be specified for each file. The library members are assumed to be type "A".

Initialization

The job exit can be installed during ASI processing or at any time afterward. Two programs (SAGJBXT and SAGIPT) must be installed in the SVA, and the program SAGINST must be run to initiate job exit processing.

In z/VSE 1.3 and above, you can include SAGJBXT in the $JOBEXIT list of eligible exits, but you must still place SAGIPT in the SVA and run SAGINST to allocate the SAGUSER table.

SAGIPT runs above the line if an appropriate 31-bit PSIZE is available. In addition, the table that stores information from SAGUSER statements is placed in 31-bit GETVIS, if available.

The following sample JCS initializes the job exit:

```
* $$ JOB JNM=SAGEXIT,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB SAGEXIT
// EXEC PROC=NETnnLIB
 SET SDL
 SAGJBXT,SVA
 SAGIPT,SVA
/*
// EXEC SAGINST
/*
/&
* $$ EOJ
```

Storage Requirements

The job exit requires a total of 84K of SVA storage. Of that total, 2K is used for program storage (PSIZE) and 82K for GETVIS storage. Under z/VSE 1.3 or above running in ESA, all of the GETVIS and 1K of the PSIZE can be run above the 16-MB line.

Activating Entire Net-Work Use of Job Exit Processing

To allow Entire Net-Work to use SAGUSER statements in the job stream, recatalog the Adabas option table (ADAOPTD) specifying JOBEXIT=YES. See the sample member ADAIOOAL.X in the WAL sublibrary.

Diagnostic Functions

After the job exit is installed, executing SAGINST with UPSI 1 produces a dump of the DD name table for diagnostic purposes.

If the size of the DD name table needs to be changed for any reason, SAGIPT may be zapped before being loaded into the SDL. At offset X`0C', the initial value is X`00014874' or a total of 84,084 bytes. This value may be zapped when some more appropriate value is required.

Each table element is 42 bytes long. The default values assume 10 SAGUSER statements per file name, 10 file names, and 20 partitions, plus 2 entries for slack. This number is an estimate of maximum concurrent residency; each statement is removed from the table when it is used.

Partition/Address Space Requirements

Entire Net-Work typically requires at least 512KB-1024KB to operate. A larger partition size may be required, depending on the number of line drivers and the buffer space defined on the NODE statement.

> **Note:** Enough real storage must be allocated to the partition to accommodate the page-fixed buffers specified by the NODE statement (if any). If necessary, change the ALLOCR statement for the Entire Net-Work partition in the IPL procedure.

The real storage should be specified by the following job control statement:

```
// SETPFIX LIMIT=nnnk
```

Running the Startup Job

The following sample job can be used for running Entire Net-Work. See the sample source member JCLNET in the source library for an alternate example.

```
* $$ JOB JNM=NETWKTST,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB NETWKTST
// ASSGN SYS009,PRINTER
// EXEC PROC=NETnnLIB
// DLBL NETPRNT,'RD.NETPRNT.DATA'                    <==== See note below
// EXTENT SYS055,VSE20B,1,0,16065,75
// ASSGN SYS055,DISK,VOL=VSE20B,SHR
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=NETWRK,FORCE=NO
ADARUN TARGETID=4003,SVC=31
ADARUN NAB=10,LU=65535,NC=10
/*
*
* THE FOLLOWING STATEMENT DEFINES THE LOCAL NODE.
*
NODE RNODE BUFFERS=(32K,32K,512K,64K)
*
* NOW DEFINE THE DRIVERS AND LINKS.
*
DRIVER IUCV ACCEPTUI=Y,SVC=45
*
* DEFINE A LINK TO THE ID TABLE MANAGER FOR ALL CMS DATABASES.
*
LINK TOITM IUCV US=DBIDSERV,MSGLIM=48,STATB=N,TYPE=2WAY, -
               STATC=N
*
* DEFINE AN IUCV LINK TO THE 'PROD' MACHINE.
*
LINK SNODE IUCV US=PRODMACH,MSGLIM=48,STATB=N, -
          TYPE=2WAY,STATC=N
/*
/&
* $$ EOJ
```

Note: 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. 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.

Running the Startup Job with the SAGUSER Job Exit

The following sample job illustrates the use of the SAGUSER job exit (see the section ***SAGUSER Job Exit Facility***).

