If you are installing Wave 2 elements in target and distribution zones that contained a previously installed release, you must replace the existing DDDEF entries for the JES2, JES3, and SDSF libraries of previous releases with the DDDEF entries for the libraries of current release.

If you use sample job HASIDDEF, IATJDDEF, or ISFISDDD to replace the existing DDDEF entries, you must change the ADD to REP in the sample job before running the job.

A condition code of 0 is expected if the DDDEF entries are successfully added to the SMP/E CSI when the sample jobs are run.

8.3.6 Set High Level Assembler Option for SDSF

For SDSF to assemble correctly, the High Level Assembler option must be set to OPTABLE(UNI); OPTABLE(ESA) will not work.

8.3.7 Set High Level Assembler Option for JES2

For JES2 to assemble correctly, set the SMP/E Global Utility ASMA90 options in the GLOBAL zone in the SMP/E CSI that is used for the installation of the z/OS release as follows:

- GOFF
- LIST(133)
- DECK
- NOOBJECT

In addition, you must do one of the following actions:

- Specify a unique output file using the print operand - for example, PRINT(ASMPRINT) - and create a DDDEF in the target and DLIB zones for ASMPRINT, specifying SYSOUT
- Add the following JCL DD card to the APPLY jobs:

```
//SYSPRINT DD SYSOUT=*,LRECL=133,RECFM=FBA
```

8.3.8 APPLY Wave 2

The following sections describe the steps needed to APPLY Wave 2.

8.3.8.1 Do an SMP/E APPLY CHECK for Wave 2:

Run an APPLY CHECK to identify any requisite service, and additional holds (for example, HOLDSYS(DOC)), that may need to be resolved before APPLY processing. Resolve any holds and receive any requisite service identified by the APPLY CHECK before the next step.

Figure 36 on page 118 shows a sample APPLY CHECK for the functions specified in the SELECT operand, plus received PTFs that are applicable only to the FMIDs listed in the FORFMID.

```
//CHECK JOB <job parameters>
//STEP1 EXEC PGM=GIMSMP,REGION=0M,TIME=NOLIMIT
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR,
//         UNIT=SYSALLDA,VOL=SER=v2r5vol1
//        DD DSN=ASM.SASMMOD1,DISP=SHR,
//         UNIT=SYSALLDA,VOL=SER=v2r5vol1
//SMPCSI DD DSN=zosv2r5.global.csi,DISP=SHR
//SMPCNTL DD *
  SET BOUNDARY(targetzone)
      OPTIONS(ZOSOPT) .
  APPLY CHECK  XZREQ
        FORFMID(HJE77D0,HQX77D0,HJS77D0,
                JJE77DJ)  /* If not ordered, remove */
        SELECT(HJE77D0,HQX77D0,HJS77D0,
                JJE77DJ)  /* If not ordered, remove */
  SOURCEID(ZOSV2R5,RSU*)
  FIXCAT(IBM.ProductInstall-RequiredService)
  GROUPEXTEND(NOAPARS,NOUSERMODS)
  BYPASS(HOLDSYSTEM,HOLDUSER,
  HOLDCLASS(ERREL,UCLREL,HIPER)) .
/*
```

Figure 36. SMP/E APPLY CHECK for Wave 2 (all FMIDs and All Service)

Note: ZOSOPT is the option name for z/OS.

Required Updates

1. Update the job parameters.
2. Either remove the STEPLIB (if you are using the Wave 1 target system to install Wave 2) or update the v2r5vol1 with the volume serial number for the MIGLIB and the SASMMOD1 libraries which were the targets of the Wave 0 install. This will allow you to access the z/OS V2R5 level of the Wave 0 elements using the STEPLIB DD statements.
3. Replace the CSI name on the SMPCSI DD statement with your CSI name.
4. Update targetzone to your target zone name.
5. The XZREQ operand only needs to be specified when cross-zone processing is required.
6. If you plan to use JES2 and SDSF only, remove the JES3 FMID HJS77D0 from the FORFMID and SELECT operands in the sample job above before running the job.

Note that if you BYPASS(HOLDCLASS(HIPER)), you should run the SMP/E REPORT ERRSYSMODS command to identify missing HIPER HOLDs before putting your system into production.

If you do not BYPASS(HOLDCLASS(HIPER)), the FMIDs may not be installed if any of the HIPER maintenance is unavailable.

Any messages other than those listed in 6.7, “Step 6: Review General Installation Notes” on page 43 or listed below should be investigated.

Notes:

1. The XZREQ operand only needs to be specified when cross-zone processing is required. If this operand is specified when there is no zone group set up, the following messages will be received, which are acceptable:

```
GIM50810W THE XZREQ OPERAND WAS SPECIFIED ON
THE APPLY COMMAND BUT SINCE NO ZONES WERE APPLICABLE
FOR CROSS-ZONE REQUISITE CHECKING,
THE XZREQ OPERAND WILL BE IGNORED.

GIM20501I APPLY PROCESSING IS COMPLETE.
THE HIGHEST RETURN CODE WAS 04.
```

Successful APPLY CHECK processing of JES2 and SDSF returns a condition code of 0. If JES3 is included in the APPLY CHECK, then successful APPLY CHECK processing returns a condition code of 4.

8.3.8.1.1 Messages expected during APPLY CHECK Processing for JES3: During the installation of the JES3 component you can expect to receive the following messages:

```
GIM61903W LMOD IATUX86 WAS NOT DELETED BY
SYSMOD HJS77D0 BECAUSE IT IS NOT IN THE target ZONE.

GIM61903W LMOD IATIMDL WAS NOT DELETED BY
SYSMOD HJS77D0 BECAUSE IT IS NOT IN THE target ZONE.
```

These messages are acceptable. Any messages other than the messages listed above or the messages identified in 6.7, “Step 6: Review General Installation Notes” on page 43 must be investigated.

A condition code of 4 is expected during APPLY CHECK processing for JES3.

8.3.8.2 Do an SMP/E APPLY for Wave 2:

Do not specify ASSEM on the APPLY command for JES2 or JES3. The specification of ASSEM on the APPLY command can cause serviceability problems.

Be certain that all exception conditions have been satisfied before adding a BYPASS(HOLDSYSTEM) during the SMP/E APPLY step.

Figure 37 on page 120 shows a sample APPLY for the functions specified in the SELECT operand, plus received PTFs that are applicable only to the FMIDs listed in the FORFMID.

```
//APPLY JOB <job parameters>
//STEP1 EXEC PGM=GIMSMP,REGION=0M,TIME=NOLIMIT
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR,
//        UNIT=SYSALLDA,VOL=SER=v2r5vol1
//        DD DSN=ASM.SASMMOD1,DISP=SHR,
//         UNIT=SYSALLDA,VOL=SER=v2r5vol1
//SMPCSI DD DSN=zosv2r5.global.csi,DISP=SHR
//SMPCNTL DD *
  SET BOUNDARY(targetzone)
      OPTIONS(ZOSOPT) .
  APPLY  XZREQ
        FORFMID(HJE77D0,HQX77D0,HJS77D0,
                JJE77DJ)  /* If not ordered, remove */
        SELECT(HJE77D0,HQX77D0,HJS77D0,
               JJE77DJ)   /* If not ordered, remove */
  SOURCEID(ZOSV2R5,RSU*)
  FIXCAT(IBM.ProductInstall-RequiredService)
  BYPASS(HOLDSYSTEM,HOLDUSER,
  HOLDCLASS(ERREL,UCLREL,HIPER))
  GROUPEXTEND(NOAPARS,NOUSERMODS)
  COMPRESS(ALL) .
/*
```

Figure 37. SMP/E APPLY for Wave 2 (FMIDs and All Service)

Note: ZOSOPT is the option name for z/OS.

Required Updates

1. Update the job parameters.
2. Either remove the STEPLIB (if you are using the Wave 1 target system to install Wave 2) or update the v2r5vol1 with the volume serial number for the MIGLIB and the SASMMOD1 libraries which were the targets of the Wave 0 install. This will allow you to access the z/OS V2R5 level of the Wave 0 elements using the STEPLIB DD statements.
3. Replace the CSI name on the SMPCSI DD statement with your CSI name.
4. Update targetzone to your target zone name.
5. The XZREQ operand only needs to be specified when cross-zone processing is required.
6. If you plan to use JES2 and SDSF only, remove the JES3 FMID HJS77D0 from the FORFMID and SELECT operands in the sample job above before running the job.

Note that if you BYPASS(HOLDCLASS(HIPER)), you should run the SMP/E REPORT ERRSYSMODS command to identify missing HIPER HOLDs before putting your system into production.

If you do not BYPASS(HOLDCLASS(HIPER)), the FMIDs may not be installed if any of the HIPER maintenance is unavailable.

Any messages other than those listed in 6.7, "Step 6: Review General Installation Notes" on page 43 or those listed below should be investigated.

Notes:

1. The XZREQ operand only needs to be specified when cross-zone processing is required. If this operand is specified when there is no zone group set up, the following messages will be received, which are acceptable:

   ```
   GIM50810W THE XZREQ OPERAND WAS SPECIFIED ON
   THE APPLY COMMAND BUT SINCE NO ZONES WERE APPLICABLE
   FOR CROSS-ZONE REQUISITE CHECKING,
   THE XZREQ OPERAND WILL BE IGNORED.

   GIM20501I APPLY PROCESSING IS COMPLETE.
   THE HIGHEST RETURN CODE WAS 04.
   ```

Successful APPLY processing of JES2 and SDSF returns a condition code of 0. If JES3 is included in the APPLY processing, then successful APPLY processing returns a condition code of 4.

8.3.8.2.1 Messages expected during APPLY Processing for JES3

During the installation of the JES3 component you can expect to receive the following messages:

```
GIM61903W  LMOD IATUX86 WAS NOT DELETED BY
SYSMOD HJS77D0 BECAUSE IT IS NOT IN THE target ZONE.

GIM61903W  LMOD IATIMDL WAS NOT DELETED
BY SYSMOD HJS77D0 BECAUSE IT IS NOT IN THE target ZONE.

GIM23903W LINK-EDIT PROCESSING FOR
SYSMOD xxxxxxx WAS SUCCESSFUL
FOR MODULE xxxxxxxx IN LMOD xxxxxxxx
IN THE xxxxxx LIBRARY. THE RETURN CODE WAS 04.
```

Note: This message might be GIM23903I, depending on the SMP/E service level you have on your driving system.

You might receive either message GIM23903W or the following GIM23913W message for load modules IATNUCF and IATNUCI:

```
GIM23913W LINK-EDIT PROCESSING FOR
SYSMOD HJS77D0 WAS SUCCESSFUL
FOR MODULE xxxxxxxx IN LMOD xxxxxxxx IN THE SIATLIB LIBRARY.
THE RETURN CODE WAS 04.
DATE yy.ddd - TIME hh:mm:ss -
SEQUENCE NUMBER nnnnnn - SYSPRINT FILE nnnnnnnn.
```

These messages are acceptable. Because of the following IEW2454W messages, and the resulting condition code, SMP/E will produce message GIM23903W for every load module that was link-edited in the same utility invocation.

During the installation of the JES3 component you might receive IEW2454W messages (unresolved external references) during the link-edit of load module IATNUCF. The references are:

```
DSDDDR DSIFCT IATCNINX LJMFFCT SMFBUFAB SMFBUFBL
TAPDDR  VARYLFCT
```

You might also receive an IEW0461 or IEW2454W message during the link-edit of load module IATNUCI with an unresolved external reference for IATINIT. These messages are acceptable.

8.4 Step 4: Do Post-APPLY work for Wave 1 and Wave 2

Once you have successfully SMP/E APPLY'd all the Wave 1 elements and Wave 2 elements, you must perform the Post-APPLY activities. Post-APPLY work for Wave 1 and Wave 2 must be run from your **driving** system unless otherwise specified.

Post-APPLY jobs include:

- Wave 1A:
  - Compile MMS data sets
  - Run Post-Apply for Communications Server IP Services
- Wave 1B:
  - There are no Post-APPLY steps for Wave 1B FMIDs.
- Wave 1C:
  - Run Post-APPLY link-edit for FFST
  - Run TSO/E Information Center Facility Post-APPLY ICQPOST1 and ICQPOST2 (when appropriate)
- Wave 1D:

  Cryptographic Services PKI Services:
  - Run sample job to allocate VSAM datasets for runtime use.
  - Run sample REXX exec to define RACF user IDs and profiles.
- General:
  - Run sample job, CLNCRSZ, an SMP/E REPORT CROSSZONE

Note: IBM has verified that every product that is now part of z/OS and that supplies CALLLIB'd libraries is upwardly compatible. Therefore, a LINK LMODS CALLLIBS is not required.

8.4.1 Wave 1A Post-Installation jobs

8.4.1.1 Compile MMS Data Sets

After installing BCP, if you choose to use the MVS Message Service (MMS) the message files must be compiled into runtime message files. If you will be installing additional elements (such as TSO/E or JES2) into the message files, it is only necessary to compile these message files once. After the installation of the additional products or elements, compilation can begin.

For the MVS message service (MMS) to handle translated messages, your installation must use the MVS message compiler to format and install message files that contain English message skeletons and translated language message skeletons. The following summarizes the steps for providing translated messages:

- Verify the message files are correctly installed on your system.
- Allocate space for each runtime message file. Edit and run the job in member CNLDEFCL to allocate runtime message files. CNLDEFCL is supplied in your SAMPLIB library on the target system after APPLY processing has completed.
- Use the MVS message compiler to format the installation message file into a runtime message file. Edit and run the job in member CNLCOMP to compile and load the runtime message files. CNLCOMP is supplied in your SAMPLIB library on the target system after APPLY processing has completed.
- Create installation exit routines, if needed.
- Create or update the SYS1.PARMLIB members to initialize values for MMS:
  CNLLSTxx define the available languages for message translation. A sample CNLLSTXX is provided in the SYS1.SAMPLIB library.
  CNLcccxx define the date and time formats.
  CONSOLxx specify the CNLLSTxx member in effect for the system.
- Activate MMS.

See the description of how to activate MMS, creating installation exits, and how to handle translated messages in *z/OS MVS Planning: Operations*.

8.4.1.2 Run Post-APPLY for Communications Server IP Services

The following sections describe the post-apply steps for Communications Server IP Services.

8.4.1.2.1 Perform SMP/E LINK for IMS™ module

The Communications Server IP Services IMS sockets interface requires an IMS load module during link-edit. The APPLY for LMOD EZAIMSLN will receive a return code of 0004. After the APPLY, these modules are not executable. At this point, you are required to run a postlink job to make the Communications Server IP Services IMS module be fully resolved. This is accomplished by executing EZAIMSCZ, supplied in *TCPIP*.SEZAINST. The EZAIMSCZ job invokes the LINK function supplied with SMP/E.

