

Entire Net-Work Administration

Installation

Version 6.4.1

October 2022

This document applies to Entire Net-Work Administration Version 6.4.1 and all subsequent releases.

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

Copyright © 2022 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

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

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

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

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

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

Document ID: WCPMF-INSTALL-641-20220203

Table of Contents

Entire Net-Work Installation	v
1 About this Documentation	1
Document Conventions	2
Online Information and Support	2
Data Protection	3
2 System Requirements	5
Supported Operating System Platforms	6
Supported Hardware	7
Adabas Requirements	7
3 Installation Overview	9
Installation Checklist	10
Using System Maintenance Aid	10
The Entire Net-Work Installation Media	11
Contents of the Release Media	11
Running Entire Net-Work	13
4 z/OS Environments	15
Library Requirements	16
Installation Procedure	16
APF Authorization	21
Execution Job Example	22
Entering Operator Commands	24
Installing Entire Net-Work for zIIP	25
5 z/VSE Environments	27
Installation Procedure	28
Copying the Installation Media Contents to Disk	31
Preparing and Installing the License File	33
Sample JCS	38
The WALvrs Sublibrary	43
Partition/Address Space Requirements	45
Running the Startup Job	45
Running the Startup Job with the SAGUSER Job Exit	47
Entering Operator Commands	47
Applying Corrective Fixes	48
Assembling and Linking Optional User Exits	51
6 BS2000 Environments	53
Installation Media Information	54
Non-SMA Installation Procedure	55
SMA Installation Instructions	57
Running in BS2000 Environments	57
Starting Entire Net-Work	58
Preparing and Installing the Product License File	63
Output File Concept	65
Entering Operator Commands	65

NETEND Utility 66

7 Zap Naming Convention 67

Index 69

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:

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

For information regarding Software AG product compatibility with IBM platforms and any IBM requirements for Software AG products, please review the [*Product Compatibility for IBM Platforms*](#) web page.

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

1

About this Documentation

■ Document Conventions	2
■ Online Information and Support	2
■ Data Protection	3

Document Conventions

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

Online Information and Support

Product Documentation

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

In addition, you can also access the cloud product documentation via <https://www.software-ag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

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

Tech Community

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

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/publishers/softwareag> and discover additional Software AG resources.

Product Support

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

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

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

2 System Requirements

■ Supported Operating System Platforms	6
■ Supported Hardware	7
■ Adabas Requirements	7

This chapter describes the system requirements of Entire Net-Work.

Supported Operating System Platforms

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

For information regarding Software AG product compatibility with IBM platforms and any IBM requirements for Software AG products, please review the [Product Compatibility for IBM Platforms](#) web page.

Before attempting to install this product, ensure that your host operating system is at the minimum required level. For information on the operating system platform versions supported by Software AG products, complete the following steps.

1. Access Software AG's Empower web site at <https://empower.softwareag.com>.
2. Log into Empower. Once you have logged in, you can expand **Products & Documentation** in the left menu of the web page and select **Product Version Availability** to access the Product Version Availability screen.
3. Use the fields on the top of this screen to filter its results for your Software AG product. When you click the **Search** button, the supported Software AG products that meet the filter criteria are listed in the table below the filter criteria.

This list provides, by supported operating system platform:

- the Software AG general availability (GA) date of the Software AG product;
- the date the operating system platform is scheduled for retirement (OS Retirement);
- the Software AG end-of-maintenance (EOM) date for the product; and
- the Software AG end-of-sustained-support (EOSS) date for the product.



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

This release of the Event Replicator Target Adapter Administration tool is supported in Windows and Linux environments that also support Eclipse.

Supported Hardware

For general information regarding Software AG product compatibility with other platforms and their requirements for Software AG products, visit Software AG's [Hardware Supported](#) web page.

Adabas Requirements

This release of Entire Net-Work requires the Adabas components supplied in Adabas Limited Load Library (WAL) and the SVC from any currently supported version of Adabas. These are distributed on the installation media. For information about currently supported Adabas releases, read *End of Maintenance*, in the *Entire Net-Work Release Notes*.



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 media, unless you have been specifically instructed otherwise by Software AG. Entire Net-Work Version requires Adabas mainframe version 8.3.4 or later or WAL 8.3.4 or later.

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 Media Information* in the *Entire Net-Work Release Notes* and the *Software AG Product Delivery Report* provided with the Entire Net-Work installation media 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.