The SAGUSER control statement is added to the Entire Net-Work startup job in two places. It replaces the ADARUN statements and the NODE, DRIVER, and LINK statements.

```
* $$ JOB JNM=NETWKTST,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB NETWKTST
// ASSGN SYS009,PRINTER
// EXEC PROC=NETnnLIB
*
* THE FOLLOWING STATEMENT CAUSES THE SAGUSER JOB
* EXIT TO READ THE ADARUN PARAMETERS IN THE CARD
* FILE OF THE ADARUN MEMBER
*
* SAGUSER FILE=CARD,MEMBER=ADARUN
*
/*
* THE FOLLOWING STATEMENT CAUSES THE SAGUSER JOB
* EXIT TO READ THE NODE, DRIVER, AND LINK STATEMENT
* PARAMETERS IN THE KARTE FILE OF THE NETWRK MEMBER
*
* SAGUSER FILE=KARTE,MEMBER=NETWRK
*
// EXEC ADARUN,SIZE=ADARUN
/*
/&
* $$ EOJ
```

Entering Operator Commands

You can communicate with Entire Net-Work while it is running by using the operator command interface. Operator commands are essentially independent of the operating system. See the section *Entire Net-Work Operator Commands* in the *Entire Net-Work Reference Guide*. For information about the operator commands specific to the line drivers, see the specific documentation concerning those line drivers.

Operator commands are submitted to Entire Net-Work in the same way that operator commands are submitted to Adabas. Enter operator commands as follows:

1. Enter an MSG command for the z/VSE partition in which Entire Net-Work is executing; for example:

   ```
   MSG Fn
   ```

   When ready for communication, Entire Net-Work responds with the message NETI2If the command ADAEND, END, HALT, NETEND, or TERMINATE causes Entire Net-Work to end the session, no outstanding reply is presented so that an orderly shutdown can occur without the need for operator intervention.

2. Enter the desired operator commands, with only one space between the command and the argument. For example:

```
CONNECT V2ZHOST
```

Multiple spaces result in error message "NET0010E: REQUIRED PARAMETER NOT SPECIFIED".

You can also enter a command followed directly by a slash (/) to end any further operator communications until the next MSG command.

3. When all desired commands have been entered, close the operator communication session by entering a null command (EOB). To enter any more commands after the session is closed, begin with Step 1 above.

   Example:

```
msg F5
5 NETI24 04003 READY FOR OPERATOR COMMUNICATION
5 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
5
```

Applying Corrective Fixes

All corrective fixes applied to Entire Net-Work components are distributed in MSHP format. Once a problem is reported, Software AG provides an MSHP fix.

To apply and implement fixes for Entire Net-Work components:

- Step 1: Prepare Entire Net-Work for Maintenance
- Step 2: Apply MSHP Fixes to Entire Net-Work Components

The sample jobs provided in Steps 1 and 2 contain the following substitution variables. To use the sample jobs for the TCP/IP line driver, use WTC in place of WCP.

mshpnumber	The number of the MSHP fix.
module	Entire Net-Work object member to be fixed and then relinked.
number	The ESD number of the CSECT to be zapped.
offset	The hexadecimal offset to the beginning of the fix.
rrrr	The replacement data for the fix.
vrs	The Entire Net-Work version, revision, and system maintenance (SM)
vvvv	The verify data for the fix.

Step 1: Prepare Entire Net-Work for Maintenance

A history file is required to apply an MSHP fix. You can use an existing file, such as the system history file or an existing Software AG generic history file, or create a new history file for this product installation.

1. Use the following sample job to create an MSHP history file:

```
* $$ JOB JNM=MSHPARCH,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB MSHPARCH CREATE A HISTORY FILE
// DLBL IJSYS02,`NETWRK.vrs.HISTORY.FILE',99/365
// EXTENT SYS002,vvvvvv,1,0,ssss,nnnn
// ASSGN SYS002,DISK,VOL=vvvvvv,SHR
// ASSGN SYSLST,PRINTER
// EXEC MSHP
  CREATE HISTORY AUXILIARY
/*
/&
* $$ EOJ
```