EZAIMSCZ performs a cross-zone link-edit. Communications Server IP Services should not be installed into the same SMP/E zone as IMS. SMP/E tracks this cross-zone link and if subsequent IMS maintenance is applied, SMP/E can automatically maintain the affected Communications Server IP Services load modules. For the automatic maintenance to occur, the IMS global zone must contain zoneindex subentries to the z/OS target and distribution zones and must have an options entry for ZOSOPT. The z/OS TARGETZONE XZLINK entry must be set to AUTOMATIC (it defaults to DEFERRED). See the *z/OS SMP/E Reference* for more information on XZLINK.

Before you submit EZAIMSCZ, your z/OS global zone containing Communications Server IP Services must contain zoneindex subentries for the target and distribution zones containing the IMS libraries. The index entries may be added using SMP/E interactive dialogs or by using JCL and SMP/E statements.

EZAIMSCZ may end with a condition code of 0004 caused by SMP/E warning message:

```
GIM69064W MODULE aaaaaa WILL BE LINK-EDITED
BUT ITS SERVICE LEVEL OR FMID IN THE DISTRIBUTION ZONE IS
DIFFERENT FROM THAT IN THE TARGET ZONE.
```

There are two possible solutions:

1. Investigate and solve the problem indicated by GIM69064.
2. Ignore the return code of 0004.

Complete the following steps:

- Create a ZONEINDEX in the z/OS global zone for the IMS target and distribution zones.
- Create a ZONEINDEX in the IMS global zone for the z/OS target and distribution zones.
- Submit the job and ensure you have a return code of 00.

8.4.2 Wave 1B Post-Installation jobs

There are no post-installation jobs for Wave 1B elements.

8.4.3 Wave 1C Post-Installation jobs

The following sections describe post-installation jobs for Wave 1C elements.

8.4.3.1 Run Post-APPLY Link-Edit for FFST

The Post-APPLY link-edit step might need to be run when maintenance is performed on FFST. Follow the specific instructions in the maintenance being installed. IEW2646W and IEW2651W messages may be generated indicating that there is a mismatch between the RMODE and AMODE values. This is not a problem and these messages can be ignored.

8.4.3.2 Run Post-APPLY for TSO/E Information Center Facility (when appropriate)

To use the Information Center function of TSO/E, you must allocate the data sets listed in Figure 38 on page 125 if they are not present on your system. The attributes for these data sets must conform to the attributes of your ISPF/PDF table libraries. The minimum space allocation for each data set is 221 blocks. You may want to allocate more space to allow room for future expansion.

The tables distributed with the Information Center Facility are copied to the data sets in Figure 38 on page 125 by the ICQPOST1 and ICQPOST2 jobs. These jobs are supplied in your SAMPLIB library on the target system after APPLY processing has completed.

Figure 38. Information Center Facility Data Sets

Data Set	Description
ICQ.ICQABTAB	Course abstracts
ICQ.ICQGCTAB	User requests
ICQ.ICQAATAB	Names
ICQ.ICQANTAB	News
ICQ.ICQTLIB	User enrollment tables
ICQ.ICQAPTAB	Printer support
ICQ.ICQAMTAB	Administrator Application Manager tables
ICQ.ICQCMTAB	User Application Manager tables

Note: New Information Center Facility installations can execute ICQPOST1 and ICQPOST2 to distribute Information Center Facility parts to the appropriate execution libraries. If you are a previous user of the Information Center Facility, and you want to preserve customized data, do not execute the ICQPOST1 and ICQPOST2 jobs. These jobs replace tables that contain current Information Center Facility data for your installation.

8.4.4 Wave 1D Post-Installation jobs

The following section describes post-installation jobs for Wave 1D of the installation process.

8.4.4.1 Run Cryptographic Services PKI Services job

Run IKYCVSAM, the sample job to allocate VSAM data sets for runtime use. Sample job IKYCVSAM can be found in the SAMPLIB library after APPLY processing has completed.

8.4.5 Wave 1E Post-Installation jobs

There are no post-installation jobs for Wave 1E elements.

8.4.6 Wave 1F Post-Installation jobs

There are no post-installation jobs for Wave 1F elements.

8.4.7 Wave 1G Post-Installation jobs

There are no post-installation jobs for Wave 1G elements.

8.4.8 Wave 2 Post-Installation jobs

There are no post-installation jobs for Wave 2 elements.

8.4.9 Wave 1 and Wave 2 general Post-Installation jobs

The following sections describe post-installation jobs for Wave 1 and Wave 2.

8.4.9.1 Run SMP/E REPORT CROSSZONE (Target Zone)

If you did not use automatic cross-zone IFREQ checking when installing z/OS V2R5 Wave 1, you may run sample job CLNCRSZ, and see *z/OS SMP/E Commands*, for information on the SMP/E REPORT CROSSZONE command. For instructions on how to set up cross-zone processing, see 7.3.1, "Create a cross-zone set" on page 54.

8.5 Step 5: Customize Wave 1 and Wave 2

Once you have successfully SMP/E APPLY'd the Wave 1 and Wave 2 elements and performed the Post-APPLY work, you must customize these elements. Customization for Wave 1 and Wave 2 may be run from your **driving** system unless otherwise specified.

It is important that you have read the migration requirements documented in *z/OS Upgrade Workflow*.

8.5.1 Required setup

The following sections describe steps that are required to set up the system.

8.5.1.1 Reassemble stand-alone dump

When you install z/OS V2R5, you must reassemble the stand-alone dump program. For information on stand-alone dump, see *z/OS MVS Diagnosis: Tools and Service Aids*.

8.5.1.2 Complete DFSMSdss actions

If you are enabling DFSMSdss, ensure that the following tasks are completed.

If you intend to use the DFSMSdss stand-alone services, use the DFSMSdss BUILDSA function to build the IPLable stand-alone services core image. Perform this step after DFSMSdss is installed and accepted by SMP/E. Stand-alone services supports the following tape subsystems:

- 3494 Enterprise Automated Tape Library
- 3495 Enterprise Automated Tape Library
- 3590 Enterprise Tape Subsystem

Restriction: Stand-alone services does not allow you to create the core image on an SMS-managed volume.

Perform the following Migration Actions:

1. Prepare for stand-alone services by creating an IPLable core image with the BUILDSA command. You can specify the device (card reader,tape drive, or DASD volume) from which stand-alone services are IPLed. You also can specify the operator console to be used for stand-alone services.
2. Use RACF or another security product to protect the SYS1.ADR.SAIPLD.Vvolser data set and for the stand-alone services modules.
3. Make a backup copy of your system that can be restored by this function.

For the procedure, see the chapter in *z/OS DFSMSdss Storage Administration*, entitled "Managing Availability with DFSMSdss", specifically the section, "Backing up Volumes". For details, see *z/OS DFSMSdfp Storage Administration*.

8.5.1.3 Write new IPL TEXT

You must run IPLTEXT to write a new copy of the IPL text before IPLing your newly built z/OS system. When using the REFORMAT command with the IPLDD parameter to write IPL Text on a volume, if IPL Text already exists, message ICK21836D is issued which requires a reply before the existing text can be overlayed.

```
ICK21836D IPL TEXT EXISTS ON volid. REPLY U TO OVERLAY, ELSE T.
```

If you wish to suppress the message ICK21836D, you can specify the IPLTXTEXIST parameter with REFORMAT command.

```
//IPLTEXT  JOB <job parameters>
//*MESSAGE AUTHORIZE IPLTEXT UPDATE FOR volser(JES2 user)
//*OPERATOR AUTHORIZE IPLTEXT UPDATE FOR volser(JES3 user)
//STP1   EXEC PGM=ICKDSF,REGION=0M
//SYSPRINT DD SYSOUT=*
//IVOL DD VOL=SER=volser,UNIT=SYSALLDA,DISP=SHR
//SAMPLIB DD DSN=SYS1.SAMPLIB(IPLRECS),DISP=SHR,UNIT=SYSALLDA,
//           VOL=SER=vvvvvv
//        DD DSN=SYS1.SAMPLIB(IEAIPL00),DISP=SHR,UNIT=SYSALLDA,
//           VOL=SER=vvvvvv
//SYSIN DD *
 REFORMAT DDNAME(IVOL) VERIFY(volser) IPLDD(SAMPLIB,OBJ) -
 BOOTSTRAP
/*
```

Figure 39. Sample JCL for Writing the IPL Text

Required Updates

1. Update the *job parameters*.
2. Replace *vvvvvv* to your volume.
3. Replace *volser* to your IPL volume.

8.5.1.4 Create and update your IODF

IODFs created on prior z/OS releases can be used to IPL z/OS V2R5. For more information on IODF compatibility and how to create an IODF, see *z/OS HCD User's Guide*. If you are migrating to HCD, have OEM processor definitions in your current IODF, and have not received instruction from your OEM vendor, you should contact your IBM account team for additional information from the HCD development group. This, however, only applies to OEM processors that have been defined in the IODF using a non-IBM processor type/model.

8.5.2 PARMLIB member considerations

After z/OS V2R5 is installed, you must tailor the system to meet your installation's needs. You will need to review, modify, and create entries in SYS1.PARMLIB, SYS1.PROCLIB, and SYS*n*.IPLPARM. Some functions of z/OS V2R5, such as APPC/MVS, require that you copy (and possibly modify) members from SYS1.SAMPLIB for use in other libraries.

z/OS provides the capability to concatenate up to ten additional data sets to SYS1.PARMLIB at IPL, creating a “logical PARMLIB.” A logical PARMLIB is a concatenation of parameter libraries that can be accessed through a set of system services. The concatenation is defined in the LOAD*xx* PARMLIB member at system initialization but can be changed later with a SET command. Programs that use these services can access PARMLIB members without being aware of which data sets actually comprise the concatenation. You can use separate PARMLIB data sets to isolate IBM-supplied and SMP/E-supported members from locally customized members. For information on PARMLIB concatenation, see *z/OS MVS Initialization and Tuning Reference*.

SMP/E installation places a number of members in the PARMLIB pointed to by the PARMLIB DDDEF in your target zone, or the PARMLIB DD statement in your SMP/E procedure. This PARMLIB data set is a copy of your production PARMLIB, as described in 6.1, “Overview for the Clone of Your System” on page 31. Before IPLing your production system, you must do one of the following:

1. Concatenate the PARMLIB referenced by the SMP/E DDDEF.
2. Copy the members that SMP/E installed into your production PARMLIB concatenation.
3. Update your existing members to reflect any needed changes.

The following is a complete list of PARMLIB members that were shipped with the exclusive element FMIDs when this program directory was written. It is possible that a PTF could add or change a PARMLIB member, so you should also check the Element Summary Report in your SMP/E APPLY output for any new elements with a type of ++PARM to ensure that this list is complete when you perform your installation.

For a list of all samples used in PARMLIB and shipped in SYS1.SAMPLIB, refer to Figure 41 on page 130.

8.5.2.1 What is new for z/OS V2R5: For a summary of PARMLIB changes in this release, see *z/OS MVS Initialization and Tuning Reference*.

8.5.2.2 z/OS V2R5 PARMLIB members: Figure 40 shows the SMP/E supported PARMLIB members that are installed directly into SYS1.PARMLIB.

Figure 40. PARMLIB Members Installed Directly in SYS1.PARMLIB

ADYSET00	ADYSET01	ADYSET02	ANTFTSO	ANTIMAIM	ANTMIN00
ANTPTSO	ANTXIN00	ANTXTSO	ASAIPCSP	ASBIPCSP	ATBIPCSP
AUTOR00	BDTIPCSP	BHIIPCSP	BLSCECT	BLSCECTX	BPXIPCSP
CEAPRM00	CEEIPCSP	CLOCK00	CNLENU00	CNLJPN00	COFDLF00
COFIPCSP	COFVLF00	COUPLE00	CSFIPCSP	CTIAXR00	CTIBPX00
CTIBPX01	CTICBR00	CTICEA00	CTICSF00	CTIDMD00	CTIDMP00
CTIEZBTN	CTIEZB00	CTIGLZ00	CTIGRS00	CTIHWI00	CTIHZS00
CTIIDS00	CTIIEAVX	CTIIEFAL	CTIIKE00	CTIITT00	CTIJES01
CTIJES02	CTIJES03	CTIJES04	CTILOG00	CTINFC00	CTINFS00
CTINSS00	CTINTA00	CTIOPS00	CTIORA00	CTIRES00	CTIRSMSP
CTIRSM01	CTISMS00	CTIXCF00	CTIXES00	DIAG00	DIAG01
DIAG02	ERBRMF00	ERBRMF01	ERBRMF02	ERBRMF03	ERBRMF04
EZAIPCSP	GLZIPCSP	GPMSRV00	GRSCNF00	GRSRNL00	GTFPARM
GTZPRM00	GXLIPCSP	HASLIPCS	HZSPRM00	IARIPCSP	IATIPCSI
IATIPCSP	IAXIPCSP	IAZIPCSP	ICHIPCSP	IDAVDT00	IEAABD00
IEACMD00	IEADMP00	IEADMR00	IEAIPCSP	IEAOPT00	IEAPAK00
IEASLP00	IEAVIPCS	IEFIPCSA	IEFIPCSI	IEFIPCST	IEFOPZ01
IEFSSN00	IGWIPCSP	IPCSPR00	ISGIPCSP	IXCIPCSP	IXGIPCSP
IXLIPCSP	IXZIPCSP				

Attention

If you do not make necessary changes to SYS1.PARMLIB, SYS1.PROCLIB, and SYS*n*.IPLPARM, the system might not initialize or run successfully. For example:

- If you do not create a LOAD*xx* PARMLIB member, your system will not IPL.
- If the copies of BLSCECT and all the other IPCS PARMLIB members used on your production system are not current, IPCS might fail when you attempt to use it.

When you migrate to z/OS V2R5, you must complete certain actions before loading the initial program for the z/OS V2R5 target system. For a complete description of all changes needed for a migration, see *z/OS Upgrade Workflow*.

For information on PARMLIB members, see *z/OS MVS Initialization and Tuning Reference*.

Figure 41 describes the PARMLIB updates that are recommended for z/OS V2R5. For additional information on PARMLIB considerations, see the individual elements' customization books and *z/OS MVS Initialization and Tuning Guide*. The PARMLIB members shown are for example only. Be sure to change the high-level qualifier of the data sets to match your installation.