Important: If an ADALNK batch link routine has been linked or modified by Software AG product modules or user exits, it cannot be used in any application startups of Adabas utility jobs or Adabas, Entire System Server, Adabas Review Hub, or Entire Net-Work nuclei.

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.

3

Installation Overview

■ Installation Checklist	10
■ Using System Maintenance Aid	10
■ The Entire Net-Work Installation Media	11
■ Contents of the Release Media	11
■ Running Entire Net-Work	13

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.

For information regarding Software AG product compatibility with IBM platforms and any IBM requirements for Software AG products, please review the [Product Compatibility for IBM Platforms](#) web page.

2. Unload the installation libraries from the installation media. Refer to the *Software AG Product Delivery Report* 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 media. Sample JCL is provided in each of the platform-specific installation procedure sections.

The Entire Net-Work Installation Media

Refer to the *Software AG Product Delivery Report* that accompanies the installation media for information about the volume serial number, density, media type, data set names, and data set sequence numbers of the media.

The media 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 installation media.

Contents of the Release Media

The following table describes most of the libraries included on the release media. Once you have unloaded the libraries from the media, 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 media. 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 .LDnn	One or more Software AG internal libraries. The vrs in the library name represents the <i>version</i> of the internal library code, which is not necessarily the same as the version of Entire Net-Work
BTE vrs .LDnn	A Software AG internal library. The vrs in the library name represents the <i>version</i> 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 <i>version</i> 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 <i>version</i> 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 Entire Net-Work SAF Security. The vrs in the library name represents the <i>version</i> of Entire Net-Work SAF Security.
WAF vrs .LOAD	The z/OS load library for Entire Net-Work SAF Security. The vrs in the library name represents the <i>version</i> of Entire Net-Work SAF Security.
WAF vrs .SRCE	The z/OS source library for Entire Net-Work SAF Security. The vrs in the library name represents the <i>version</i> of Entire Net-Work 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 <i>version</i> of Adabas Limited Load Library.

Library Name	Description
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 <i>version</i> 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 <i>version</i> of Adabas Limited Load Library.
WCC vrs .LOAD	The z/OS load library for Entire Net-Work CTC Option. The vrs in the library name represents the <i>version</i> 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 <i>version</i> of Entire Net-Work.
WCP vrs .SRCE	The z/OS source library for Entire Net-Work. The vrs in the library name represents the <i>version</i> 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 <i>version</i> 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 <i>version</i> 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 <i>version</i> 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 <i>version</i> 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 <i>version</i> 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 <i>version</i> 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 <i>version</i> of Entire Net-Work XCF Option.

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 parameters* 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 Parameters* 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*

4

z/OS Environments

■ Library Requirements	16
■ Installation Procedure	16
■ APF Authorization	21
■ Execution Job Example	22
■ Entering Operator Commands	24
■ Installing Entire Net-Work for zIIP	25

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 Media Information* in the *Entire Net-Work Release Notes* and the *Software AG Product Delivery Report* 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 Installation Media Contents](#)
- [Step 2: Install the Adabas SVC](#)
- [Step 3: Prepare and Install the Product License File](#)
- [Step 4: Customize the Entire Net-Work Startup Jobs](#)
- [Step 5: Modify the Entire Net-Work Parameters](#)
- [Step 6: Install Appropriate Line Drivers](#)
- [Step 7: Prepare Access Method-Specific Definitions](#)
- [Step 8: Apply Corrective Maintenance](#)

Step 1: Copy the Installation Media Contents

Copy the data sets from the supplied installation medium to your disk before you perform the individual installation procedure for each component to be installed.

The way you copy the data sets depends on the installation method and the medium used:

- If you use System Maintenance Aid (SMA), refer to the copy job instructions provided in the *System Maintenance Aid* documentation.
- If you are not using SMA and want to copy the data sets from CD-ROM, refer to the README.TXT file on the CD-ROM.
- If you are not using SMA and want to copy the data sets from tape, follow the instructions in this section.

This section explains how to copy all data sets from tape to disk.

- [Step 1: Copy Data Set COPY.JOB from Tape to Disk](#)
- [Step 2: Modify hilev.COPY.JOB on Your Disk](#)

■ Step 3: Submit COPY.JOB

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

- Modify the following sample job according to your requirements:

```
//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-volser),
// LABEL=(2,SL)
//SYSUT2 DD DSN=hilev.COPY.JOB,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=disk-volser,
// SPACE=(TRK,(1,1),RLSE),
// DCB=*.SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//
```

where:

tape-volser is the VOLSER of the tape, for example: T12345,
hilev is a valid high-level qualifier, and
disk-volser is the VOLSER of the disk.

