z/VSE Environments

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

- Installation Procedure

- Preparing and Installing the License File

- Sample JCS

- The WALvrs Sublibrary

- Partition/Address Space Requirements

- Running the Startup Job

- Running the Startup Job with the SAGUSER Job Exit

- Entering Operator Commands

- Applying Corrective Fixes

- Assembling and Linking Optional User Exits

Installation Procedure

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

- Step 1: Define an Installation Library

- Step 2: Copy the Tape Contents to Disk

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

- Step 4: Prepare and Install the Product License File

- Step 5: Install the Adabas SVC (optional)

- Step 6: Customize the NETnnLIB Procedure

- Step 7: Modify the Entire Net-Work Startup Parameters

- Step 8: Perform the Line Driver-Dependent Installations

- Step 9: Prepare Access Method-Specific Definitions

- Step 10: Apply Corrective Maintenance

Step 1: Define an Installation Library

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

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

Step 2: Copy the Tape Contents to Disk

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

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

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

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

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

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

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

where:

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

Step 2: Modify COPYTAPE.JOB

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

Step 3: Submit COPYTAPE.JOB

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

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

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

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

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

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

Step 4: Prepare and Install the Product License File

Prepare and install the product license file as described in *Prepare and Install the License File*.

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* ..

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

Step 6: Customize the NET*nn*LIB Procedure

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

The following specific items must be customized:

- File IDs for the libraries

- Volumes for the libraries

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

Step 7: Modify the Entire Net-Work Startup Parameters

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

Step 8: Perform the Line Driver-Dependent Installations

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

Step 9: Prepare Access Method-Specific Definitions

Prepare the required access method-specific definitions.

Step 10: Apply Corrective Maintenance

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

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

Preparing and Installing the License File

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

Installing the license file.

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

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

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

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

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

```
      * ********************************************************************* 000100
      *                                                                    *
      *  This job assembles/link-edits your NET-WORK License NETLIC        *
      *                                                                    *
      *  Note: The Net-Work license file must have RECFM FB and LRECL=80   *
      *         in order to be processed by NETLICAV.                      *
      *                                                                    *
      *         If received separately and moved to the machine with FTP   *
      *         use the binary option such that the file is not converted  *
      *         to EBCDIC                                                  *
      *                                                                    *
      *  This sample reads the Net-Work license file from a VSE library    *
      *  member.                             *                             *
      *                                                                    *
      *  In the example below, the standard label area is assumed to       *
      *  contain label information for library USERLIB.                    *
      *                                                                    *
      *  The first step produces an assembler input for the Net-Work license*
      *  The second step assembles the license and creates a load module   *
      *  NETLIC.                                                           *
      *                                                                    *
      * ********************************************************************* 000100
      * Replace the following symbols in this manner through the whole JCL: * 000200
      *                                                                    * 000300
      *      Before:            After:                                     * 000400
      *                                                                    * 000500
      *        $+      ------>    /+                                       * 000600
      *        $*      ------>    /*                                       * 000600
      *        $&      ------>    /&                                       * 000700
      *        X $$    ------>    * $$                                     * 000800
      *                                                                    * 000900
      * ********************************************************************* 001000
      *                                                                    * 000900
      * Edit the following sample JCS to your own library and file         * 001400
      * specifications and run the job.                                     001400
      *                                                                    * 000900
      * ********************************************************************* 001000
      X $$ JOB JNM=NETLICAV,DISP=D,CLASS=0
      X $$ LST DISP=D,CLASS=A
      // JOB NETLICAV  GENERATE AN Net-Work LICENSE KEY
      // DLBL SAGLIC,'SAG.MLCvrs.VSE.LIBRARY'
      // EXTENT SYS018
      // ASSGN SYS018,DISK,VOL=vollib,SHR
      // LIBDEF *,SEARCH=(SAGLIC.MLCvrs),TEMP
      // DLBL IJSYSPH,'WORK.FILE.SYSPCH',0
      // EXTENT SYSPCH,volpch,1,0,pchsta,pchtrk
      ASSGN SYSPCH,DISK,VOL=volpch,SHR
      // EXEC LICUTIL,SIZE=LICUTIL
      MAKE
      X $$ SLI MEM=LICENSE.XML,S=USERLIB.sublib
      $+
      END
      $*
      CLOSE SYSPCH,PUNCH
      // DLBL IJSYSIN,'WORK.FILE.SYSPCH',0
      // EXTENT SYSIPT,volpch,1,0,pchsta,pchtrk
      ASSGN SYSIPT,DISK,VOL=volpch,SHR
      // DLBL IJSYSPH,'WORK.FILE.SYSPCH2',0
      // EXTENT SYSPCH,volpc2,1,0,pc2sta,pc2trk
      ASSGN SYSPCH,DISK,VOL=volpc2,SHR
      // OPTION DECK
      // EXEC ASMA90,SIZE=1024K
```

```
$*
CLOSE SYSIPT,SYSRDR
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'WORK.FILE.SYSPCH2',0
// EXTENT SYSIPT,volpc2,1,0,pc2sta,pc2trk
ASSGN SYSIPT,DISK,VOL=volpc2,SHR
// DLBL SAGLIB,'WCP621.Vvrs.VSE.LIBRARY'
// EXTENT SYS018
// ASSGN SYS018,DISK,VOL=vollib,SHR
// EXEC LIBR,PARM='ACC S=SAGLIB.usrlib; CATALOG NETLIC.OBJ  R=Y'
$*
CLOSE SYSIPT,READER
// LIBDEF OBJ,SEARCH=(SAGLIB.usrlib),TEMP
// LIBDEF PHASE,CATALOG=SAGLIB.usrlib
$*
// OPTION CATAL
 PHASE NETLIC,*
 MODE AMODE(24) RMODE(24)
 INCLUDE NETLIC
 ENTRY NETLIC
// EXEC LNKEDT
$*
// EXEC LISTLOG
$*
// JOB RESET
* ******************************************************
*       RESET SYSIPT AND SYSPCH
* ******************************************************
ASSGN SYSIPT,READER
ASSGN SYSPCH,PUNCH
$*
$&
X $$ EOJ
```