Figure 41 (Page 1 of 11). PARMLIB Updates

PARMLIB Member	**TEXT**	**Element Name**
APPCPM*xx*	See SAMPLIB members APPCPMRX and APPCPMXX.	BCP
ASCHPM*xx*	See SAMPLIB members ASCHPMRX and ASCHPMXX.	BCP
AXR00	See SAMPLIB member AXR00.	BCP
BPXPRM*xx*	FILESYSTYPE TYPE(ZFS) ENTRYPOINT(IOEFSCM)	BCP (see note)
Notes: 1. The FILESYSTYPE statement allows you to mount a ZFS which is needed in Wave 1 and Wave 2.		
COFVLF*xx*	CLASS NAME(CSVLLA) EMAJ(LLA)	BCP
	CLASS NAME(IRRGTS) EMAJ(GTS) CLASS NAME(IRRGMAP) EMAJ(GMAP) CLASS NAME(IRRSMAP) EMAJ(SMAP) CLASS NAME(IRRUMAP) EMAJ(UMAP) CLASS NAME(IRRACEE) EMAJ(ACEE)	RACF (see note)
Note: The RACF sample is provided in member RACPARM of SAMPLIB.		

Figure 41 (Page 2 of 11). PARMLIB Updates

<table>
<tr><th>PARMLIB Member</th><th>TEXT</th><th>Element Name</th></tr>
<tr><td rowspan="5">COMMND<i>xx</i></td><td>COM='S EPWFFST.FFST,SUB=MSTR'</td><td>FFST (see note 1)</td></tr>
<tr><td>COM='S EZAZSSI,P=nodename'</td><td>Communications Server IP Services</td></tr>
<tr><td>COM='S RMF.RMF,,,MEMBER(00)'</td><td>RMF</td></tr>
<tr><td>COM='S SDSF'</td><td>SDSF (see note 2)</td></tr>
<tr><td>COM='S ICEOPT,ICEPRM=xx'
COM='S ICEOPT,ICEPRM=(x1,...,xn)'</td><td>DFSORT (see note 3)</td></tr>
<tr><td colspan="3"><b>Notes:</b>
1. Start procedure FFSTPROC contains DD statements for SYSPRINT and SYSABEND that contain SYSOUT=A. If you want FFST to be started by the master scheduler, change the DD statements for SYSPRINT and SYSABEND so that they contain data set information.
2. Use the 'S SDSF' command to start the SDSF server whenever the system is IPLed. The SDSF server must be active on every system for which SDSF is to display data.
3. You can use the 'S ICEOPT,ICEPRM=xx' command to activate an ICEPRMxx member that contains changes to the installation defaults of DFSORT. You can use the 'S ICEOPT,ICEPRM=(x1,...,xn)' command to activate multiple ICEPRMxx members that contain changes to the installation defaults of DFSORT. See <i>z/OS DFSORT Installation and Customization</i> for details.</td></tr>
<tr><td rowspan="5">CTI<i>xxxxx</i></td><td>Default PARMLIB members provided: CTIAXR00, CTIBPX00, CTIBPX01, CTICEA00, CTIDMP00, CTIGRS00, CTIHWI00, CTIHZS00, CTIIEAVX, CTIIEFAL, CTIITT00, CTIJES01, CTIJES02, CTIJES03, CTIJES04, CTILOG00, CTIOPS00, CTIRSMSP, CTIRSM01, CTIXCF00, CTIXES00</td><td>BCP</td></tr>
<tr><td>Default PARMLIB member provided: CTICBR00 CTISMS00</td><td>DFSMS</td></tr>
<tr><td>Default PARMLIB members provided: CTIDMD00, CTIEZBTN, CTIEZB00, CTIIDS00, CTIIKE00, CTINSS00, CTINTA00, CTIORA00, CTIRES00</td><td>Communications Server IP Services</td></tr>
<tr><td>Default PARMLIB members provided: CTINFC00 (for NFS Client), CTINFS00 (for NFS Server)</td><td>NFS</td></tr>
<tr><td>Default PARMLIB members provided: CTICSF00</td><td>Cryptographic Services ICSF</td></tr>
<tr><td>ERBRMF<i>xx</i></td><td>Default PARMLIB members provided: ERBRMF00, ERBRMF01, ERBRMF02, ERBRMF03, ERBRMF04</td><td>z/OS Data Gatherer</td></tr>
<tr><td>GPMSRV<i>xx</i></td><td>Default PARMLIB member GPMSRV00 provided for RMF Distributed Data Server.</td><td>RMF</td></tr>
<tr><td>GTZPRM00</td><td>Recommended PARMLIB member GTZPRM00 provided for IBM Generic Tracker for z/OS</td><td>BCP</td></tr>
<tr><td>HZSPRM00</td><td>Default PARMLIB member HZSPRM00 provided for IBM Health Checker for z/OS</td><td>BCP</td></tr>
</table>

Figure 41 (Page 3 of 11). PARMLIB Updates

PARMLIB Member	**TEXT**	**Element Name**
*xxx*IPCSP	Default PARMLIB members: ASAIPCSP, ASBIPCSP, ATBIPCSP, BHIIPCSP, BPXIPCSP, COFIPCSP, GXLIPCSP, IARIPCSP, IAXIPCSP, IAZIPCSP, IEAIPCSP, IEAVIPCS, IEFIPCSA, IEFIPCSI, IEFIPCST, ISGIPCSP, IXCIPCSP, IXGIPCSP, IXLIPCSP, IXZIPCSP	BCP
	Default PARMLIB member: BDTIPCSP	BDT
	Default PARMLIB member: EZAIPCSP	Communications Server IP Services
	Default PARMLIB member: IGWIPCSP	DFSMS
	Default PARMLIB member: CSFIPCSP	Cryptographic Services ICSF
	Default PARMLIB member: HASLIPCS	JES2
	Default PARMLIB members: IATIPCSI, IATIPCSP	JES3
	Default PARMLIB member: CEEIPCSP	Language Environment
	Default PARMLIB member: ICHIPCSP	RACF
	Copy member from SISFJCL: ISFIPCSP	SDSF
IEASYS*xx*	Update suffixes as required to specify updated PARMLIB members OMVS=xx. OMVS=xx specifies BPXPRMxx parmlib member.	z/OS UNIX System Services
Note: The OMVS parameter specifies the PARMLIB member or members to use to locate the PARMLIB statements to configure the z/OS UNIX System Services kernel.		
IECIOSxx	`MIH TIME=mm:ss,DEV=(dddd-dddd)`	Communications Server IP Services
Note: Update IECIOSxx with the ranges of addresses used on the Communication devices. For a description of the appropriate advice with respect to MIH, see *z/OS Communications Server: IP Configuration Reference*		

Figure 41 (Page 4 of 11). PARMLIB Updates

PARMLIB Member	TEXT	Element Name
IEFSSN*xx*	SUBSYS SUBNAME(JES2) PRIMARY(YES) START(YES) SUBSYS SUBNAME(IRLM) SUBSYS SUBNAME(JRLM)	BCP (see note 1)
	SUBSYS SUBNAME(BDT)	BDT

	SUBSYS SUBNAME(SMS)
	INITRTN(IGDSSIIN)
	INITPARM('ID=xx,PROMPT=DISPLAY')
	SUBSYS SUBNAME(OAM1)
	INITRTN(CBRINIT)
		DFSMS
		(see note 3)

	SUBSYS SUBNAME(FFST)	FFST

	SUBSYS SUBNAME(RACF)
	  INITRTN(IRRSSI00)
	  INITPARM('%,X')
		RACF
		(see note 2)

	SUBSYS SUBNAME(TNF) SUBSYS SUBNAME(VMCF) or TNF VMCF	Communications Server IP Services (see note 4)

Notes:

1. The BCP member is provided in member IEFSSN00 of PARMLIB.
2. An additional RACF sample is provided in member RACPARM of SAMPLIB.
3. where xx refers to your IGDSMSxx member of PARMLIB. See *z/OS MVS Initialization and Tuning Reference* for more information about IEFSSNxx.
4. VMCF and TNF are restartable beginning with TCP/IP 3.2 and with this enhancement, display and easy modification of the VMCF and TNF subsystems are possible. See *z/OS Communications Server: IP Configuration Guide* for more information and command syntax associated with the restartable VMCF and TNF enhancements.

Figure 41 (Page 5 of 11). PARMLIB Updates

<table>
<tr><th>PARMLIB Member</th><th>TEXT</th><th>Element Name</th></tr>
<tr><td>IFAPRD*xx*</td><td>To define the enablement policy for base elements and features. For details, see 8.5.2.3, “IFAPRDxx considerations” on page 140.</td><td>BDT SNA NJE
BDT File-to-File
XL C/C++
DFSMSdss
DFSMShsm
DFSMSrmm
DFSMStvs
DFSORT
GDDM-PGF (*)
GDDM-REXX
HCM
HLASM Toolkit (*)
Infoprint Server
Communications Server IP Services
Communications Server IP Services CICS Sockets
Communications Server IP Services IMS Sockets
JES3
SDSF
RMF
Security Server RACF
Restricted Use Common Service Area (RUCSA)
zEnterprise Data Compression (zEDC)
z/OS Advanced Data Gatherer</td></tr>
<tr><td colspan="3"><b>Notes:</b>
• (*) indicates the element may register as a z/OS V2R5 feature or as a stand-alone product.
• Communications Server IP Services is the element name for TCP/IP, but the name entry for TCP/IP remains unchanged in the IFAPRDxx member.
• IFAPRD00 is built by CBPDO RIMLIB job ZOSREG.</td></tr>
<tr><td>IGDSMS*xx*</td><td><code>SMS ACDS(sys1.xxx.ACDS)</code>
<code>COMMDS(sys1.xxx.COMMDS)</code></td><td>DFSMS (see note below)</td></tr>
<tr><td colspan="3"><b>Note:</b> <code>sys1.xxx.ACDS</code> is the name of your Active Control Dataset and <code>sys1.xxx.COMMDS</code> is the name of your Communications data set. See <i>z/OS MVS Initialization and Tuning Reference</i> for more information about IGDSMSxx.</td></tr>
</table>

Figure 41 (Page 6 of 11). PARMLIB Updates

PARMLIB Member	TEXT	Element Name
IKJTSO*xx*	AUTHCMD NAMES(BINDDATA BDATA EXPORT EXP IMPORT IMP LISTD LISTDS LISTDATA LDATA PRINT REPRO RMM SETCACHE SETC SHCDS VERIFY VFY	DFSMS
	AD ADDSD AG ADDGROUP AU ADDUSER ALG ALTGROUP ALD ALTDSD ALU ALTUSER BLKUPD CO CONNECT DD DELDSD DG DELGROUP DU DELUSER IRRDPI00 LD LISTDSD LG LISTGRP LU LISTUSER RACDCERT RACLINK RACMAP RACPRIV RALT RALTER RDEF RDEFINE RDEL RDELETE RE REMOVE RL RLIST RVARY PASSWORD PW PHRASE PE PERMIT SETR SETROPTS SR SEARCH	RACF

Figure 41 (Page 7 of 11). PARMLIB Updates

PARMLIB Member	**TEXT**	**Element Name**
IKJTSO*xx* (continued)	LPQ LPR LPRM MODDVIPA MVPXDISP PING RSH TRACERTE	Communications Server IP Services
	ADYOPCMD CONSPROF LISTB LISTBC IKJPRMLB PARMLIB RACONVRT RECEIVE SE SEND SYNC TESTAUTH TESTA TRANSMIT XMIT)	TSO/E
	AUTHPGM NAMES(	
	AOPCMND	Infoprint Server
	CSFDAUTH CSFDPKDS	Cryptographic Services ICSF
	EDGAUD EDGBKUP EDGHSKP EDGRPTD EDGUPDT EDGUTIL	DFSMS RMM
	GIMSMP	SMP/E
	ICHDSM00 IRRDPI00 IRRUT100 IRRUT200 IRRUT400 IRRXUTI2	RACF
	IOEAGFMT IOEAGSLV IOEFSUTL IOEZADM)	z/OS File System (see note below)

Figure 41 (Page 8 of 11). PARMLIB Updates

<table>
<tr><th>PARMLIB Member</th><th>TEXT</th><th>Element Name</th></tr>
<tr><td>IKJTSO*xx*
(continued)</td><td>AUTHTSF NAMES(</td><td></td></tr>
<tr><td></td><td>CSFDAUTH
CSFDPKDS</td><td>Cryptographic Services ICSF</td></tr>
<tr><td></td><td>EDGAUD
EDGBKUP
EDGHSKP
EDGRPTD
EDGUPDT
EDGUTIL</td><td>DFSMS RMM</td></tr>
<tr><td></td><td>GIMSMP</td><td>SMP/E</td></tr>
<tr><td></td><td>ICQASLI0
IKJEFF76)</td><td>TSO/E</td></tr>
<tr><td colspan="3"><b>Note:</b> The following updates specifically apply to the support listed below:
<b>zFS</b> IOEAGSLV, IOEAGFMT, IOEFSUTL, IOEZADM</td></tr>
<tr><td colspan="3"><b>Note:</b> For more information on using IKJTSO<i>xx</i>, see <i>z/OS MVS Initialization and Tuning Reference</i>. See <i>z/OS MVS Initialization and Tuning Guide</i> for information on specifying HELP data sets in IKJTSO<i>xx</i>. The commands in the IKJTSO<i>xx</i> PARMLIB member should be modified as required for your installation. The entries shown in this table for the IKJTSO<i>xx</i> PARMLIB member are the recommended values at the time this program directory was written. For the latest information, refer to the sample IKJTSO00 member in SYS1.SAMPLIB after the product is installed.</td></tr>
<tr><td>IRROPT<i>xx</i></td><td>ALLOCATE FILE(SYSUT1)
DATASET
('SYS1.SAMPLIB(IRRDPSDS)')
IRRDPI00 UPDATE
FREE FILE(SYSUT1)</td><td>RACF</td></tr>
<tr><td>ISFPRM<i>xx</i></td><td>Copy members ISFPRM00, ISFPRM01 from SISFJCL.</td><td>SDSF</td></tr>
<tr><td colspan="3"><b>Note:</b> By default, SDSF reads the ISFPRMxx members from SYS1.PARMLIB and any data set concatenated to PARMLIB. ISFPRM00 is the default unless overridden on the start command or changed in the JCL. You can use a different partitioned data set by adding it to the server JCL using ddname SDSFPARM.</td></tr>
<tr><td>IVTPRM00</td><td>Default PARMLIB member IVTPRM00 is provided. For details, see <i>z/OS Communications Server: New Function Summary</i>.</td><td>Communications Server for z/OS SNA Services</td></tr>
<tr><td>IZUPRM<i>xx</i></td><td>See SAMPLIB member IZUPRM00</td><td>IBM z/OS Management Facility (z/OSMF)</td></tr>
</table>

Figure 41 (Page 9 of 11). PARMLIB Updates

PARMLIB Member	TEXT	Element Name
LNKLST*xx* or PROG*xx*	SYS1.CMDLIB	BCP, DFSMS, TSO/E
	SYS1.SBDTLINK	BDT
	CBC.SCCNCMP	XL C/C++ without Debug Tool
	CBC.SCLBDLL CBC.SCLBDLL2	Runtime Library Extensions
	SYS1.SICELINK	DFSORT
	SYS1.SEPWMOD2	FFST
	SYS1.SEPWMOD3	FFST
	GDDM.SADMMOD	GDDM
	ASM.SASMMOD1	HLASM
	TCPIP.SEZALOAD	Communications Server IP Services
	ISP.SISPLOAD	ISPF
	SYS1.SHASLNKE SYS1.SHASMIG	JES2
	SYS1.SIATLIB SYS1.SIATLINK SYS1.SIATMIG	JES3
	CEE.SCEERUN CEE.SCEERUN2	Language Environment
	REXX.SEAGALT	Alternate Library for REXX
	SYS1.SERBLNKE	RMF
	SYS1.SGRBLINK	z/OS Data Gatherer
	ISF.SISFLOAD	SDSF
	SYS1.DFQLLIB	ISMF
	SYS1.DGTLLIB	ISMF
	SYS1.SGIMLMD0	SMP/E

Notes:

- If the LNKAUTH parameter in the IEASYS*xx* member of SYS1.PARMLIB does not specify LNKLST, you must add these data sets to the APF table to make them APF-authorized. See *z/OS MVS Initialization and Tuning Guide* for more information.
- See 8.5.2.5, “LNKLSTxx considerations” on page 143.