2. Use the following sample job to define Entire Net-Work to MSHP as a product/component using the MSHP ARCHIVE process:

```
* $$ JOB JNM=MSHPARCH,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB  MSHPARCH DEFINE Entire Net-Work vrs TO MSHP
// EXEC PROC=NETnnLIB
// DLBL IJSYSHF,'NETWRK.vrs.HISTORY.FILE',99/365
// EXTENT SYS002,vvvvvv,1,0,ssss,nnnn
// ASSGN SYS002,DISK,VOL=vvvvvv,SHR
// ASSGN  SYSLST,PRINTER
// EXEC  MSHP
 ARCHIVE WCPvrs
 COMPRISES 9001-WCP-00
 RESOLVES 'Entire Net-Work vrs - Software AG'
 ARCHIVE 9001-WCP-00-vrs
 RESIDENCE PRODUCT=WCPvrs
           PRODUCTION=SAGLIB.WCPvrsZ
           GENERATION=SAGLIB.WCPvrsZ
/*
/&
* $$ EOJ
```

The X. member MSHPARCH is provided in the Entire Net-Work library as an additional example to create an alternate history file and define WCP and WAL as products and components to MSHP.

Step 2: Apply MSHP Fixes to Entire Net-Work Components

1. Use the following sample job to apply MSHP fixes to the WCPvrs sublibrary within a test environment:

```
* $$ JOB JNM=WCPMSHP,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB ZAP APPLY MSHP TO Entire Net-Work OBJECT
// OPTION LOG
// EXEC PROC=NETnnLIB
// DLBL IJSYSHF,'NETWRK.vrs.HISTORY.FILE',99/365
// EXTENT SYS002,vvvvvv,1,0,ssss,nnnn
// ASSGN SYS002,DISK,VOL=vvvvvv,SHR
// EXEC MSHP
  CORRECT 9001-WCP-00-vrs:fixname
  AFFECTS MODULE=module ESDID=number
  ALTER offset vvvv:rrrr
  INVOLVES LINK=linkname
  RESOLVES 'fix description'
/*
/&
* $$ EOJ
```

2. The MSHP fix relinks the affected phases using one or more of the jobs found in the Entire Net-Work WCPvrs sublibrary, member type X (LINKxxxx). Each fix will indicate the job used to relink the phase.
3. If a fix needs to be removed, use the following sample job, which uses the MSHP UNDO facility, to remove the MSHP fix from the WCP`vrs` component.

```
* $$ JOB JNM=WCPUNDO,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB WCPUNDO
// OPTION LOG
// EXEC PROC=NETnnLIB
// DLBL IJSYSHF,'NETWRK.vrs.HISTORY.FILE',99/365
// EXTENT SYS002,vvvvvv,1,0,ssss,nnnn
// ASSGN SYS002,DISK,VOL=vvvvvv,SHR
// ASSIGN SYSLST,PRINTER
// EXEC MSHP
   UNDO  9001-WCP-00-vrs:fixname
/*
/&
* $$ EOJ
```

Assembling and Linking Optional User Exits

The source member ASMNET.X contains sample JCS to assemble and link optional Entire Net-Work user exits.

6 BS2000/OSD Environments

This section provides information about installing and running Entire Net-Work in the BS2000/OSD operating system environment.

The information in this section uses the substitution variable `vrs`, which stands for the current version, revision, and system maintenance (SM) level of the product; for example: "WCP621".

Refer to the Entire Net-Work *Report of Tape Creation* for the names and sequence of the libraries and files on the installation tape.

Installation Tape Information

Tape Contents

The Entire Net-Work installation tape for BS2000/OSD contains all components required to run Entire Net-Work as sequential files:

- Entire Net-Work Mainline Processor and Service Routines
- Entire Net-Work TCP/IP Line Driver (if ordered)
- Adabas components required to run Entire Net-Work if Adabas is not installed
- Installation-dependent files

Disk Contents

After successful installation, the disk will contain the following LMS libraries:

Library Name	Description
WCPvrs.LIB	Entire Net-Work Main Line and Service Routines
WTCvrs.LIB	Entire Net-Work TCP/IP Line Driver (optional)

All load modules, sources, and jobs are contained in the appropriate libraries.

Prerequisites

Verify that any required updates are applied to the WAL 8 libraries so it runs Entire Net-Work Administration in BS2000 environments.

Non-SMA Installation Procedure

If you are not using Software AG's System Maintenance Aid (SMA), follow the instructions in this section.

- Step 1: Copy the Contents of the Tape to Disk
- Step 2: Apply Corrective Maintenance
- Step 3: Perform the Line Driver-Dependent Installations

Step 1: Copy the Contents of the Tape to Disk

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 files from tape to disk. You will then need to perform the individual installation procedure for each component to be installed.