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

```
* **********************************************************************
*                                                                      *
*  This job assembles/link-edits your NET-WORK License NETLIC          *
*                                                                      *
*  Note: The Net-Work license file must have RECFM FB and LRECL=80     *
*        in order to be processed by WCPLICAV.                         *
*                                                                      *
*        If received separately and moved to the machine with FTP      *
*        use the binary option such that the file is not converted     *
*        to EBCDIC                                                      *
*                                                                      *
*  This sample reads the Net-Work license file from a sequential disk  *
*  data set.                                                           *
*                                                                      *
*  In the example below, the standard label area is assumed to         *
*  contain label information for library USERLIB.                      *
*                                                                      *
*  The first step produces an assembler input for the Net-Work license *
*  The second step assembles the license and creates a load module     *
*  NETLIC.                                                             *
*                                                                      *
* **********************************************************************
* Replace the following symbols in this manner through the whole JCS:  *
*                                                                      *
*      Before:              After:                                     *
```

```
*                                                                       *
*          $*      ------>    /*                                        *
*          $&      ------>    /&                                        *
*          X $$    ------>     * $$                                     *
*                                                                       *
* **********************************************************************
*                                                                       *
* Edit the following sample JCS to your own library and file           *
* specifications and run the job.                                       *
*                                                                       *
* **********************************************************************
X $$ JOB JNM=NETLICAV,DISP=D,CLASS=0
X $$ LST DISP=D,CLASS=A
// JOB NETLICAV  GENERATE AN NET-WORK LICENSE KEY
// UPSI 1
// DLBL INPSEQ,'WCP621.Vvrs.LICENSE'
// EXTENT SYS018
// ASSGN SYS018,DISK,VOL=vollic,SHR
// EXEC DITTO
$$DITTO SL FILEIN=INPSEQ,
$$DITTO    LIBOUT=USERLIB.usrsublb,MEMBEROUT=LICENSE.WORKFILE
/*
// UPSI 0
// DLBL SAGLIC,'SAG.MLCvrs.VSE.LIBRARY'
// EXTENT SYS018
// ASSGN SYS018,DISK,VOL=vollib,SHR
// LIBDEF *,SEARCH=(SAGLIC.MLCvrs),TEMP
// DLBL IJSYSPH,'WORK.FILE.SYSPCH',0
// EXTENT SYSPCH,volpch,1,0,pchsta,pchtrk
ASSGN SYSPCH,DISK,VOL=volpch,SHR
// EXEC LICUTIL,SIZE=LICUTIL
MAKE
X $$ SLI MEM=LICENSE.WORKFILE,S=USERLIB.usrsublb
$+
END
$*
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'WORK.FILE.SYSPCH',0
// EXTENT SYSIPT,volpch,1,0,pchsta,pchtrk
ASSGN SYSIPT,DISK,VOL=volpch,SHR
// DLBL IJSYSPH,'WORK.FILE.SYSPCH2',0
// EXTENT SYSPCH,volpc2,1,0,pc2sta,pc2trk
ASSGN SYSPCH,DISK,VOL=volpc2,SHR
// OPTION DECK
// EXEC ASMA90,SIZE=1024K
$*
CLOSE SYSIPT,SYSRDR
CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'WORK.FILE.SYSPCH2',0
// EXTENT SYSIPT,volpc2,1,0,pc2sta,pc2trk
ASSGN SYSIPT,DISK,VOL=volpc2,SHR
// DLBL SAGLIB,'WCP621.Vvrs.VSE.LIBRARY'
// EXTENT SYS018
// ASSGN SYS018,DISK,VOL=vollib,SHR
// EXEC LIBR,PARM='ACC S=SAGLIB.usrlib; CATALOG NETLIC.OBJ  R=Y'
$*
CLOSE SYSIPT,READER
// LIBDEF OBJ,SEARCH=(SAGLIB.usrlib),TEMP
// LIBDEF PHASE,CATALOG=SAGLIB.usrlib
$*
// OPTION CATAL
 PHASE NETLIC,*
```

```
 MODE AMODE(24) RMODE(24)
 INCLUDE NETLIC
 ENTRY NETLIC
// EXEC LNKEDT
$*
// EXEC LISTLOG
$*
// JOB RESET
* *******************************************************
*       RESET SYSIPT AND SYSPCH
* *******************************************************
ASSGN SYSIPT,READER
ASSGN SYSPCH,PUNCH
$*
$&
X $$ EOJ
```