Figure 41 (Page 10 of 11). PARMLIB Updates

PARMLIB Member	TEXT	Element Name
LOAD*xx*	For detailed information, see *z/OS MVS Initialization and Tuning Reference*.	BCP
Note: Parmlib symbolic preprocessor can be used to set up and validate the new LOAD*xx* PARMLIB member.		
LPALST*xx*	SYS1.SBDTLPA	BDT
	SYS1.SICELPA SYS1.SORTLPA (see note 3)	DFSORT
	TCPIP.SEZALPA	Communications Server IP Services
	ISP.SISPLPA	ISPF
	SYS1.SIATLPA	JES3
	CEE.SCEELPA (see note 1)	Language Environment
	ISF.SISFLPA (see note 2) ISF.SISFLOAD	SDSF
	SYS1.SDWWDLPA	DFSMS
	SYS1.SGRBLPA	z/OS Data Gatherer

Notes:

1. CEE.SCEELPA should be added to LPALSTxx only if CEE.SCEERUN is added to LNKLSTxx. See 8.5.2.5.1, "Making the Run-time library available" on page 143.
2. To execute SDSF from the link pack area (LPA) for improved performance, add ISF.SISFLOAD in your LPALST*xx* member in SYS1.PARMLIB. In z/OS V2R3 SDSF, if SISFLOAD is in the LPALST concatenation, it must be specified in the STEPLIB DD statement in SDSF and SDSFAUX address spaces. But it has been changed in z/OS V2R4 SDSF so that there is no need for the STEPLIB. As of V2R5, SDSF cannot be loaded using a STEPLIB. ISF.SISFLPA must be in the lpalst and ISF.SISFLOAD must be in the lnklst. ISF.SISFLOAD can optionally be added to the lpalst.
3. To make DFSORT programs available on the system:
   - To make DFSORT resident, load the modules from SORTLPA into the link pack area (LPA) or dynamic LPA, and add SIEALNKE before the SICELINK to the link list.
   - To make DFSORT nonresident, add SIEALNKE before the SICELINK and SORTLPA (in this order) to the link list (this assumes SORTLPA is removed from the LPA list), or use a JOBLIB or STEPLIB for SIEALNKE, SICELINK and SORTLPA (in this order).

   If you want to make DFSORT SVC accessible at IPL, put SICELPA into the LPA or modified link pack area (MLPA).

Figure 41 (Page 11 of 11). PARMLIB Updates

PARMLIB Member	**TEXT**	**Element Name**
PROG*xx* or IEAAPF*xx*	All LNKLST*xx* (or PROG*xx*) libraries. See item 3 on page 144 in 8.5.2.6, "PROGxx and IEAAPFxx considerations" on page 144.	All
	All STEPLIB or ISPLLIB libraries. See Figure 44 on page 149 for the appropriate data sets.	All
	SYS1.SBDTLIB SYS1.SBDTLINK	BDT
	CSF.SCSFMOD0	Cryptographic Services ICSF
	SYS1.SEPWMOD1	FFST
	TCPIP.SEZATCP TCPIP.SEZALNK2 TCPIP.SEZADSIL	Communications Server IP Services
	SYS1.SIEALNKE	All
	SYS1.SERBLNKE	RMF
	SYS1.SGRBLINK	z/OS Data Gatherer
	SYS1.VTAMLIB SYS1.SISTCLIB	Communications Server for z/OS SNA Services
	SYS1.NFSLIBE	Network File System
	ISF.SISFLOAD	SDSF
Note: See 8.5.2.6, "PROGxx and IEAAPFxx considerations" on page 144.		
SCHED*xx*	The default program properties table, IEFSDPPT, shipped by z/OS V2R5 BCP includes entries for all z/OS elements. Therefore, the SCHED*xx* entries that were previously required by elements are no longer required. See *z/OS MVS Initialization and Tuning Reference* for more details of the specific entries that are defaulted to.	

8.5.2.3 IFAPRDxx considerations: With z/OS, products can use registration services to determine if they are enabled to run on a particular system. This requires the product be defined appropriately in the enablement policy for the system using the IFAPRD*xx* PARMLIB member.

Note: When the IFAPRD*xx* PARMLIB member is updated, an IPL is required for the Integrated Accelerator for zEDC device driver to recognize the enablement.

Customers must ensure the policy in IFAPRD*xx* enables only that which they are licensed. Use of (and enablement of) z/OS features is subject to the z/OS license terms and conditions and must be done with the knowledge of your asset manager according to the terms and conditions for z/OS. See the Usage Restriction section of the *z/OS Licensed Program Specifications* for additional license terms and conditions.

When you order an IBM product that packages priced optional features with the base product (such as z/OS), IBM supplies a tailored IFAPRD00 PARMLIB member that enables the product and any optional features ordered with the product. Thus, any feature ordered with the product is enabled during installation when you copy the contents of the tailored IFAPRD00 member to an active IFAPRD*xx* member and issue SET command or IPL. If you do not order a feature, it is shipped disabled but you can enable it later. *z/OS MVS Product Management* contains information on how to enable a z/OS feature and how to discontinue use of a feature.

Therefore, you must ensure IFAPRD*xx* is set up properly. CBPDO provides a job, ZOSREG, in the RIMLIB data set to place a tailored IFAPRD00 member into your SYS1.PARMLIB. If you ordered any optional features that use the register service, the STATE statement is set as ENABLED for you to use the product; otherwise the STATE is set as DISABLED. This job must be run before IPL, or the SET command be used if it has been run after the IPL, or some elements and features will not work. Furthermore, if you added any other products to your IFAPRD00 PARMLIB member and they are still applicable, you will need to make those changes to the member built by ZOSREG or use multiple IFAPRD*xx* members.

See *z/OS Planning for Installation* for additional information on enabling z/OS features.

IBM supplies you with a tailored IFAPRD00 parmlib member that has been customized for your order. You should use this IFAPRD00 parmlib member if you have determined from the above information that it is required for your environment.

8.5.2.4 BPXPRMxx updates:

z/OS provides the capability to dynamically add Physical File Systems to a UNIX System Services configuration. SETOMVS RESET=(yy) is a console command designed to be used with a subset of BPXPRMxx parmlib statements. FILESYSTYPE, SUBFILESYSTYPE, NETWORK, and system limits (such as MAXPROCSYS, etc.) can be changed dynamically by issuing SETOMVS RESET=(xx). An updated or new parmlib member, BPXPRMyy, can be concatenated with other parmlib members at initial IPL or through the SET OMVS=(xx,yy,...) command.

You must copy the sample BPXPRMxx member from SYS1.SAMPLIB to your PARMLIB, renaming it to fit your environment (BPXPRMxx). Then, create or update another PARMLIB member, BPXPRMyy, with the following information for elements z/OS UNIX System Services Application Services, Communications Server IP Services, z/OS File System, and Network File System. The BPXPRMyy member can then be added dynamically by issuing the SETOMVS RESET=(yy) command. Also, you will need to update your IEASYSxx PARMLIB member to contain the OMVS=(yy,xx,...) parameter for future IPLs. For more information, see *z/OS UNIX System Services Planning* and *z/OS MVS Initialization and Tuning Guide*.

```
MAXTHREADTASKS(500)
MAXTHREADS(500)
MAXPROCUSER(100)
FILESYSTYPE TYPE(UDS) ENTRYPOINT(BPXTUINT)
FILESYSTYPE TYPE(INET) ENTRYPOINT(EZBPFINI)
  STEPLIBLIST('etc/steplist')                /* See Note 2 */
FILESYSTYPE TYPE(ZFS) ENTRYPOINT(IOEFSCM)
ASNAME(ZFS)                                  /* See Note 4 */
   NETWORK DOMAINNAME(AF_UNIX)
```

```
                DOMAINNUMBER(1)
                MAXSOCKETS(2000)
                TYPE(UDS)
        NETWORK DOMAINNAME(AF_INET)
                DOMAINNUMBER(2)
                MAXSOCKETS(64000)
                TYPE(INET)
        IPCSEMNSEMS(50)
        IPCSHMMPAGES(2048)
FILESYSTYPE TYPE(NFS) ENTRYPOINT(GFSCINIT)
        PARM(' ')
        ASNAME(GFSCPROC)
        ASNAME(GFSCPROC)
```

Notes:

1. The values listed in the previous figure are the minimum values needed to load the initial program.
2. `STEPLIBLIST('etc/steplib')` will need to be added under `FILESYSTYPE(INET) ENTRYPOINT(EZBPFINI)`.
3. For installation parameters, see *z/OS Network File System Guide and Reference*.
4. As of z/OS V2R2, this entry is required to run ZFS in its own address space. The ASNAME keyword can be left off to run ZFS in the OMVS address space. For more information, see *z/OS File System Administration*.
5. See *z/OS UNIX System Services Planning* for information about how to specify the appropriate MAXFILEPROC value in the BPXPRMxx PARMLIB member.

If you have more than one AF_NET transport providers on an MVS image, you must use the common INET, CINET. CINET is defined in BPXPRMxx to identify the use of the Common INET physical file system. For example, in the preceding created PARMLIB member, BPXPRMyy will need to be updated instead with the following information:

```
FILESYSTYPE TYPE (CINET) ENTRYPOINT(BPXTCINT)
    NETWORK DOMAINNAME(AF_INET)
        DOMAINNUMBER(2)
        MAXSOCKETS(64000)
        TYPE(CINET)
        INADDRANYPORT(4000)                  /* SEE NOTE */
        INADDRANYCOUNT(2000)                 /* SEE NOTE */
    SUBFILESYSTYPE NAME(tcpjob1)
                   TYPE(CINET)
                   ENTRYPOINT(EZBPFINI)
    SUBFILESYSTYPE NAME(tcpjob2)
                   TYPE(CINET)
                   ENTRYPOINT(EZBPFINI)
    SUBFILESYSTYPE NAME(tcpjob3)
                   TYPE(CINET)
                   ENTRYPOINT(EZBPFINI)
    SUBFILESYSTYPE NAME(STC1)
```

```
TYPE(CINET)
ENTRYPOINT(EZBPFINI)
```

Note: The INADDRANYPORT assignment is coordinated with similar PORT assignments in the PROFILE.TCPIP data set. For customizing the Communications Server IP Services file system statements, see *z/OS UNIX System Services Planning* and *z/OS MVS Initialization and Tuning Guide*.

8.5.2.5 LNKLSTxx considerations

1. In addition to the data sets you specify in LNKLSTxx or on LNKLST statements in PROGxx, the system automatically places these data sets at the beginning of the link list (unless overridden by a SYSLIB statement in PROGxx):

   - SYS1.LINKLIB
   - SYS1.MIGLIB
   - SYS1.CSSLIB
   - SYS1.SIEALNKE
   - SYS1.SIEAMIGE

   Note: As of z/OS V1R8, data sets SYS1.SIEALNKE and SYS1.SIEAMIGE MUST be cataloged to ensure that these data sets are located during IPL.

2. TCPIP.SEZALNK2 is optional but not recommended. The member in SEZALNK2 is reserved for the NCS administrator and is not copied to your existing LINKLIB data sets. The NCS administrator must have a separate TSO logon procedure, with SEZALNK2 added to the STEPLIB concatenation. SEZALNK2 must be RACF-protected so that unauthorized users are denied access. You must also APF authorize the SEZALNK2 library.

3. To make DFSORT programs available on the system, use either of the following procedures:

   - To make DFSORT resident, load the modules from SORTLPA into the link pack area (LPA) or dynamic LPA, and add SIEALNKE before the SICELINK to the link list.
   - To make DFSORT nonresident, add SIEALNKE before the SICELINK and SORTLPA (in this order) to the link list (this assumes SORTLPA is removed from the LPA list), or use a JOBLIB or STEPLIB for SIEALNKE, SICELINK and SORTLPA (in this order).

8.5.2.5.1 Making the Run-time library available

Many elements, features and functions of z/OS V2R5 require the run-time libraries provided by Language Environment, SCEERUN and SCEERUN2, to be made available in the program search order. The best way to do that is by adding the SCEERUN and SCEERUN2 data sets in LNKLST. Below lists the z/OS V2R5 elements, features and functions in Wave 1 that require Language Environment. The list was generated by analyzing which elements use the SMP/E CALLLIBS function to include Language Environment stubs and reviewing Post-APPLY link edit jobs supplied by elements.

Wave 1 elements that require Language Environment

- BCP
- BCP - Capacity Provisioning
- XL C/C++

- C/C++ Host Performance Analyzer
- Communications Server IP Services
- Communications Server for z/OS SNA Services
- Cryptographic Services
- Cryptographic Services ICSF
- DFSMS (Character Data Representation Architecture, CDRA)
- DFSORT Locale Processing
- HCD
- IBM HTTP Server Base
- Infoprint Server
- IBM TDS
- ISPF
- Network File System
- RMF
- Security Server (RACF)
- z/OS Security Level 3 System SSL Security Level 3
- z/OS UNIX System Services

8.5.2.5.1.1 z/OS Elements and Features Using STEPLIB for Language Environment

The following sections list the z/OS elements and features that use STEPLIB for Language Environment.