- Step 1: Copy the Library SRVvrs.LIB from Tape to Disk
- Step 2: Copy the Procedure COPY.PROC from Tape to Disk
- Step 3: Copy all Product Files from Tape to Disk

Step 1: Copy the Library SRVvrs.LIB from Tape to Disk

This step is not necessary if you have already copied the library `SRVvrs.LIB` from another Software AG installation tape. For further information, refer to the element `#READ-ME` in this library. The library `SRVvrs.LIB` is stored on the tape as a sequential file named `SRVvrs.LIBS` containing LMS commands. The current version `vrs` can be obtained from the *Report of Tape Creation*. To convert this sequential file into an LMS library, execute the following commands:

```
/IMPORT-FILE  SUPPORT=*TAPE(FILE-NAME=SRVvrs.LIBS,-
/   VOLUME=volser, DEV-TYPE=tape-device)
/ADD-FILE-LINK  LINK-NAME=EDTSAM, FILE-NAME=SRVvrs.LIBS,-
/   SUPPORT=*TAPE(FILE-SEQ=3), ACC-METH=*BY-CAT,-
/   BUF-LEN=*BY-CAT, REC-FORM=*BY-CAT, REC-SIZE=*BY-CAT
/START-EDT
@READ  '/'
@SYSTEM 'REMOVE-FILE-LINK  EDTSAM'
@SYSTEM 'EXPORT-FILE  FILE-NAME=SRVvrs.LIBS'
@WRITE  'SRVvrs.LIBS'
@HALT
/ASS-SYSDTA  SRVvrs.LIBS
/MOD-JOB-SW  ON=1
/START-PROG  $LMS
/MOD-JOB-SW  OFF=1
/ASS-SYSDTA  *PRIMARY
```

where:

tape-device is the device type of the tape, for example: `TAPE-C4`
volser is the VOLSER of the tape (see *Report of Tape Creation*)

Step 2: Copy the Procedure COPY.PROC from Tape to Disk

To copy the procedure `COPY.PROC` to disk, call the procedure `P.COPYTAPE` in the library `SRV`*`vrs`*`.LIB`:

```
/CALL-PROCEDURE  (SRVvrs.LIB,P.COPYTAPE), -
/   (VSNT=volser, DEVT=tape-device)
```

If you use a TAPE-C4 device, you may omit the parameter `DEVT`.

Step 3: Copy all Product Files from Tape to Disk

To copy all Software AG product files from tape to disk, enter the procedure `COPY.PROC`:

```
/ENTER-PROCEDURE  COPY.PROC, DEVT=tape-device
```

If you use a TAPE-C4 device, you may omit the parameter `DEVT`. The result of this procedure is written to the file `L.REPORT.SRV`.

Step 2: Apply Corrective Maintenance

Apply corrective maintenance if necessary (refer to the *Report of Tape Creation*).

Caution: 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.

Step 3: Perform the Line Driver-Dependent Installations

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

SMA Installation Instructions

If you are using Software AG's System Maintenance Aid (SMA) install, proceed as usual. After installation, the PLAM libraries WCP*vrs*.LIB and optionally WTC*vrs*.LIB will exist on your disk and be ready for use.

For more information about the SMA installation process, refer to the *System Maintenance Aid Manual*.

Running in BS2000/OSD Environments

The following points should be considered when running Entire Net-Work in the BS2000/OSD environment:

- Correctly configured TCP/IP environments are prerequisites for starting Entire Net-Work.
- The library WCP*vrs*.LIB contains a set of predefined elements that is usually ready to use. The J-Element START can be used to start Entire Net-Work with a minimum set of configuration data. It contains an example configuration including a DCAM and a TCP/IP connection.
- If the new tracing facilities are required, set the TRACE parameter to YES in the call to the STARTP procedure.
- In order to define which Adabas modules will be used to run Entire Net-Work, either:
  1. Create a JOB-VARIABLE named 'ADALIB' and set it to the correct name of the Adabas or WAL library, or
  2. Modify the START procedure so that the name is supplied as the ADALIB parameter in the call to STARTP, as follows:

     ```
     /CALL-PROC -
     /   FROM-FILE = *LIB(LIB = WCPvrs.LIB, -
     /                    ELE = STARTP), -
     /   PROC-PAR  = (DUMP  = YES, -
     /                ADALIB= xxxxxx, -
     /                TRACE= NO, -
     /                OPMSG = YES, -
     /                TIME  = NO), -
     /   LOGGING   = *YES
     ```

     where *xxxxxx* is the name of your library. For example: $SAG.WAL814.MOD
- If the Simple Connection Line Driver is to be used, then an extra task will be started automatically. The parameters for starting this task are defined in the variable NWBS2SUB within the STARTP

procedure. The library or procedure names may be altered if required (see the section *Starting the TCP/IP Line Driver* in the *Entire Net-Work Administration Guide*).