- Execute the job to copy the data set COPY.JOB to your disk.

Step 2: Modify hilev.COPY.JOB on Your Disk

- Modify *hilev*.COPY.JOB according to your requirements:

Set EXPDT to a valid expiration date, for example, 99365.

Set HILEV to a valid high-level qualifier, for example, USERLIB.

Set LOCATION to a storage location, for example, STORCLAS=ABC or UNIT=3390,VOL=SER=USR123.

Step 3: Submit COPY.JOB

- Execute `hilev.COPY.JOB` to copy single, multiple, or all data sets 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: 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 media 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 media or separately via an e-mail attachment. Before you can install the license, you must transfer it from e-mail or the installation media 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 media.

➤ **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*, in the *Software AG Mainframe Product Licensing*.
- 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 media, complete the following steps:

- Verify that the license file is stored from the installation media 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*, in the *Software AG Mainframe Product Licensing*.
- 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 4: 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 4: Customize the Entire Net-Work Startup Jobs](#).

Step 4: 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_{vr}s.LOAD library. You must either:
 - Copy LICMAIN and LICUTIL into NETWRK._{vr}s.LOAD; or
 - Concatenate MLC_{vr}s.MVSLOAD with NETWRK._{vr}s.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 3: 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._{vr}s.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 installation media (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 5: 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 6: Install Appropriate Line Drivers

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

Step 7: Prepare Access Method-Specific Definitions

Prepare the required access method-specific definitions.

Step 8: Apply Corrective Maintenance

Apply corrective maintenance.

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

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 6.3 requires Adabas mainframe Version 8 or above) or the Adabas limited library (WAL), as provided on the Entire Net-Work installation media, 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 parameters or it identifies the library member in which the ADARUN parameters are defined. These ADARUN parameters define the interregion communications parameters for Entire Net-Work. See the section *ADARUN Parameters* 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
```

Installing Entire Net-Work for zIIP

This section is organized as follows:

- [Prerequisites](#)
- [Libraries](#)
- [License](#)
- [Installation Steps](#)

Prerequisites

Prerequisite for Entire Net-Work for zIIP is a z13 mainframe with one or more zIIP engines, running z/OS 2.1 or above. Entire Net-Work for zIIP needs one zIIP engine to perform effectively.

To run with zIIP support enabled (e.g., ADARUN parameter `ZIIP=YES`), the Entire Net-Work kernel must be APF-authorized.

Libraries

The support of Entire Net-Work for zIIP requires the following Adabas for zIIP library:

- AZP835.LOAD

Entire Net-Work for zIIP requires that the Adabas SVC contains support for zIIP:

- Module ADASVC from the AZP835.LOAD library supports Adabas Version 8.3 databases.
- If the Adabas SVC used for running Entire Net-Work for zIIP is also to be used for Adabas Version 8.4 databases, use a Version 8.4 ADASVC with the following maintenance applied:
 - ADASVC from ADA842.LOAD: Apply zap AO842003.
 - ADASVC from ADA841.LOAD: Apply zap AO841008.

For the Entire Net-Work for zIIP license check (see [License](#) below), use the Mainframe License Check version 1.3.3 from the following library:

- MLC133.LOAD

License

Entire Net-Work for zIIP requires an associated license file (AZPAD) in addition to the Entire Net-Work license file. This is the same license file that is used for an Adabas nucleus. If the AZPAD license is not provided or erroneous, the kernel will run with Entire Net-Work for zIIP deactivated (Z IIP=NO).

The AZPAD license can be transferred to the mainframe in the same way as the Entire Net-Work license. The license can be made available to the nucleus as a load module with the name AZPADLIC. This is similar to the handling of the Entire Net-Work license load module NETLIC described in *Software AG Mainframe Product Licensing*. Alternatively, the license file can be referred to by a 'DDLAPAD' DD statement in the nucleus job/started task. This is a fallback in case the AZPADLIC module cannot be loaded.

Installation Steps

➤ Perform the following steps to install Entire Net-Work for zIIP

- 1 Add the Adabas for zIIP load library to your Entire Net-Work startup jobs (see also [Libraries](#), above).
- 2 Set up the required license modules or datasets (see [License](#) above).
- 3 Install the Adabas SVC that contains zIIP support.
 - Module ADASVC from the AZP835.LOAD library supports Adabas Version 8.3 databases.
 - Module ADASVC from the ADA842.LOAD library, with zap AO842003 applied, supports Adabas Version 8.4 and Adabas Version 8.3 databases, including Adabas for zIIP.

Alternatively, use module ADASVC from the ADA841.LOAD library, with zap AO841008 applied.

- 4 Specify ADARUN parameter Z IIP=YES.
- 5 Start the Entire Net-Work kernel.

5

z/VSE Environments

■ Installation Procedure	28
■ Copying the Installation Media Contents to Disk	31
■ Preparing and Installing the License File	33
■ Sample JCS	38
■ The WALvrs Sublibrary	43
■ Partition/Address Space Requirements	45
■ Running the Startup Job	45
■ Running the Startup Job with the SAGUSER Job Exit	47
■ Entering Operator Commands	47
■ Applying Corrective Fixes	48
■ Assembling and Linking Optional User Exits	51

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



Note: Entire Net-Work 6.3 SP1 requires that zap WD631001 be installed in z/VSE environments. This zap is needed to successfully complete the NETSIP, which in turn is needed for IUCV to run properly.

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 Installation Media 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 media.

If all of the sublibraries are restored, approximately five cylinders of 3390 DASD space is required. Refer to the *Software AG Product Delivery Report* .

Step 2: Copy the Installation Media Contents to Disk

Copy the data sets from the supplied installation medium to your disk before you perform the individual installation procedure for each component to be installed.

The way you copy the data sets depends on the installation method and the medium used:

- If you use System Maintenance Aid (SMA), refer to the copy job instructions provided in the *System Maintenance Aid* documentation.
- If you are not using SMA and want to copy the data sets from CD-ROM, refer to the README.TXT file on the CD-ROM.

- If you are not using SMA and want to copy the data sets .LIBJ, .LIBR and .LICS from tape, follow the instructions in this section.

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.

- [Step 1: Copy Data Set COPYTAPE.JOB to Disk](#)
- [Step 2: Modify COPYTAPE.JOB on Your Disk](#)
- [Step 3: Submit COPYTAPE.JOB](#)

Step 1: Copy Data Set COPYTAPE.JOB to Disk

- Modify the following sample job according to your requirements:

```
* $$ JOB JNM=LIBRCAT,CLASS=0,                                     +
* $$ 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, and

lib.sublib is the library and sublibrary in which the data set COPYTAPE.JOB is to be stored.

- Execute the job to copy the data set COPYTAPE.JOB to disk.

COPYTAPE.JOB contains the JCL required to copy the data sets .LIBJ, .LIBR and .LICS from tape to disk.

Step 2: Modify COPYTAPE.JOB on Your Disk

- Modify COPYTAPE.JOB according to your requirements and set the disk space parameters as appropriate.

Step 3: Submit COPYTAPE.JOB

- Execute COPYTAPE.JOB to copy the data sets .LIBJ, .LIBR and .LICS 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 ALLOC 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](#).

If the Adabas SVC in use is from Adabas 8.2, the zaps AD825011 or AD826003 must be applied before using the SVC with this version of Entire Net-Work in order to avoid the following error:

```
NET0119I ... Reply cannot be matched to RQE...
```



Note: The SVC from Adabas versions older than 8.2 SP5 cannot be used.

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 vr s 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 *Software AG Product Delivery Report*).



Note: Entire Net-Work 6.3 SP1 requires that zap WD631001 be installed in z/VSE environments. This zap is needed to successfully complete the NETSIP, which in turn is needed for IUCV to run properly.

Copying the Installation Media Contents to Disk

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

If you are not using SMA, follow the instructions in this section.

This section explains how to:

- Copy data set COPYTAPE.JOB from the installation media to disk.
- Modify this member to conform with your local naming conventions.

The JCL in this member is then used to copy all data sets from the installation media to disk. If the data sets for more than one product are delivered on the media, the member COPYTAPE.JOB contains the JCL to unload the data sets for all delivered products from the installation media to your disk, except the data sets that you can directly install from the media (for example, Natural INPL objects). After that, you will have to perform the individual installation procedure for each component.