Network File System

Add SCEERUN to the startup procedures; shipped samples are

- GFSAPROC (server)
- GFSCPROC (client)

8.5.2.6 PROGxx and IEAAPFxx considerations

1. See *z/OS MVS Initialization and Tuning Guide* for information about using the PROGxx and IEAAPFxx PARMLIB members.

2. If the default for the LNKAUTH system parameter is taken (LNKAUTH=LNKLST) or is specified in IEASYS*xx*, or by the operator, libraries in the LNKLST concatenation are authorized when accessed as part of the LNKLST concatenation. However, if they are accessed outside the LNKLST concatenation (for example, through a STEPLIB) they will not be APF authorized unless they are specified in PROG*xx* or IEAAPF*xx*. Therefore, you must include any LNKLST libraries that are to be accessed outside the LNKLST in PROG*xx* or IEAAPFxx.

3. Except for concatenations opened during system initialization, an unauthorized library concatenated to any authorized libraries will cause the system to consider all the concatenated libraries to be **unauthorized**. Therefore, if one library in a concatenation must be APF authorized, you must include all libraries in that concatenation in PROG*xx*.

8.5.3 PROCLIB member considerations

This section describes the PROCLIB customization that must be performed for the Wave 1 and Wave 2 elements. PROCLIB customization consists of three steps:

1. Ensure the default PROCLIB members have been copied to your default PROCLIB to pick up the new and changed members.
2. Update individual sample members provided and ensure they are accessible to the system, as shown in Figure 42.
3. Ensure entire libraries are accessible to the system, as shown in Figure 43 on page 149.

The following figure, shown in ripple order, describes the PROCLIB updates that are required for z/OS V2R5. Note that PARMLIB concatenation function has been incorporated into some sample procedures. **IBM recommends** you base your customized procedures on the IBM-supplied samples. For additional information on PROCLIB considerations, see the customization books for the particular element.

8.5.3.1 Copying default PROCLIB members:

After the PROCLIB updates are complete, the members listed in Figure 42 must be made accessible to the system by copying them to a procedure library that is in your JES procedure library concatenation.

Figure 42 (Page 1 of 4). PROCLIB Member Updates

PROCLIB Member	**Action to Take**	**Element Name**
For BCP element		

Figure 42 (Page 2 of 4). PROCLIB Member Updates

PROCLIB Member	**Action to Take**	**Element Name**
APPC ASCH ASCHINT AVM AXRNN AXRPSTRT BLSJIPCS BLSJPRMI BPXAS BPXOINIT CEA DEALLOC DLF DUMPSRV GLZ GTF GTZ HIS HWISTART HZR HZSPROC IEESYSAS IEEVMPCR INIT IXGLOGRS LLA OMVS RDR RDR3200 RDR400 RMTGEN VLF XWTR	Ensure these procedures are correctly customized for your environment. They are provided in your PROCLIB data set.	BCP
For DFSMS element		

Figure 42 (Page 3 of 4). PROCLIB Member Updates

PROCLIB Member	Action to Take	Element Name
CDRAINIT CVAFTR DEVMAN DFM DFMX0001 DFM00 GDEAPDEF GDEAPPC GDEASCH GDELOGMD GDEPRTLU GDETPDEF IDAVDT IFGEDI IFGOCETR LINKS LOGROUTE	Ensure these procedures are correctly customized for your environment. They are provided in your PROCLIB data set.	DFSMS
For HCD element		
CBDJCMPR CBDJIMPT CBDJIOCP CBDJRPTS CBDJXMIT CBDQAJSK CBDQDISP	Ensure these procedures are correctly customized for your environment. They are provided in your PROCLIB data set.	HCD
For Others		
BDT$V2SP BDT$V2TP	Copy these procedures and update for your environment. BDT$V2SP is the started procedure for BDT. BDT$V2TP is the started procedure for the BDT transaction queuing integrity (TQI) address space. Both of these procedures are provided in your SBDTSAMP library.	BDT
CFZCIM	Procedure is provided in your PROCLIB data set. Update for your environment.	Common Information Model (CIM)
EZAZSSI	A sample member is provided in your SEZAINST library.	Communications Server IP Services
ICEOPT	The procedure is provided in your PROCLIB data set.	DFSORT
IOEP0004	These are SIOEPROC library members. The alias ZFS is provided to refer to IOEP0004. ZFS is the sample procedure for running ZFS. Ensure the alias is maintained and reflect local updates to these sample procedures.	z/OS File System

Figure 42 (Page 4 of 4). PROCLIB Member Updates

PROCLIB Member	Action to Take	Element Name
EPWFFST	Copy this procedure and update for your environment. Sample member is provided in your SEPWSRC2 library.	FFST
ASMAC ASMACL ASMACLG ASMACG	Change these procedures for your environment. Sample members are provided in your SASMSAM1 library. If the previous release of High Level Assembler has been used, users might have become accustomed to using the old procedure names of HLASMC, HLASMCG, HLASMCLG and HLASMCL. To make the transition from the previous release, in your SASMSAM1 library, these old procedure names appear as aliases for the sample supplied procedures.	High Level Assembler
ANFWPROC AOPDEMON AOPPRINT AOPSTART AOPSTAR2 AOPSTAT AOPSTOP AOPSTOP2 APIJPJCL	Procedures are provided in your PROCLIB data set. Update your environment, as described in *z/OS Infoprint Server Customization*.	Infoprint Server
DSSRV	DSSRV is an alias in the SGLDSAMP library.	IBM TDS Server
IZUANG1 IZUFPROC IZUINSTP IZUSVR1	Procedures are provided in your PROCLIB data set.	IBM z/OS Management Facility (z/OSMF)
GFSAPROC GFSCPROC	Update for your environment. These members are provided in your SAMPLIB ibrary. GFSAPROC is a sample MVSNFS procedure, GFSCPROC is a sample MVSNFSC procedure.	Network File System
RACF	This supplied procedure may be updated. See member RACPROC in SAMPLIB.	RACF
RMFM3B GPMSERVE GPM4CIM	These supplied procedures may be updated, if you would like. See the comments provided in this member of your PROCLIB data set.	RMF
SDSF	Copy members ISF.SISFJCL(SDSF) and ISF.SISFJCL(SDSFAUX) to PROCLIB. As of V2R3, the SDSF and SDSFAUX address spaces are required when running SDSF.	SDSF
RMF RMFGAT	These supplied procedures may be updated, if you would like. See the comments provided in this member of your PROCLIB data set.	z/OS Data Gatherer

8.5.3.2 Ensuring that entire libraries are accessible: The following procedure libraries, which are shown in ripple order, must either be copied to SYS1.PROCLIB or be added to a PROCLIB concatenation accessible to JES. Customization considerations for these data sets are detailed after this figure.

Figure 43. PROCLIB Concatenation

DDNAME	Element	Note
SIOEPROC	z/OS File System	Contains alias ZFS (see Figure 42 on page 145); ensure the alias is maintained.
SGLDSAMP	IBM Tivoli Directory Server for z/OS	This data set contains various LDAP procedures and sample jobs.
SCEEPROC	Language Environment	
SCCNPRC	XL C/C++	

8.5.3.3 Customize procedures for XL C/C++: You may customize the procedures provided by C/C++. For more information, see *z/OS XL C/C++ User's Guide*.

8.5.3.4 Customize Language Environment procedures: You may customize the procedures provided by Language Environment. For more information, see *z/OS Language Environment Customization*.

8.5.4 z/OS V2R5 ISPF setup considerations

Figure 44 on page 149 shows the data sets that must be concatenated to ISPF DDNAMEs in the logon procedure to use z/OS V2R5 functions. See Figure 45 on page 154 for Logon Proc using language libraries.

Figure 44 (Page 1 of 6). Logon PROC Updates

DDNAME	DDDEF	ELEMENT
ISPLLIB	DFQLLIB DGTLLIB	DFSMS
	SCBDHENU	HCD
	SGIMLMD0	SMP/E

Note:

- To avoid needing SYS1.NUCLEUS in your ISPLLIB, use the parameter UIM_LIBNAME in the HCD profile to point to the data set containing the UIMs and UDTs. If the keyword is omitted, SYS1.NUCLEUS is assumed. If UIM_LIBNAME=* is specified, HCD assumes the UIM data sets are part of the ISPLLIB or STEPLIB DDNAME.
- Data sets in the LNKLST*xx* or the LPALST*xx* should not be included in the ISPLLIB or STEPLIB DDNAME.

Figure 44 (Page 2 of 6). Logon PROC Updates

DDNAME	**DDDEF**	**ELEMENT**
ISPMLIB	SBLSMSG0 SBPXMENU	BCP
	SBDTMSG	BDT
	DGTMLIB DFQMLIB SEDGMENU	DFSMS
	SADMMSG	GDDM
	SCBDMENU	HCD
	SEZAMENU	Communications Server IP Services
	SCSFMSG0	Cryptographic Services ICSF
	SAOPMENU	Infoprint Server
	SISPMENU	ISPF
	SIATMSG0	JES3
	SERBMENU	RMF
	SISFMLIB	SDSF
	HRFMSG	Security Server (RACF)
	SGIMMENU	SMP/E
	ICQMLIB	TSO/E

Figure 44 (Page 3 of 6). Logon PROC Updates

DDNAME	DDDEF	ELEMENT
ISPPLIB	SBLSPNL0 SBPXPENU	BCP BCP (z/OS UNIX)
	SBDTPNL0	BDT
	DFQPLIB DGTPLIB SEDGPENU	DFSMS
	SEPWPENU	FFST
	SADMPNL	GDDM
	SCBDPENU	HCD
	SEZAPENU	Communications Server IP Services
	SCSFPNL0	Cryptographic Services ICSF
	SAOPPENU	Infoprint Server
	SISPPENU	ISPF
	SHASPNL0	JES2
	SIATPNL0	JES3
	SERBPENU	RMF
	SISFPLIB	SDSF
	HRFPANL	Security Server (RACF)
	SGIMPENU	SMP/E
	ICQPLIB	TSO/E
ISPSLIB	SBLSKEL0	BCP
	DGTSLIB	DFSMS
	SCSFSKL0	Cryptographic Services ICSF
	SISPSLIB	ISPF
	SISPSENU	ISPF
	SISFSLIB	SDSF
	HRFSKEL	Security Server (RACF)
	SGIMSENU	SMP/E
	ICQSLIB	TSO/E

<table>
<tr><td colspan="3"><i>Figure 44 (Page 4 of 6). Logon PROC Updates</i></td></tr>
<tr><th>DDNAME</th><th>DDDEF</th><th>ELEMENT</th></tr>
<tr><td rowspan="11">ISPTLIB</td><td>SBLSTBL0
SBPXTENU</td><td>BCP
BCP (z/OS UNIX)</td></tr>
<tr><td>DGTTLIB</td><td>DFSMS</td></tr>
<tr><td>SCBDTENU</td><td>HCD</td></tr>
<tr><td>SCSFTLIB</td><td>Cryptographic Services ICSF</td></tr>
<tr><td>SEZAPENU</td><td>Communications Server IP Services</td></tr>
<tr><td>SISPTENU</td><td>ISPF</td></tr>
<tr><td>SIATTBL0</td><td>JES3</td></tr>
<tr><td>SERBT
SERBTENU</td><td>RMF</td></tr>
<tr><td>SISFTLIB</td><td>SDSF</td></tr>
<tr><td>SGIMTENU
SMPTABL</td><td>SMP/E</td></tr>
<tr><td>ICQTLIB</td><td>TSO/E</td></tr>
<tr><td colspan="3"><b>Note:</b> SMPTABL is required for SMP/E in the ISPF concatenation, but does not have a DDDEF. For more information on how to allocate the SMPTABL, see 8.5.4.1, “SMP/E customization considerations for ISPF” on page 156.</td></tr>
<tr><td rowspan="9">SYSEXEC</td><td>SBPXEXEC</td><td>BCP (z/OS UNIX)</td></tr>
<tr><td>SEZAEXEC</td><td>Communications Server</td></tr>
<tr><td>SEDGEXE1</td><td>DFSMS</td></tr>
<tr><td>SIOEEXEC</td><td>z/OS File System</td></tr>
<tr><td>SEPWSRC1</td><td>FFST</td></tr>
<tr><td>SAOPEXEC</td><td>Infoprint Server</td></tr>
<tr><td>SISPEXEC</td><td>ISPF</td></tr>
<tr><td>SISFEXEC</td><td>SDSF</td></tr>
<tr><td>SGLDEXEC</td><td>IBM TDS Server</td></tr>
<tr><td colspan="3"><b>Notes:</b>
1. Ensure all libraries in the SYSEXEC concatenation have the same record format.
2. The libraries in SYSEXEC could be put in the SYSPROC DD instead.</td></tr>
</table>

Figure 44 (Page 5 of 6). Logon PROC Updates

DDNAME	DDDEF	ELEMENT
SYSHELP	HELP	BCP, DFSMS, z/OS UNIX System Services, RMF, TSO/E, Communications Server IP
	SBDTHELP	BDT
	SISPHELP	ISPF
	SISFHELP	SDSF
	HELPENP	TSO/E

Notes:

1. z/OS V2R4 is the last release to support the RACF TSO help commands. Beginning in z/OS V2R5, RACF HELP panels will no longer be shipped.

DDNAME	DDDEF	ELEMENT
SYSPROC	SBLSCLI0	BCP
	SBDTCLI0	BDT
	SCCNUTL	XL C/C++
	DGTCLIB	DFSMS
	SEPWCENU	FFST
	SCBDCLST	HCD
	SCSFCLI0	Integrated Cryptographic Service Facility
	SISPCLIB	ISPF
	SERBCLS	RMF
	SGRBCLS	z/OS Data Gatherer
	HRFCLST	Security Server (RACF)
	SGIMCLS0	SMP/E

Notes:

1. Ensure all libraries in the SYSPROC concatenation have the same record format.
2. EOY2GDF, EOY2PSEG, and EOYPARM members of SEOYCLIB should be customized for the customer site.
3. Data Sets SCCNUTL contain only REXX EXECs.

Figure 44 (Page 6 of 6). Logon PROC Updates

DDNAME	DDDEF	ELEMENT
STEPLIB or LNKLST*xx*	SBDTLIB	BDT
	SEPWMOD4	FFST
	SCBDHENU	HCD

Note: If the following libraries are removed from LNKLST*xx*, add them to the STEPLIB concatenation and authorize every library in the concatenation using PROG*xx* or IEAAPF*xx*.

- SISPLOAD
- SCEERUN
- SCLBDLL
- SCLBDLL2
- SCCNCMP

The logon PROC updates for the Language libraries are listed below. Add appropriate Language libraries to your logon PROC. Unless the English library is specified as required in the table, you can remove the English library after you add the Language libraries.