Starting Entire Net-Work

To actually start Entire Net-Work, enter the following command:

```
/ENTER-JOB (WCPvrs.LIB,START)
```

The SAG supplied J-Element START is executed. It has the following sample content:

```
/SET-LOGON-PARAMETERS -
/   USER-ID   = *NO,-
/   ACCOUNT   = *NONE,-
/   JOB-CLASS = *STD,-
/   JOB-NAME  = NETWORK
/ASSIGN-SYSOUT -
/   TO   = START.JNL
/CALL-PROC -
/   FROM-FILE = *LIB(LIB = WCPvrs.LIB,-
/                    ELE = STARTP),-
/   PROC-PAR  = (DUMP  = YES,-
/                ADALIB= &(ADALIB),-
/                LICLIB= LICvrs.MOD,-
/                WCPLIB = WCPvrs.LIB,-
/                WTCLIB = WTCvrs.LIB,-
/                MLCLIB = MLCmvrs.LIB,-
/                VERSION = vrs,-
/                TRACE = NO,-
/                OPMSG = YES,-
/                TIME  = NO),-
/   LOGGING   = *YES
ADARUN PROG=NETWRK,TARGETID=vrs2,IDTNAME=ADANET02
ADARUN TASKCTGY=TP,FORCE=YES,GROUPS=NO,CMDQMODE=ABOVE
* The next 2 lines are only needed when running with a
* Sockets version less than 2.2
ADARUN CMSIZE=32000000,CMFIX=Y,CMSCOPE=GROUP,CMADDR=X'01300000'
ADARUN CMLSIZE=4096000,CMLFIX=Y,CMLSCOPE=GROUP
/EOF
NODE    MHK2 NT=128,LOG=N,PASSWORD=ALL,-
       TRANS DEF ID=(0,0),TOASCII=E2ASIE,TOEBCDIC=A2ESIE
DRIVER  DCAM
DRIVER  SMH  API=BS2,SERVERID=8010
DRIVER  TCPI API=BS2,ACCEPTUI=N,SERVERID=8882,APITRACE=N,-
            OPTIONS1=(0,0,0,0,0,0,1024)
LINK MHK1 DCAM PRONAME=HOST
LINK SMHLINK  SMH
LINK IBM TCPI I=157.189.160.95,SERVERID=2000
LINK H60 TCPI ADJHOST=SIH1,SERVERID=8881
LINK PCMHK TCPI ADJHOST=PCMHK,SERVERID=8882
/EXIT-JOB -
/   MODE    = NORMAL,-
/   SYS-OUT = NONE
```

Checking the Sockets Version

To determine the version of Sockets that is running, input the following command:

```
SHOW-SUBSYSTEM-STATUS SUBSYTEM-NAME=SOCKETS
```

The Sockets version number will be returned.

JCL Notes for Running Sockets

The default CMSIZE is around 32MB. We recommend that this size be set to the largest possible in the address space. This area is used by all drivers by the buffer pool manager. It is important to set this as large as possible, particularly when transferring large data objects. This system runs a subtask called NETTCP.

If optional zap WY622002 is applied, the ADARUN parameters CMSIZE, CMFIX, CMSCOPE, CMADDR, CMLSIZE, CMLFIX and CMLSCOPE need not be applied. Here, the ability to pass large objects through Entire Net-Work is dependent on the size of the address space alone. This system does not run a subtask NETTCP.

⚠ **Important:** Do not change the parameters CMFIX=Y, CMSCOPE=GROUP, or CMADDR=X'01300000'

The extra ADARUN entries, beginning with CMSIZE and CMLSIZE, are required for the Simple Connection Line Driver.