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

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

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

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

2. Make sure any previously-created NETLIC phase is inaccessible in the Entire Net-Work load library
    being used by the Entire Net-Work startup jobs. Entire Net-Work first tries to load NETLIC and if
    unsuccessful it reads from DDLIC.

3. Provide all Entire Net-Work startup jobs with a DLBL statement in the following format:

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

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

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

1. Verify that the license file is stored in a sequential file (with RECFM=F or FB and LRECL=80),
    taking care to preserve its format as ASCII.

2. Make sure any previously created NETLIC phase is inaccessible in the Entire Net-Work load library
    being used by the Entire Net-Work startup jobs. Entire Net-Work first tries to load NETLIC and, if
    unsuccessful, it reads from DDLIC.

3. Provide all Entire Net-Work startup jobs with DLBL, EXTENT and ASSGN statements in the
    following format:

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

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

Sample JCS

This section provides sample JCS for:

- defining the Entire Net-Work library as a non-VSAM library

- defining the Entire Net-Work library as a VSAM library

- restoring the Entire Net-Work installation tape

- defining the Entire Net-Work catalog procedure

- installing the Adabas SVC

- unloading the ZAPS data sets

User Substitution Variables

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

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

Non-VSAM Library

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

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

```
  DEF S=SAGLIB.WCPvrsZ R=Y
  DEF S=SAGLIB.WTCvrs R=Y
  DEF S=SAGLIB.WALvrs R=Y
/*
/&
* $$ EOJ
```

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

VSAM Library

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

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

Installation Tape Restore

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

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

```
// ASSGN SYS006,UA
/*
/&
* $$ EOJ
```

Note:

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

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

Note:

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

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

Catalog Procedure Definition

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

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

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

Adabas SVC

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

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

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

Unload ZAPS Data Sets

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

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

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

The WAL*vrs* Sublibrary

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

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

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

- WTOVSE

- 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 control statements in place of SYSIPT.

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

Initialization

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

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

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

The following sample JCS initializes the job exit:

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

Storage Requirements

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

Activating Entire Net-Work Use of Job Exit Processing

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

Diagnostic Functions

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

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

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

Partition/Address Space Requirements

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

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

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

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

Running the Startup Job

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

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

Note:
All diagnostic information from tracing, logging, and abends is written to the NETPRNT file if it is open (otherwise it is written to the DDPRINT file). NETPRNT can be allocated to a large data set that can be

copied when closed. To do this, allocate the file SHR. This causes the data set to be erased at the time the file is opened. Be aware that the diagnostic information is very large and will fill a data set quickly. When this happens, the file is closed and all additional output is sent to DDPRINT. This diagnostic information is created by Entire Net-Work and does not include the operating system dump information written to SYSUDUMP.

Running the Startup Job with the SAGUSER Job Exit

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

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

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

Entering Operator Commands

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

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

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

   ```
   MSG Fn
   ```

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

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

```
CONNECT V2ZHOST
```

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

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

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

Example:

```
msg F5
5 NETI24 04003 READY FOR OPERATOR COMMUNICATION
5 D STATS
NET0090I: BUFFER USAGE STATISTICS:
NET0091I: ASYNCH. BUFFERS:     000016 (= 24.2 %) OF 000064 K USED
NET0091I: LONG TERM BUFFERS:   000000 (=  0.4 %) OF 000064 K USED
NET0091I: SHORT TERM BUFFERS:  000000 (=  6.1 %) OF 001025 K USED
NET0091I: ATTACHED BUFFERS:    000000 (= 11.9 %) OF 000080 K USED
NET0091I: REQUEST QUEUE:       000000 (=  6.0 %) OF 000050 RQES USED
NET0087I: 0000010847 REQUESTS FROM LOCAL RQ
5
```

Applying Corrective Fixes

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

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

● Step 1: Prepare Entire Net-Work for Maintenance

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

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

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

Step 1: Prepare Entire Net-Work for Maintenance

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

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

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

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

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

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

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

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

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

2. The MSHP fix relinks the affected phases using one or more of the jobs found in the Entire Net-Work WCPvrs sublibrary, member type X (LINKxxxx). Each fix will indicate the job used to relink the phase.

3. If a fix needs to be removed, use the following sample job, which uses the MSHP UNDO facility, to remove the MSHP fix from the WCP*vrs* component.

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

Assembling and Linking Optional User Exits

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