Figure 45 (Page 1 of 2). Logon PROC Updates for Language libraries

DDNAME	DDDEF	ELEMENT
ISPMLIB	SBPXMJPN	BCP Japanese
	DFQMKLB DGTMKLB SEDGMJPN	DFSMS Japanese
	SCBDMJPN	HCD Japanese
	SAOPMJPN	Infoprint Server Japanese
	SISPMENP SISPMJPN	ISPF Upper Case English ISPF Japanese
	SCEEMSGP	Language Environment Upper Case English
	SERBMJPN	RMF Japanese
	SICHMJPN	Security Server (RACF) Japanese
	SGIMMJPN	SMP/E Japanese
	ICQKMLIB	TSO/E Japanese

Figure 45 (Page 2 of 2). Logon PROC Updates for Language libraries

DDNAME	DDDEF	ELEMENT
ISPPLIB	SBPXPJPN	BCP Japanese
	DFQPKLB DGTPKLB SEDGPJPN	DFSMS Japanese
	SCBDPJPN	HCD Japanese
	SAOPPJPN	Infoprint Server Japanese
	SISPPENP SISPPJPN	ISPF Upper Case English ISPF Japanese
	SERBPJPN	RMF Japanese
	SICHPJPN	Security Server (RACF) Japanese
	SGIMPJPN	SMP/E Japanese
	ICQKPLIB	TSO/E Japanese
ISPSLIB	DGTSKLB	DFSMS Japanese
	SISPSENP SISPSJPN	ISPF Upper Case English ISPF Japanese
ISPTLIB	SBPXTJPN	BCP Japanese
	SCBDTJPN	HCD Japanese
	SISPTENP SISPTJPN	ISPF Upper Case English ISPF Japanese
	SERBTJPN	RMF Japanese
	SGIMTJPN SGIMTENU	SMP/E Japanese SMP/E English
	ICQKTABL	TSO/E Japanese
Note: Both the Japanese and English SMP/E libraries are required in the ISPTLIB concatenation.		
SYSEXEC	ICQKCLIB	TSO/E Japanese
SYSHELP	KHELP	BCP (z/OS UNIX), TSO/E, z/OS UNIX System Services Japanese
	HLPKLB	DFSMS Japanese

8.5.4.1 SMP/E customization considerations for ISPF

Use the ISPCTL1 and ISPCTL2 files to generate JCL for submitted SMP/E jobs. The SMP/E job submit facility lets you browse and edit this JCL. You can omit these files from your logon procedure and let ISPF automatically allocate them as needed. To save the input JCL generated by the dialogs, allocate a permanent sequential data set to ISPCTL1 (LRECL=80, RECFM=FB) before you enter the SMP/E dialogs or just use EDIT CREATE.

Allocate a single, installation-wide table data set to the ISPTLIB and SMPTABL DD statements. SMP/E uses this table data set to save process status information for the SYSMOD management dialogs. The data set must be a partitioned data set (LRECL=80, RECFM=FB). Because the data set is also in the concatenation of ISPTLIB, make the block size compatible with the block size of the corresponding ISPF data sets.

8.5.4.2 Update ISPF Command Table ISPTCM (if Required): For more information and instructions about updating ISPTCM, see *z/OS ISPF Planning and Customizing*.

Note: ISPTCM was updated to remove MVSSERV by ISPF APAR OA58883 in z/OS V2R4.

8.5.5 Element customization

After the elements are installed, you should examine (and if necessary, change) any installation exits and modifications to meet your installation's needs.

8.5.5.1 RACF security considerations

To perform the customization procedures for several of the z/OS V2R5 Wave 1 elements, you must have certain security permissions available. You may set up the security for all of the Wave 1 elements at this time. The following list shows elements that require RACF security and the location of more information.

- DFSMS (see *z/OS DFSMSdfp Storage Administration*)
- FFST (see 8.5.5.5.3, “FFST customization considerations” on page 169)
- HCD (see *z/OS HCD User's Guide*)
- HCM (see *z/OS and z/VM HCM User's Guide*)
- Communications Server IP Services (see 4 on page 164)
- Cryptographic Services ICSF (see *z/OS Cryptographic Services ICSF Administrator's Guide* and *z/OS Cryptographic Services ICSF System Programmer's Guide*)
- RMF (see 8.5.5.7.3, “RMF customization considerations” on page 173)
- Security Server (see 8.5.5.6.2, “Security Server (RACF) customization considerations” on page 171)
- IBM Tivoli Directory Server for z/OS (see 8.5.5.3.4, “IBM Tivoli Directory Server for z/OS customization considerations” on page 158)
- TSO/E (see *z/OS TSO/E Customization*)
- z/OS UNIX System Services (see 8.5.5.2, “z/OS UNIX System Services customization considerations” on page 157)
- z/OS UNIX System Services Application Services (see 8.5.5.4.2, “z/OS UNIX System Services Application Services customization” on page 168)

- IBM Knowledge Center for z/OS (see 8.5.5.8.4, “IBM Knowledge Center for z/OS customization considerations” on page 178)
- z/OS File System (see 8.5.5.9.1, “RACF updates for zFS” on page 179)
- Infoprint Server (see 8.5.5.9.2.4, “Establish security for Infoprint Server” on page 180)
- Network File System (see *z/OS Network File System Guide and Reference*)
- IBM z/OS Management Facility (see *IBM z/OS Management Facility Configuration Guide*)

Note: The SDSF element, which is installed in Wave 2, requires certain security permissions to be defined when SDSF is used in a JES3 environment.

8.5.5.2 z/OS UNIX System Services customization considerations

Note: This section must be completed before the Communications Server IP Services customization section can be performed.

A SAMPLIB member is provided with z/OS UNIX System Services Kernel, BPXISEC1, which is a sample TSO CLIST that provides all the RACF commands needed for the security setup of z/OS UNIX System Services. This CLIST is being provided as a central location for these commands and should be used by customers in conjunction with *z/OS UNIX System Services Planning* to reference detailed information. You may also need to refer to:

- *z/OS Security Server RACF Security Administrator's Guide*
- *z/OS Security Server RACF System Programmer's Guide*
- *z/OS Security Server RACF Command Language Reference*

This CLIST is structured into separate sections, each with a set of block comments, describing the RACF function that will be setup and what needs to be changed by the customer. **Ensure you execute this CLIST from a RACF special user ID!**

8.5.5.3 Customization considerations for Wave 1A

The following sections contain additional customization considerations for Wave 1A.

8.5.5.3.1 Integrated Security Services Network Authentication Service customization considerations

The following customization actions should be done at this time:

- Catalog all of the EUVF.SEUVF*xxx*, where *xxx* is LIB (Library) and EXC (REXX execs for the TSO logon procedure).
- Copy the SKRBKDC started task proc in EUVF.SEUVFSAM to SYS1.PROCLIB
- Copy the SKRBWTR started task proc in EUVF.SEUVFSAM to SYS1.PROCLIB or to a proclib in the IEFPDSI DD name in your MSTJCLxx member in your PARMLIB concatenation.
- Customize the SKRBWTR procedure according to your system installation needs.
- Copy the appropriate CTIKDCxx member from EUVF.SEUVFSAM to your installation's PARMLIB member and customize as necessary to support the component's tracing that you may need to perform.

Detailed steps to get the program into operational status are defined in *z/OS Integrated Security Services Network Authentication Service Administration*.

8.5.5.3.2 Language Environment customization considerations

Refer to *z/OS Language Environment Customization* for more information and instructions on Language Environment customization. You can customize any of the following items for Language Environment:

- Default run-time options
- User exits:
  - Assembler user exit
  - High-level language user exit
  - Abnormal termination exit
- CLISTs
- Place Language Environment modules in the link pack area
- COBOL COBPACKs
- Fortran LIBPACKs
- Make Language Environment available under CICS (optional)
  - Add CICS PPT entries
  - Add DCT entries
  - Add Language Environment-CICS data sets to CICS startup job stream
- Language-specific features:
  - OS/VS COBOL Compatibility Library
  - COBOL parameter list exit
  - COBOL reusable environment
  - C/C++ Base locale time information
  - Improving performance on IMS/DC
  - Fortran unit attribute table
  - VS Fortran compatibility unit attribute table
  - VS Fortran compatibility run-time options
  - VS Fortran compatibility error option table

8.5.5.3.3 Cryptographic Services ICSF customization considerations

Refer to *z/OS Cryptographic Services ICSF Administrator's Guide* and *z/OS Cryptographic Services ICSF System Programmer's Guide* for more information on Cryptographic Services ICSF customization.

8.5.5.3.4 IBM Tivoli Directory Server for z/OS customization considerations

Refer to the *z/OS IBM Tivoli Directory Server Administration and Use for z/OS*.

8.5.5.3.5 IBM Generic Tracker for z/OS customization considerations

Note that this list of available samples is not meant to be a list of required customization steps when installing the current release. Instead this list is meant to be a simple inventory of potential tools available to execute typical tracking facility tasks or to learn more about the tracking facility functions.

GTZCNIDJ creates GTZPRM*xx* parmlib members from existing CNIDTR*xx* parmlib members.
GTZPRNTJ prints data and associated information currently stored in the tracking facility.

GTZSMFJ Sample JCL for formatting SMF records of record type 125 in text form.
GTZSHCK a sample local health check in METAL C to report tracked events
GTZSHCKJ builds the GTZSHCK sample health check
GTZSHCKX a sample REXX health check to report tracked events

Refer to the samples' prologue and *z/OS MVS Diagnosis: Tools and Service Aids* for more information and instructions about IBM Generic Tracker customization.

8.5.5.3.6 IBM Health Checker for z/OS customization considerations

Note that only HZSALLCP in the list below is recommended to be used to complete your IBM Health Checker for z/OS setup. The remaining list is meant to be a simple inventory of potential tools to execute typical tasks or to learn more about the IBM Health Checker for z/OS functions. All these items can be found in SYS1.SAMPLIB.

HZSALLCP allocates the persistent data dataset. This sample must be updated so HZSPROC and this sample reflect the same dataset.
HZSPRINT prints the current, or logstream copies of message buffers
HZSMSGNJ generates a health check's message table and associated message files
HZSSADCK a sample HZSADDCHECK dynamic exit routine
HZSSCHKP a sample local health check with use of persistent data services HZSPREAD and HZSPWRIT
HZSSCHKR a sample local health check with use of check message service HZSFMSG
HZSSMSGT a sample message definition file as input to message table generating tool HZSMSGNJ
HZSSRCHC a sample remote health check with use of check parameter parsing service HZSCPARS
HZSSRCHK a sample remote health check with use of check message service HZSFMSG and format MGBFORMAT=1
HZSSSYMD a sample symbol definition file to be used as additional input to message table generating tool HZSMSGNJ
HZSSXCHK a sample REXX health check with use of a REXXIN dataset
HZSSXCHN a sample REXX health check with use of check message service HZSLFMSG and REQUEST=DIRECTMSG

Refer to *IBM Health Checker for z/OS User's Guide* for more information and instructions about IBM Health Checker customization.

8.5.5.3.7 ISPF customization considerations

Refer to *z/OS ISPF Planning and Customizing* for more detailed information on customizing ISPF.

8.5.5.3.7.1 z/OS V2R5 Sample panels

Sample ISPF panels are provided to enable panels for most z/OS V2R5 elements. These panels reside in the SISPPENU data set after APPLY processing. Two sample panels are supplied:

ISR@390S This sample panel is distributed with options used by system programmers and administrators. It includes options for:

- GDDM Print Queue Manager

- HCD I/O configuration
- APPC Administration Dialog
- Work Load Manager
- FFST dump formatting
- Infoprint Server
- RMF
- SMP/E
- TCP/IP NPF

ISR@390U This sample menu panel includes the options used by most ISPF users. It includes options for:

- DFSMSrmm/ISMF
- DFSMSdfp/ISMF
- BDT File-to-File Feature
- IPCS for dump reading
- z/OS Unix Browse Files
- z/OS Unix Edit Files
- z/OS Unix Shell
- Security Server
- TSO/E Information Center Facility
- SDSF

Options for these elements have not been included in the samples:

- Communications Server for z/OS SNA Services IPCS Trace

  The panels for this element require extensive setup. Refer to "Installing the VTAM® Dump Analysis Enhancements and the VIT Analysis Tool" in *z/OS Communications Server: New Function Summary* for information about enabling these panels.

ISPF customization for RACF panels

If your installation uses the TSO/E session manager, and you want it to capture the output of RACF panels, specify PANEL(ICHP00SM) instead of PANEL(ICHP00), in ISR@390U. For example, change:

```
12,'PANEL(ICHP00) SCRNAME(SECURITY)'
```

to

```
12,'PANEL(ICHP00SM) SCRNAME(SECURITY)'
```

8.5.5.3.8 Communications Server IP Services customization considerations

Perform the following steps to complete the installation and initial customization of the Communications Server IP Services address space and key servers. At the completion of these steps, the Communications Server will have been set up for a basic configuration with no network attachments. **Additional customization will be required** to link the host to the network, bring up additional services, and optimize the use of initially configured services.

By choosing appropriate data set names and procedure names, these configuration steps can be done without interfering with any existing IP configuration. However, some customers may choose to simply review these steps and instead do IP configuration according to procedures they already have in place.

Refer to *z/OS Communications Server: New Function Summary* and *z/OS Communications Server: IP Configuration Guide* for more detailed customization instructions.

Before running the verification procedure you must have a fully functional z/OS UNIX environment and a fully functional security environment, either RACF or some other security program.

The following steps must be run from a TSO user ID with an OMVS segment defined. The user ID must be uid=0 or permitted to access the 'BPX.SUPERUSER' resource in the FACILITY class. The user ID must also be able to create and write to the required TSO datasets and hierarchical file system files and to perform RACF definitions. Steps in the EZARACFI job provide guidance for creating the entries that are required to complete the Communications Server IP IVP.

The PROCLIB used to contain the IVP procedures (*IVP_proclib_dsn*) must be in the PROCLIB concatenation. Before executing the steps, the following values must be known:

- ***target_lib_hlq***: high level qualifier of the TCP/IP target libraries (SEZAxxxx data sets). This will have been selected when the HLQ variable was chosen for the EZACSALC job run earlier in the Program Directory steps.
- ***dyn_alc_hlq***: high level qualifier of dynamically allocated TCP/IP data sets. Default is TCPIP, but we recommend use of TCPIVP.
- ***Default_RACF_group***: RACF group to which TCPIP and server procedures will be permitted. Default is OMVSGRP, which is the value used by the z/OS UNIX IVP. If you change the RACF Group for the z/OS UNIX IVP, it will need to be changed here as well. This group must be defined before the instructions below are executed.
- ***TCPIP_PROFILE_name***: data set name for the TCPIP.PROFILE. Defaults to TCPIP.PROFILE.TCPIP, but we recommend use of 'TCPIVP.TCPPARMS(PROFILE)'. For the purpose of these installation verification procedures, this must be an MVS data set, not a ZFS file.
- ***TCPIP_DATA_name***: data set name for the TCPIP.DATA data set. Defaults to TCPIP.TCPIP.DATA, but we recommend use of 'TCPIVP.TCPPARMS(TCPDATA)'. For the purpose of these installation verification procedures, this must be in an MVS data set, not a ZFS file.
- ***TCPIP_procname***: procedure name chosen for the TCP/IP address space; defaults to TCPIP.
- ***SYSLOGD_procname***: procedure name for syslog daemon; defaults to SYSLOGD
- ***FTPD_procname***: procedure name for FTP daemon; defaults to FTPD.
- ***IVP_proclib_dsn***: data set name of the IVP-specific procedure library.
- ***IVP_jcllib_dsn***: data set name for batch jobs used for installation and verification.
- ***IVP_configuration_dsn***: name of partitioned dataset to be used to contain configuration files for executing the IVP. We recommend use of 'TCPIVP.TCPPARMS'
- ***hostname***: The name of the system as it will be known in the TCP network.