If you wish to use a different job class for Entire Net-Work, it may be supplied to the START procedure as follows (xxxxxxxxx is your defined JOB-CLASS):

```

/SET-LOGON-PARAMETERS -
/   USER-ID   = *NO,-
/   ACCOUNT   = *NONE,-
/   JOB-CLASS = xxxxxxxx,-
/   JOB-NAME  = NETWORK
```

START calls procedure STARTP, which is shown below:

```
  /"* **************************************************************** *"
/"*                                                                  *"
/"*               Entire Net-Work STARTP Procedure                   *"
/"*                                                                  *"
/"*  MODE SINGLE: ONLY ONE NET-WORK SESSION PER USER-ID ALLOWED      *"
/"*                                                                  *"
/"* **************************************************************** *"
/SET-PROCEDURE-OPTIONS -
/    IMPLICIT-DECLARATION = *YES,-        &* For SDF-BASYS
/    LOGGING-ALLOWED      = *YES,-        &*
/    SYSTEM-FIL-CONTEXT   = *SAME-AS-CAL,-&* For ADARUN
/    JV-REPLACEMENT       = *AFTER,-      &* Allow JV for ADALIB
/    ERROR-MECHANISM      = *SPIN-OFF     &* For SDF-BASYS
/BEGIN-PARAMETER-DECLARATION
/  DECLARE-PARAM -                        &* Used to form WCP and WTC
/       VERSION(INIT='621')               &* library names.
/  DECLARE-PARAM -                        &* Adabas or WAL library.
/       ADALIB                            &*
/  DECLARE-PARAM -                        &* License Module
/       LICLIB                            &*  Library
/  DECLARE-PARAM -                        &*
/       WCPLIB                            &* Net-Work main library
/  DECLARE-PARAM -                        &*
/       WTCLIB                            &* Net-Work TCP/IP Library
/  DECLARE-PARAM -                        &*
/       MLCLIB                            &* License Library
/  DECLARE-PARAM -                        &* Determines whether a dump
/       DUMP(INIT='NO')                   &* is created for errors.
/  DECLARE-PARAM -                        &* For tracing facilities
/       TRACE-TO-NETPRNT(INIT='NO')       &*
/  DECLARE-PARAM -                        &* TCP/IP operator messages.
/       OPMSG(INIT='YES')                 &*
/  DECLARE-PARAM -                        &* CPU limit for TCP sub task;
/       TIME(INIT='NO')                   &* NO, STD or 1 - 32767
/END-PARAMETER-DECLARATION
/
/  IF (OPMSG = 'NO')
/     MOD-USER-SW ON = 1
/  END-IF
/
/  MODIFY-JOB-OPTIONS -                   &* Log messages to the system
/         OPERATOR = *YES                 &* Console.
/
/  MODIFY-TEST-OPTIONS -
/         DUMP     = &DUMP
/
/  CALL-PROC -
/         FROM-FILE = *LIB(LIB = &(WCPLIB), -
/                          ELE = NEWGEN), -
/         PROC-PAR  = (NET.PROTO.OUT)     &* Create the next generation
/                                         &* in this file group, then
/  ASSIGN-SYSOUT -                        &* direct SYSOUT to it.
/         TO        = NET.PROTO.OUT(+0)
/
/ IF (TRACE-TO-NETPRNT = 'YES')
/   CALL-PROC
/        FROM-FILE = *LIB(LIB = &(WCPLIB), -
/                     ELE = NEWGEN), -
/        PROC-PAR  = (NET.NETPRNT)          &* Create the next generation
/                                           &* in this file group, then
/   SET-FILE-LINK -                         &* direct NETPRNT tracing
/        LINK-NAME = NETPRNT,-              &* to it.
/        FILE-NAME = NET.NETPRNT(+0), -
/        ACC-METH  = SAM,-
/        REC-FORM  = *FIXED(REC-SIZ=121)
/
/END-IF
/  SHOW-JOB-STATUS
/
/" *-----------------------------------------------------------------*"
/"         The following section is for the TCP/IP driver            "
/"                                                                    "
/" 1) The task is started according to the parameters supplied in the "
/"    variable NWBS2SUB.                                              "
/" 2) If a different procedure is to be used, the LIB and ELE names   "
/"    in the FROM parameter may be altered.                           "
/" 3) If the subtask code is to be loaded from a different library,   "
/"    then the WTCLIB name in the PROC-PAR parameter may be changed.  "
/"                                                                    "
/"         ****  Do NOT start the subtask manually! ****              "
/"*------------------------------------------------------------------*"
/  SHOW-FILE-ATTRIBUTES -               &* Test whether TCP/IP driver
/         FILE-NAME = &(WTCLIB),-       &* is available. If it is...
/         INFORMAT  = *MIN
/
/  SET-FILE-LINK -                      &* add WTC file link for BLS,
/         LINK-NAME = BLSLIB01, -
/         FILE-NAME = &(WTCLIB)
/
/  DECLARE-VARIABLE -                   &* then set up the parameters
/         NWBS2SUB(INIT=-               &* for the NWBS2SUB task.
              /'ENTER-PROC FROM    = (LIB = WCP&VERSION..LIB,-
              /                       ELE = SUBTASKPROC),-
              /            PROC-PAR= (WTCLIB= &(WCPLIB),-
              /                       WCPLIB= &(WTCLIB),-
              /                       DUMP  = &DUMP),-
              /            JOB-CLAS= *STD,-
              /            JOB-NAME= NETTCP,-
              /            RESOURCE= *PAR(CPU-LIMIT=&TIME),-
              /            SYS-OUTP=
*DELETE')

/SET-JOB-STEP                           &* Catch spinoff
/
```

```
/  SET-FILE-LINK -                      &* add WCP file link for BLS.
/         LINK-NAME = BLSLIB00, -
/         FILE-NAME = &(WCPLIB)
/
/  SET-FILE-LINK -                      &* add a file link for ADARUN.
/         LINK-NAME = DDLIB, -
/         FILE-NAME = &ADALIB
/  SET-FILE-LINK -                        &* Add a file link for BLS.
/         LINK-NAME = BLSLIB02, -
/         FILE-NAME = &(ADALIB)
/
/  SET-FILE-LINK -                        &* Add a file link for BLS.
/         LINK-NAME = BLSLIB03, -
/         FILE-NAME = &(LICLIB)
/
/  SET-FILE-LINK -                        &* Add a file link for BLS.
/         LINK-NAME = BLSLIB04, -
/         FILE-NAME = &(MLCLIB)
/
/
/  SEND-MSG -
/       TO = *OPERATOR,-
/       MSG = 'STARTING Entire Net-Work (V &VERSION)'
/
/  START-PROGRAM -
/         FROM      = *MODULE(LIB = &ADALIB, ELEM = ADARUN, PROG = ANY,-
/                             RUN-MOD = ADV(ALT-LIB = YES))
/
/  IF (OPERATOR-MSG = 'NO')             &* Unset switch 1 if
/       MOD-USER-SW OFF = 1             &* it was set earlier.
/  END-IF                               &*
/
/EXIT-PROC
```