➤ **To copy all data sets from the installation media to disk:**

- 1 The data set COPYTAPE.JOB (file 5) contains the JCL to unload all other existing data sets from the installation media to disk. To unload COPYTAPE.JOB, use the following sample JCL:

```
* $$ JOB JNM=LIBRCAT,CLASS=0,                                     +
* $$ DISP=D,LDEST=(*,UID),SYSID=1
* $$ LST CLASS=A,DISP=D
// JOB LIBRCAT
* *****
*       CATALOG COPYTAPE.JOB TO LIBRARY
* *****
// ASSGN SYS004,NNN                                             <----- installation media ↵
address
// MTC REW,SYS004
// MTC FSF,SYS004,4
ASSGN SYSIPT,SYS004
// TLBL IJSYSIN,'COPYTAPE.JOB'
// EXEC LIBR,PARM='MSHP; ACC S=lib.sublib'                     <----- for catalog
/*
// MTC REW,SYS004
ASSGN SYSIPT,FEC
/*
/&
```

Substitute a valid installation media address for *NNN* and the library and sublibrary names of the catalog for *lib.sublib*.

- 2 Modify COPYTAPE.JOB to conform to your local naming conventions and set the following disk space parameters before submitting the job.
- 3 Submit COPYTAPE.JOB to unload all other data sets from the media to your disk.

Preparing and Installing the License File

The product license file is supplied on the individual customer installation media or separately via an e-mail attachment. If the license file is provided on an installation media, 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*, in the *Software AG Mainframe Product Licensing* 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,YSRDR
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,YSRDR
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=vol1lib,SHR
// EXEC LIBR,PARM='ACC S=SAGLIB.usr1ib; CATALOG NETLIC.OBJ  R=Y'
$*
CLOSE SYSIPT,READER
// LIBDEF OBJ,SEARCH=(SAGLIB.usr1ib),TEMP
// LIBDEF PHASE,CATALOG=SAGLIB.usr1ib
$*
// 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 $$ E0J

```

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/sublib/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 media
- 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
<i>vrs</i>	The product version, revision, and system maintenance (SP) level
<i>vvvvvv</i>	Specified volume for the Entire Net-Work library
<i>ssss</i>	Starting track number
<i>nnnn</i>	Number of tracks or cylinders for the Entire Net-Work library
<i>cuu</i>	Channel unit address
<i>cccccc</i>	VSAM catalog name
<i>tt</i>	Number of installation media (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
/*
/&
* $$ E0J
```



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(vvvvvvv)              -
    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 Media Restore

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

```
* $$ JOB JNM=RESNET,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB RESNET
// ASSGN SYS005,IGN
// ASSGN SYS006,uuu
// MTC REW,SYS006
// PAUSE MOUNT THE NETWORK 5 INSTALLATION MEDIA
// 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 *Software AG Product Delivery Report* to determine whether any *pppvrs.ZAPS* data sets have been distributed on the installation media. 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 media 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
- SAGUSER Job Exit Facility
- Initialization
- Storage Requirements
- Activating Entire Net-Work Use of Job Exit Processing
- Diagnostic Functions

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 parameters 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 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 NET-PRNT 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.

<i>mshpnumber</i>	The number of the MSHP fix.
<i>module</i>	Entire Net-Work object member to be fixed and then relinked.
<i>number</i>	The ESD number of the CSECT to be zapped.
<i>offset</i>	The hexadecimal offset to the beginning of the fix.
<i>rrrr</i>	The replacement data for the fix.
<i>vrs</i>	The Entire Net-Work version, revision, and system maintenance (SP)
<i>vvvv</i>	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
/*
/&
* $$ E0J
```

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'
/*
/&
* $$ E0J
```

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 WCPvrs 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 Environments

■ Installation Media Information	54
■ Non-SMA Installation Procedure	55
■ SMA Installation Instructions	57
■ Running in BS2000 Environments	57
■ Starting Entire Net-Work	58
■ Preparing and Installing the Product License File	63
■ Output File Concept	65
■ Entering Operator Commands	65
■ NETEND Utility	66

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

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

Refer to the *Entire Net-Work Software AG Product Delivery Report* for the names and sequence of the libraries and files on the installation media.

Installation Media Information

Media Contents

The Entire Net-Work installation media for BS2000 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.

For customers wishing to access networks using TCP/IP IPV6, the minimum requirement is Sockets 2.5.

In addition, for customers running BS2000 Sockets 2.0 only, the preparation of the startup of the Sockets subtask has been extended, so please check the following part of the STARTP procedure:

```

/ 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= WTC&(VERSION).LIB,-
      /                      WCPLIB= WCP&(VERSION).LIB,-
      /                      DUMP  = &(DUMP)), -
      / JOB-CLAS= *STD,-
      / JOB-NAME= NETTCP,-
      / RESOURCE= *PAR(CPU-LIMIT=&TIME),-
      / SYS-OUTP= *DELETE')

```

The operand in bold in the example is new and is required. Without this additional parameter, the socket subtask will fail to start.

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 Installation Media to Disk](#)
- [Step 2: Apply Corrective Maintenance](#)
- [Step 3: Perform the Line Driver-Dependent Installations](#)

Step 1: Copy the Contents of the Installation Media to Disk

Copy the files (data sets) from the supplied installation medium to your disk before you perform the individual installation procedure for each component to be installed.

The way you copy the files depends on the installation method and the medium used:

- If you use System Maintenance Aid (SMA), refer to the copy job instructions provided in the *System Maintenance Aid* documentation.
- If you are not using SMA and want to copy the files from CD-ROM, refer to the README.TXT file on the CD-ROM.
- If you are not using SMA and want to copy the files from tape, follow the instructions in this section.

This section explains how to copy all files from tape to disk.

- [Step 1: Copy 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 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 *Software AG Product Delivery Report*.

- Execute the following commands to convert `SRVvrs.LIBS` into an LMS library:

```
/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`, and
volser is the VOLSER of the tape (see the *Software AG Product Delivery Report*).

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

- Call the procedure `P.COPYTAPE` in the library `SRVvrs.LIB` to copy the procedure `COPY.PROC` to disk:

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

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

Step 3: Copy all Product Files from Tape to Disk

- Enter the procedure `COPY.PROC` to copy all product files to disk:

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

If you use a TAPE-C4 device, you can 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 *Software AG Product Delivery Report*).

Step 3: Perform the Line Driver-Dependent Installations

Perform the line driver-dependent installation procedure and prepare the `DRIVER` and `LINK` statements for 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 `WCPvrs.LIB` and optionally `WTCvrs.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 Environments

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

- Correctly configured TCP/IP environments are prerequisites for starting Entire Net-Work.
- The library `WCPvrs.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.WAL826.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 can be omitted if the WY optional zap 2
* has been applied and the Sockets version is
* greater than or equal to 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    TCPI API=BS2,ACCEPTUI=N,SERVERID=8882,APITRACE=N, -
          OPTIONS1=(0,0,0,0,0,0,1024)
LINK MHK1 DCAM PRONAME=HOST
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 SUBSYSTEM-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.

 **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 (xxxxxxx 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 media 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 media or separately via an e-mail attachment. Before you can install the license, you must transfer it from e-mail or the installation media 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*, in the *Software AG Mainframe Product Licensing*.
- 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 media, 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 `WCPvrs.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 `WCPvrs.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 `MLCvrs.MOD` library. Do either (but not both) of the following:
 - Copy LICMAIN and LICUTIL into the `WCPvrs.LIB` library.
 - Concatenate `MLCvrs.MOD` into the BLSLIB chain with `WCPvrs.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 `WCPvrs.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 `WCPvrs.LIB` job STARTP (J):

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

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

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 <i>WCP_{vr}s</i> .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 (SP) level number of Entire Net-Work.

nnn

Specifies the actual zap number.

Index

A

Adabas 7
 data conversion, 8
AMODE (31), 21
APF authorization, 21

B

BS2000
 entering operator commands, 65
 execution job example, 57
 running in, 57
 starting Entire Net-Work, 58

C

control statements, 13
Converting data
 Adabas 7, 8

D

Data conversion
 Adabas 7, 8

E

Empower
 platform support, 6
Examples
 JCL for copying from the installation media (z/VSE), 31
 NAB ADARUN parameter, 24
 OS execution job, 22

H

hardware support, 7

L

license file
 preparing and installing (z/VSE), 33

M

Microsoft Windows support, 6

N

NETEND utility, BS2000, 66

O

operating system coverage, 6
Operator commands
 BS2000, 65
 in OS environments, 24
 in z/VSE environments, 47
OS
 entering operator commands, 24
 run-time examples, 22

P

platform support, 6
product support
 supported platforms, 6

R

requirements
 operating system coverage, 6
 system, 5
Run job example, OS, 22

S

Starting Entire Net-Work
 BS2000 environment, 58
startup job, z/VSE, 45
support
 platforms supported, 6
 supported hardware, 7
 supported operating systems, 6
 supported platforms, 6
SVC
 parameter, 8
 system requirements, 5

U

UNIX
 supported platforms, 6

Z

z/VSE

- entering operator commands, 47
- installation JCL, 31
- preparing and installing the license file, 33