Note: To avoid accidently overlaying an existing system environment, a default is not used. The preceding environment settings are meant to be discarded and not used in a live network.

When executing the following steps, you must use the same value for each of the *highlighted variables* (for example, *dyn_alc_hlq*) for each step where that variable is required.

Also verify:

- the Communications Server IP Services datasets have been added to the appropriate LNKLSTxx and LPALSTxx members of PARMLIB.
- BPXPRMxx parmlib member in use by z/OS UNIX has been updated for Communications Server IP Services.
  - Check there is a FILESYSTYPE statement specifying entry point EZBPFINI, as shown in the following example.

    ```
    FILESYSTYPE TYPE(INET) ENTRYPOINT(EZBPFINI)
    NETWORK DOMAINNAME(AF_INET)
    DOMAINNUMBER(2)
    MAXSOCKETS(60000)
    ```

  - If any changes are required, the new OMVS parameters may be updated by using the SETOMVS RESET=(yy) command or by re-IPLing.

Customization steps

1. Allocate required data sets
   - Customize the EZAGETIN job
     - This job allocates required TCPIP datasets, such as translation tables, HOSTS.LOCAL, ETC.SERVICES, etc
     - Copy the sample job to a JCL data set of your choice: *target_lib_hlq*.SEZAINST(EZAGETIN) to *IVP_jcllib_dsn*(EZAGETIN)
     - Fill in any required parameters on the //JOB statement
     - Set the DSPREFIX variable on the //PROC statement to *dyn_alc_hlq* or accept the default of TCPIP.
     - Set the TGTPRFX variable to *target_lib_hlq* or accept the default of TCPIP.
     - Set variables on the //PROC statement for the DASD volume serial and type where the datasets will be allocated
   - Run the EZAGETIN job.
     - Expected results: The job should complete with all steps receiving a condition code 0.
   - Update the *dyn_alc_hlq*HOSTS.LOCAL file
     - Copy an existing HOST entry, change the IP address to the loopback address and fill in the appropriate *hostname* value; for example:

       ```
       HOST : 127.0.0.1   :hostname ::::
       ```

     - Comment out the additional sample HOST, NET, and GATEWAY statements.
   - Run makesite to create the hlq.HOSTS.ADDRINFO and hlq.HOSTS.SITEINFO datasets
     - From TSO, issue the following command:

       ```
       makesite hlq=dyn_alc_hlq
       ```

     - Expected results:
       - The following messages should be seen at the TSO session:

```
EZA0547I Writing out dyn_alc_hlq.HOSTS.SITEINFO ...
EZA0548I Writing out dyn_alc_hlq.HOSTS.ADDRINFO ...
```

- Messages EZA0550I through EZA0566I will also be generated showing statistics from the input data set read and the output data sets created.

2. Allocate the partitioned dataset for IVP configuration

   - If not already allocated, *IVP_configuration_dsn* must be allocated now. To do so, copy the allocation job: *target_lib_hlq*.SEZAINST(ALOCPARM) to *IVP_jcllib_dsn*(ALOCPARM).
   - Fill in any required parameters on the //JOB statement.
   - Change the //ALLOCT EXEC statement to set the DSN variable to *IVP_configuration_dsn*, or accept the default of 'TCPIVP.TCPPARMS'.
   - Change the //ALLOCT EXEC statement to set the DSP variable to the appropriate final disposition of the dataset, or accept the default of 'CATLG'.
   - Change the //ALLOCT EXEC statement to set the TVOL1 variable to the volume serial of the volume on which you wish to allocate the data set.
   - Submit the ALOCPARM job.
     - Expected results: The job should complete with a condition code 0.

3. Customize the TCP/IP address space

   - Customize the procedure for starting the TCP/IP address space:
     - Copy the sample start procedure from the SEZAINST library to the appropriate proclib: *target_lib_hlq*.SEZAINST(TCPIPROC) to *IVP_proclib_dsn(TCPIP_procname)*
     - Make changes necessary to fit your environment
       - If desired, change the default jobname to the chosen procedure name on the //PROC statement: TCPIP to *TCPIP_procname*

         Note: If the system is configured for common INET (CINET is specified in the BPXPRMxx member of SYS1.PARMLIB), the name chosen for the TCP/IP start procedure must match the one used for the SUBFILESYSTYPE NAME value. For this IVP, however, common INET will not be used. See *z/OS UNIX System Services Planning* for details.
       - Set up the //PROFILE DD card to point to the data set to be used to contain the TCPIP.PROFILE.
         - Either uncomment one of the provided statements (to use either TCPIP.PROFILE.TCPIP or TCPIVP.TCPPARMS(PROFILE) as the value of *TCPIP_PROFILE_name*) **-or-**
         - Copy one of the provided //PROFILE DD statements, uncomment it, and set DSN equal to *TCPIP_PROFILE_name*
       - If desired, change the default TCPIP.DATA name to the selected *TCPIP_DATA_name*
         - Set up the //SYSTCPD DD card to point to data set to be used to contain the TCPIP.DATA
           - Either uncomment one of the provided statements (to use either TCPIP.SEZAINST(TCPDATA) or the recommended value of TCPIVP.TCPPARMS(TCPDATA) as the value of *TCPIP_DATA_name*) **-or-**
           - Copy one of the provided //SYSTCPD DD statements, uncomment it, and set DSN equal to *TCPIP_DATA_name*

  - Comment out the PARM= card of the //EXEC statement. Uncomment the SET statement on the line above the EXEC statement and specify the appropriate dsname for the TCPIP.DATA data set. Uncomment the two lines for the PARM, following the EXEC statement, that specifies the setting for the RESOLVER_CONFIG environment variable.

    ```
    //* SET PARM1=TCPIVP.TCPPARMS(TCPDATA) ...
    //* PARM=('&PARMS',
    //* 'ENVAR("RESOLVER_CONFIG=//''&PARM1''")')
    ```

  - If not using 'TCPIVP.TCPPARMS(TCPDATA)' for *TCPIP_DATA_name*, change the ENVAR setting on the //EXEC statement to use *TCPIP_DATA_name*.

- Customize the TCPIP profile data set
  - – Copy the sample TCPIP profile dataset from the SEZAINST library to the selected location: *target_lib_hlq*.SEZAINST(SAMPPROF) to *TCPIP_PROFILE_name*

    Note: If *TCPIP_PROFILE_name* is not an existing data set, it will need to be allocated.
  - – If not using the default value for *FTPD_procname*, change the PORT statement that specifies FTPD1 to the chosen *FTPD_procname*1 (be sure to add the 1 at the end).
- Customize the TCPIP.DATA data set
  - – Copy the sample TCPIP.DATA dataset from the SEZAINST library to the selected location: *target_lib_hlq*.SEZAINST(TCPDATA) to *TCPIP_DATA_name*.

    Note: If *TCPIP_DATA_name* is not an existing data set, it will need to be allocated.
  - – Set TCPIPJOBNAME to the value of *TCPIP_procname*.
  - – Set HOSTNAME statement to the *hostname* of the system as it will be known in the TCP network.
  - – Set DOMAINORIGIN statement to the domain for this host or comment out the statement.
  - – Set DATASETPREFIX to the value of *dyn_alc_hlq*.

4. Do required RACF definitions

- Customize the EZARACFI job
  - – This job will perform the RACF definitions and define OMVS segments required for the TCP/IP address and servers.
  - – Copy the sample job to a JCL data set of your choice: *target_lib_hlq*.SEZAINST(EZARACFI) to *IVP_jcllib_dsn*(EZARACFI).
  - – Fill in any required parameters on the //JOB statement.
  - – If not using the default value for *Default_RACF_group*, change the DFLTGRP(OMVSGRP) definitions on the ADDUSER statements for the DAEMONS step to use the selected value for *Default_RACF_group*. (Note, it is assumed this group has already been created.).
  - – If desired, comment out ADDUSER and RDEFINE statements for any servers for which you do not want to do RACF definitions at this time. Note, however, the step will be terminated by the first comment statement, so move any comments to the end of the //SYSTSIN input.
  - – For this IVP, RACF definitions are required for the TCPIP, SYSLOGD, and FTPD started procedures.
  - – If not using the default value for *TCPIP_procname*, change 'TCPIP' on the ADDUSER and RDEFINE statements to the selected value for *TCPIP_procname*.

  - If not using the default value for *SYSLOGD_procname*, change 'SYSLOGD' on the ADDUSER and RDEFINE statements to the selected value for *SYSLOGD_procname*.
  - If not using the default value for *FTPD_procname*, change 'FTPD' on the ADDUSER and RDEFINE statements to the selected value for *FTPD_procname*.

    Note: For the purposes of the IVP, the preceding RACF definitions are the only ones used. You may, however, wish to uncomment additional RACF definitions for other applications that are not part of the IVP in the EZARACF sample.
  - Additional steps are required when running with BPX.DAEMON set up (If you are not running with BPX.DAEMON, skip to Step 4.)
    - If you have a BPX.DAEMON environment, you must ensure all load modules loaded into an address space come from controlled libraries. If a module is loaded from a noncontrolled library, the address space loses its authorization. This means you must define all the libraries from where modules may be loaded as program controlled. See *z/OS UNIX System Services Planning* for more information on the BPX.DAEMON resource in the FACILITY class. The following steps, found in EZARACF, can be used for including z/OS IP in an existing BPX.DAEMON environment.
    - Permit servers to the BPX.DAEMON resource.(EZARACF)
      - Uncomment the //BPXDAEMN EXEC statement and the //SYSTSPRT DD and //SYSTSIN DD cards associated with it.
      - Uncomment (remove the //*) RACF statements to permit (PE) the TCP/IP address space and the FTP daemon to a resource in the FACILITY class.
      - If not using the default value for *TCPIP_procname* change 'TCPIP' on the PE statement to the selected value for *TCPIP_procname*.
      - If not using the default value for *FTPD_procname* change 'FTPD' on the PE statement to the selected value for *FTPD_procname*.
    - Enable program control for runtime library modules
      - Uncomment the //PADS EXEC statement and the //SYSTSPRT and //SYSTSIN DD cards associated with it.
      - Uncomment (remove the //*) RALT statements for 'SYS1.LINKLIB', 'cee.SCEERUN', and 'tcpip.SEZALOAD'.
      - Change the high level qualifier for the C run-time library from 'cee' to the appropriate high level qualifier.
      - Change the 'volser' value on the RALT statement to the volume serial for the C run-time library.
      - Change the high level qualifier for the TCP/IP load libraries from 'tcpip' to *target_lib_hlq*.
      - Change the 'volser' value on the RALT statement for the TCP/IP load library to the volume serial for the TCP/IP load library.
      - Uncomment the SETROPTS statement and move it to immediately after the last uncommented RALT statement.
- Run the EZARACFI job
  - Expected results: The job completes with condition code 0 and no error messages.
    - Note, if the users are already defined to RACF, error messages for 'invalid user ID' are generated. Successful ADDUSER definitions produce only the TSO 'READY' prompt in the output.

- The condition code returned is the condition code for the last RACF command issued, so read the job output to verify the RACF definitions were done.

5. Customize SYSLOGD

   - Copy the sample start procedure from the SEZAINST library to the appropriate PROCLIB: *target_lib_hlq*.SEZAINST(SYSLOGD) to *IVP_proclib_dsn(SYSLOGD_procname)*.
   - If desired, change the default jobname to the chosen procedure name on the //PROC statement: SYSLOGD to *SYSLOGD_procname*.
   - Comment out the //CONFHFS EXEC statement and uncomment the //CONFPDS EXEC statement.
   - On the //CONFPDS EXEC statement, either accept the default location for the configuration file or change TCPIVP.TCPPARMS(SYSLOG) to *IVP_configuration_dsn*(SYSLOG).
   - Copy the sample configuration file to the IVP-specific configuration files: /usr/lpp/tcpip/samples/syslog.conf to *IVP_configuration_dsn*(SYSLOG).

     From TSO, this can be done with an OGET command (note that the command should be issued on one line):

     ```
     oget '/usr/lpp/tcpip/samples/syslog.conf'
     'IVP_configuration_dsn (SYSLOG)'
     ```

6. Customize FTP

   - Set up the FTPD proc
     - Copy the sample start procedure from the SEZAINST library to the appropriate proclib: *target_lib_hlq*.SEZAINST(FTPD) to *IVP_proclib_dsn(FTPD_procname)*.
     - If desired, change the default jobname to the chosen procedure name on the //PROC statement: FTPD to *FTPD_procname*.
     - Comment out the first PARM= card on the //EXEC statement. Uncomment the SET statement and both lines of the second PARM= on the //EXEC statement. Change PARM1= on the // SET statement to point to the appropriate TCPIP.DATA data set.
   - Uncomment the //SYSFTPD DD statement which specifies TCPIVP.TCPPARMS(FTPSDATA).
     - On the //SYSFTPD DD statement, either accept the default location for the server FTP.DATA file or change TCPIVP.TCPPARMS(FTPSDATA) to *IVP_configuration_dsn*(FTPSDATA).
   - Uncomment the //SYSTCPD DD statement which specifies TCPIVP.TCPPARMS(TCPDATA).
   - On the //SYSTCPD DD statement, either accept the default location for the TCPIP.DATA set or change TCPIVP.TCPPARMS (TCPDATA) to *TCPIP_DATA_name*.

7. Set up the server and client FTP.DATA files

   - Copy the sample server FTP.DATA file to the appropriate server FTP.DATA file: *target_lib_hlq*.SEZAINST(FTPSDATA) to *IVP_configuration_dsn*(FTPSDATA)
   - Copy the sample client FTP.DATA file to the appropriate client FTP.DATA file: *target_lib_hlq*.SEZAINST(FTCDATA) to *IVP_configuration_dsn*(FTCDATA)

- **Note**: See *z/OS Communications Server: SNA Resource Definition Reference* and *z/OS Communications Server: SNA Network Implementation Guide* for detailed information on setting up and verifying VTAM configuration.