Notes:

1. Modify OPMSG=NO if possible. This disables all dependent Simple Connection Line Driver operator messages.
2. Modify DUMP=YES if possible. This allows your software AG representative to assist you more quickly, should this be necessary.
3. As Entire Net-Work starts, it will display a number of messages on the system console. When it has initialized successfully, it will display the message: NET1000I: N E T - W O R K node MHK2 active
4. This messages is preceded by several other messages that are also displayed on the system console.

Preparing and Installing the Product License File

You must install a valid license on all mainframe platforms in which your Software AG mainframe product is installed. The license file is provided as an XML document. The document's 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, contact your Software AG technical support representative for assistance.

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 using either method.

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

1. Transfer the license to BS2000, as described in ***Transferring a License File from PC to a BS2000 Host Using FTP***, elsewhere in this guide.
2. Verify that the transferred license file is a SAM sequential file, taking care to preserve its format as ASCII.

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

- Verify that the transferred license file is a SAM sequential file, taking care to preserve its format as ASCII.

Check the license file by modifying and running the job WCP*vrs*.LIB(LICCHECK,J).

Installing the Product License File

To install the product license file, complete the following steps:

1 Convert the license to a loadable element (NETLIC) that is then loaded by Entire Net-Work by modifying and running the job WCP*vrs*.LIB(NETLICA,J).

This will put the NETLIC into a license module library.

2 Update your Entire Net-Work startup jobs to reference the license library. Complete the following steps:

1. Make sure any previously-created NETLIC 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, then reads from a DDLIC link name referenced in the Entire Net-Work startup job.
2. Software AG licensing requires that the modules LICMAIN(R) and LICUTIL(R) be loaded when Entire Net-Work starts up. These modules are distributed in the MLC*vrs*.MOD library. Do either (but not both) of the following:
   - Copy LICMAIN and LICUTIL into the WCP*vrs*.LIB library.
   - Concatenate MLC*vrs*.MOD into the BLSLIB chain with WCP*vrs*.LIB.
3. Verify that the license file is correctly referenced in each startup job. Do either (but not both) of the following:
   - Verify that the NETLIC load module 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 BLSLIB concatenation of the Entire Net-Work startup job or copy the NETLIC library into the WCP*vrs*.LIB library.
   - Verify that there is *no* NETLIC load module accessible to the Entire Net-Work load library and that the following link statement is included in the WCP*vrs*.LIB job STARTP (J):

```
/ADD-FILE-LINK DDLIC,license_file
```

where *license_file* is the name of the license file loaded from the tape (in ASCII format). Entire Net-Work first tries to load NETLIC and, if unsuccessful, it reads from the DDLIC link name.

Note: To read the license file via the DDLIC link name, the correction WCP621L001 (library WCP621.BS2L001) must be applied.

Output File Concept

Entire Net-Work produces 2 FGGs with 7 generations, which are:

FGG	Description
NET.PROTO.OUT	contains all main task output*
NET.PROTO.TCP.OUT (Sockets 2.0 only)	contains all subtask output (TCP/IP)*
NET.NETPRNT	(optional) contains tracing data

*All console messages are also contained in the above output files.

Entering Operator Commands

Entire Net-Work operator commands (as described in their corresponding sections) are generally issued as follows:

```
/INF-PROG 'message text',*TSN(<tsn>)
```

```
/INTR <tsn>,<command>
```

<tsn> is the task sequence number of Entire Net-Work's main task or subtask

<command> is the command of Entire Net-Work's main task or subtask

The following examples assume the task serial number of the Entire Net-Work task is 1234:

```
/INF-PROG D 'LINKS',*TSN(1234)
```

displays all links currently known by Entire Net-Work.

```
/INF-PROG C 'TOUNIX1',*TSN(1234)
```

connects to foreign node known by link named "tounix1".

```
/INF-PROG C 'MYPC',*TSN(1234)
```

connects to link named "mypc".

```
/INF-PROG 'NETEND',*TSN(1234)
```

shuts down Entire Net-Work.

NETEND Utility

It is not normally possible to shut down an Entire Net-Work node without access to either a console or a privileged user ($TSOS).

Entire Net-Work Version 6 provides the NETEND utility procedure, which allows local or remote Entire Net-Work nodes to be shut down without such access. The NETEND procedure can be invoked as follows:

```
CALL-PROC  *LIB($SAG.WCPvrs.LIB,NETEND),(
                 ADALIB=<Adabas or WAL library>,
                 IDTNAME= <idtname>,
                 NETLIB=$SAG.WCPvrs.LIB,
                 TARGETID= <node number>)
```

NETEND uses the Programmable Command Interface (PCI). If you wish to use this utility, see the NODE statement keywords PASSWORD, ENDCMS, and REMCMD in the section *Programmable Command Interface* in the *Entire Net-Work Application Programming Guide*.

7 Zap Naming Convention

The naming convention used for Entire Net-Work zaps is:

```
Wcvrsnnn
```

Each of the letters in this naming convention is now described.

W

The letter W is used to indicate that the zap is for the Software AG Entire Net-Work product.

c

Specifies a system component code that identifies the Entire Net-Work component to which the zap applies. The following table describes the component codes that might be used in an Entire Net-Work zap:

Code	Entire Net-Work Component
D	z/VSE-dependent mainline
M	Operating system-independent mainline
O	OS/390 and z/OS-dependent mainline
T	Operating system-independent, Simple Connection Line Driver
U	Simple Connection Line Driver z/OS-dependent
V	Simple Connection Line Driver z/VSE-dependent
Y	Optional zaps, placed in the ZAPOPT member in the WCP*vrs*.MVSSRCE library.
Z	Simple Connection Line Driver BS2000-dependent

v

Identifies the version number of Entire Net-Work.

r

Identifies the release number of Entire Net-Work.

s

Identifies the system maintenance (SM) level number of Entire Net-Work.

nnn

Specifies the actual zap number. For most Entire Net-Work components, these are simply incremental numbers.

For the Simple Connection Line Driver zaps, the numbers are assigned in hundreds, based on the operating system to which they apply. The following table describes which series is used for whichSimple Connection Line Driver zap.

Zap Name Starting With	Platform	Number Series Used
WT	operating system-independent	000 - 099
WU	z/OS-dependent	100 - 199
WV	z/VSE-dependent	200 - 299
WZ	BS2000-dependent	300 - 399

Index