- Ensure the *IVP_Default_LUs* are defined to VTAM. Sample definitions are provided in *target_lib_hlq*.SEZAINST(IVPLU).
  - The *IVP_Default_LUs* must be defined in a dataset that is in the concatenation of datasets specified on the VTAMLST DD statement in the procedure used to start VTAM.
  - If changes were made to the VTAMLST concatenation, stop and restart VTAM.
- Refresh LLA

  If LLA is running and has not been refreshed since the Communications Server code was installed, refresh LLA before proceeding with the IVP.

8.5.5.3.8.1 CICS sockets interface customization considerations

Refer to the *z/OS Communications Server: IP CICS Sockets Guide* for the customization required to use the CICS Sockets Interface.

8.5.5.3.8.2 IMS sockets interface customization considerations

Refer to the *z/OS Communications Server: IP IMS Sockets Guide* for the customization required to use the IMS Sockets Interface.

8.5.5.3.8.3 Network Print Facility customization considerations

Refer to the *z/OS Communications Server: IP Configuration Reference*.

8.5.5.3.8.4 Communications Server Security Level 3 customization considerations

Refer to the *z/OS Communications Server: IP Configuration Reference*.

8.5.5.4 Customization considerations for Wave 1B

The following sections contain additional customization considerations for Wave 1B.

8.5.5.4.1 DFSMS customization considerations

For information on activating Object Access Method, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* or *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

For information on activating DFSMShsm, see *z/OS DFSMShsm Implementation and Customization Guide*.

For information on activating DFSMSrmm, see *z/OS DFSMSrmm Implementation and Customization Guide*.

8.5.5.4.1.1 Make ISMF available to the TSO user

To make ISMF available to the TSO users, refer to *z/OS DFSMS Using the Interactive Storage Management Facility*. For cataloging information, see *z/OS DFSMS Managing Catalogs*.

8.5.5.4.1.2 3800/3900 Printing Subsystem

If you are installing an IBM 3800 or 3900 Printing Subsystem for the first time, you must install library character sets, graphic character modification modules, and character arrangement tables in SYS1.IMAGELIB. A job to accomplish this task is contained in member LCSBLD1 of SYS1.SAMPLIB after DFSMS has been installed. If you have run LCSBLD1 previously, you do not need to rerun it after installing DFSMS.

8.5.5.4.1.3 Starter set information

The Starter Set is a pre-defined Source Control Data Set (SCDS) containing information that defines a specific storage management policy.

This is provided as a sample SCDS to allow new SMS installations to start SMS before having to go through defining the construct definitions, and writing the Automatic Class Selection (ACS) routines.

The sample SCDS named SCDS.PRIMARY.LINEAR is packaged in the target library SYS1.IGDVBS1 as member DFPSSCDS.

SYS1.SAMPLIB will contain the following members:

DFPSSDCR source ACS routine for DATACLAS

DFPSSMCR source ACS routine for MGMTCLAS

DFPSSSCR source ACS routine for STORCLAS

DFPSSSGR source ACS routine for STORGRP

DFPSSIMP JCL to IMPORT SCDS

Note: After the SMP/E APPLY is performed for DFSMS, run job DFPSSIMP available in SYS1.SAMPLIB to IMPORT 'SCDS.PRIMARY.LINEAR' from target library SYS1.IGDVBS1.

For more information about the Starter Set, refer to the *z/OS DFSMSdfp Storage Administration*. This manual contains a detailed description of the sample SCDS.

8.5.5.4.1.4 Activate CDRA

If your installation will use the Character Data Representation Architecture (CDRA) and numeric conversion routines provided in this release of z/OS, you must activate the function by tailoring and running SYS1.SAMPLIB(CDRAINIT).

Language Environment is required for Distributed FileManager (DFM/MVS) to do CDRA conversions. If your installation's link list does not refer to the Language Environment runtime library, SCEERUN, you must add the appropriate STEPLIB DD statement to DFM startup procedure SYS1.PROCLIB(DFM). You must also make the same change to SYS1.SAMPLIB(GDETPDEF) and rerun it.

8.5.5.4.2 z/OS UNIX System Services Application Services customization

For details on getting the shell and utilities in operational status, see *z/OS UNIX System Services Planning*.

8.5.5.5 Customization considerations for Wave 1C

The following sections contain additional customization considerations for Wave 1C.

8.5.5.5.1 EREP customization considerations

The JCL job used to run EREP should include an EXEC PGM=IFCEREP1 statement. Refer to *EREP User's Guide* for more information. For more information on creating, maintaining, and refreshing the SYS1.LOGREC data set or the Logrec Log Stream, see *z/OS MVS Diagnosis: Tools and Service Aids*.

8.5.5.5.2 ESCON Director Support customization considerations: The following sections describe customization considerations for ESCON Director Support.

8.5.5.5.2.1 ESCON Director Support security

Installation of ESCON Director Support enables nonauthorized programs to issue I/O to online switches using the EXCP access method, if the switch control unit/device has been defined to z/OS V2R5. To control access to a switch from a host processor, any combination of the following can be used:

- Keep the device offline to z/OS V2R5 to prevent nonauthorized programs from allocating the device. OFFLINE=YES is the IOCP default. (ESCON Director Support itself does not require a switch to be online to z/OS V2R5 in order to retrieve device information.)
- Add a z/OS V2R5 system authorization facility (SAF) exit to monitor and control allocation to switches. z/OS V2R5 allocation will call SAF whenever a user or program allocates a switch.
- Monitor and control allocation to switches using RACF. RACF provides a resource class, DEVICES, to control allocation to unit record, teleprocessing, and graphics devices.

8.5.5.5.3 FFST customization considerations

See FFST operations in *z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT* for more information on how to get the program into operational status by coding macros.

- Considerations when invoking FFST in a NetView environment:

  Make sure NetView tasks CNMCALRT and BNJDSERV are active for Alerts to appear on the NPDA alert screen. See the help for NCCF LIST and NCCF START for more information. If your installation uses NetView filters to prevent certain data from being collected or displayed, ensure that a filter is not blocking CPU information. You may automate the setting of the CPU hardware monitor filter during NetView initialization by updating CNMSTYLE as follows:
  – Locate "PDFILTER" and in that section add the following statement to allow FFST to pass CPU Generic Alerts to NetView, where *x* represents a unique suffix.

    ```
    NPDA.PDFILTER.x = NPDA SRFILTER AREC PASS TREF CPU
    ```

  See *NetView Administration Reference* for more information about this statement.
- Considerations when invoking FFST with RACF:

  If you use RACF or another security program to protect any of the FFST data sets, verify the application programmer has the necessary access for the following data sets:
  – FFST message log to review the messages that are issued when a probe statement is executed
  – FFST CLIST library to invoke the translators and to format the diagnostic output

- FFST panel library to invoke the help screens for the translators and the dump formatter
- Data sets allocated for the outputs produced by FFST
- In addition, you need to associate the RACF SYSSTC1 user ID with the FFST started task using the RACF STARTED class or using an entry in the RACF started procedures table, ICHRIN03. See *z/OS Security Server RACF System Programmer's Guide* for additional information.

8.5.5.5.4 GDDM and GDDM-PGF customization considerations

Refer to *GDDM System Customization and Administration* for more information on GDDM customization. Customization includes tuning the operating-system-related values and defining input and output devices to GDDM. These activities include:

- Activating language features
- Updating GDDM defaults
- Defining GDDM to TSO
- Defining GDDM to CICS
- Defining GDDM to IMS
- Making GDDM-OS/2 available
- Making GDDM-PCLK available
- Making GDDM-REXX available
- Reblocking any existing objects

8.5.5.5.5 ICKDSF customization considerations

ICKDSF provides a sample job for creating the stand-alone tape. Edit and submit SYS1.SAMPLIB(ICKSAT) to copy the ICKDSF Stand-Alone program to an unlabeled tape. Consult the instructions in the sample job for more information. Successful creation of a stand-alone tape returns a condition code of 0. Refer to the *Device Support Facilities (ICKDSF) User's Guide and Reference* for additional information on loading the stand-alone tape.

8.5.5.5.6 TSO/E customization considerations

Refer to *z/OS TSO/E Customization* for more information on TSO/E customization.

8.5.5.5.7 z/OS Host - 3270 Workstation File Send/Receive customization considerations

The following sections describe various customization considerations for z/OS Host - 3270 Workstation File Send/Receive.

8.5.5.5.7.1 Modify translate tables for z/OS Host - 3270 Workstation File Send/Receive

The translate tables included in this program, which convert between ASCII and EBCDIC codes, might not be suitable for your installation. These tables may be modified by patching the IND$FILE module.

The translate character that is most likely to cause problems is the EBCDIC character `X'3F'` that translates to an ASCII `X'1A'`. This later character is treated as an end-of-file (EOF) in many personal computer programs, especially editors.

The entry points to the IND$FILE module are:

EBCDASCI EBCDIC to ASCII translate table

ASCIEBCD ASCII to EBCDIC translate table

Refer to the binder listing for the proper offsets for these entry points. It is advised that the USERMOD SMP/E format be used to modify the translation tables.

The example below shows how to change the EBCDIC character “3F” to translate to “5B” instead of “1A”:

```
1. Find the offset of the EBCDIC-to-
   ASCII translation table (EBCDASCI)             020F40
2. Add the character value "3F" to that address      +   3F
                                                   ------
3. This is the verify address                    = 020F7F

++USERMOD(*******) .              (******* = ALPHABETIC NAME)
++VER(Z038)   FMID(HFX1102) .
++ZAP(EBCDASCI) .
  NAME EBCDASCI
VER 020F7F 1A
REP 020F7F 5B
  IDRDATA *******
```

If you change the EBCDASCI table to translate X'3F' to some other character, you should also change the ASCIEBCD table to translate that ASCII character back to X'3F'.

8.5.5.6 Customization considerations for Wave 1D:

The following sections describe customization considerations and tasks for Wave 1D.

8.5.5.6.1 Alternate Library for REXX customization considerations

The following sections describe various considerations for customizing Alternate Library for REXX.

8.5.5.6.1.1 Activating Alternate Library for REXX

Alternate Library for REXX is fully operational when the SMP/E installation and the customization activities described below are complete. If you do not APF-Authorize the load module library, REXX.SEAGALT, you will get an abend condition when the Alternate Library for REXX routines are requested by an APF-Authorized routine.

8.5.5.6.2 Security Server (RACF) customization considerations

Refer to *z/OS Security Server RACF System Programmer's Guide* for information on z/OS Security Server RACF customization and refer to *z/OS Upgrade Workflow* for migration actions.

8.5.5.6.2.1 Prepare the RACF database

Before you load the initial program with the new RACF active, complete either of the following actions:

- Allocate, format, and catalog a RACF database on the target system (IRRMIN00 with PARM=NEW).
- Upgrade the existing RACF database templates to the current level of the RACF templates that are shipped with z/OS V2R5 (IRRMIN00 with PARM=UPDATE).

If your templates are already at the level required by this release, you do not need to update the templates. Message IRR8025 may be issued and no further action is required.

You can create or upgrade a RACF database by running the IRRMIN00 utility, as described in *z/OS Security Server RACF System Programmer's Guide*. Sample jobs are provided in SYS1.SAMPLIB member RACJCL.

It is recommended that a backup database is created that RACF will maintain as a duplicate of the primary. Those migrating from a prior level of RACF should not update both the primary and backup database templates at the same time.

8.5.5.6.2.2 Automate Dynamic Parse initialization

Note: Because all RACF installations must activate dynamic parse after each IPL, it is **strongly** recommended that you automate dynamic parse initialization.

The IRRDPI00 command builds a “dynamic” table for parsing segment-related keywords. Dynamic parse is used to add, list, alter, or delete DFP, TSO, or any other nonbase segment information with the RACF commands.

There are two methods to automate dynamic parse initialization. The first involves updating the RACF parameter library to automatically invoke the IRRDPI00 UPDATE command at every IPL. This method requires the RACF subsystem address space to be active. Example of the first method (IRROPTxx) is shown in Figure 41 on page 130. The second method involves PARMLIB and PROCLIB updates to START IRRDPTAB at every IPL. These methods are described in *z/OS Security Server RACF System Programmer's Guide*.

8.5.5.6.2.3 Additional considerations before IPLing your system

The following is a list of considerations to consider before IPLing your system. For information on these topics, see *z/OS Security Server RACF System Programmer's Guide*.

- Define RACF coupling facility structures.
- Create or update the database range table.
- Create or update the database name table or alter the master JCL. Note that if this step is not done, the system prompts the operator during IPL. The operator can then reply with the RACF database name.
- Either the RACF database table or the RACF range table or both, maybe replaced by specifying the equivalent information in a IRRPRMxx member in parmlib. Up to 3 RACF parmlib members maybe concatenated together.

  Note: The existence and use of a RACF parmlib member will override all other specifications.
- Update the RACF class descriptor table and the RACF router table.
- Identify the RACF address space as an MVS subsystem.
- Create or update the started procedures table or use the STARTED CLASS.
- Create exit routines and a naming convention table.

You might also want to consider activating the REXX RACVAR function, which is described in *z/OS Security Server RACF Macros and Interfaces*.

8.5.5.6.2.4 Other system considerations for initial installation

If you plan to make the change listed below, be sure to do so before you define your users, groups, and resources to RACF:

- Add TSO profiles to the UADS data set for any new TSO users by using the ADD subcommand of TSO/E ACCOUNT. See *z/OS TSO/E Administration*.

  Note: RACF provides your installation with the option of placing TSO user logon information in the TSO segment of the RACF user profile.

8.5.5.7 Customization considerations for Wave 1E:

The following sections describe considerations for customizing elements that are part of Wave 1E.

8.5.5.7.1 Runtime Library Extensions customization considerations

The following sections describe considerations for customizing Runtime Library Extensions.

8.5.5.7.1.1 Customization of the Runtime Library Extensions for CICS

To ensure that CICS can communicate with the Runtime Library Extensions, you must add a Runtime Library Extensions entry to the CICS System Definition (CSD). To do so, update the CICS System Definition (CSD) file using the program definitions in member CLB3YCSD in the Runtime Library Extensions data set SCLBJCL. This member contains the necessary input to the CICS System Definition (CSD) file utility program to define Runtime Library Extensions routines in the PPT. The group name for the Runtime Library Extensions routines is **CLB**.

Note: If you intend to use the Runtime Library Extensions under CICS, add the SCLBDLL data set to the DFHRPL DD concatenation in your CICS startup job stream and ensure that Language Environment is available under CICS (see 8.5.5.3.2, “Language Environment customization considerations” on page 158 for more information).

8.5.5.7.2 CIM customization considerations

The customization for element CIM has to be done after IPL of your target system. For detailed information on how to customize CIM for z/OS, refer to *z/OS Common Information Model User's Guide*.

8.5.5.7.3 RMF customization considerations

If you are a new RMF customer, skip the first two steps. These steps are optional for existing RMF customers.

1. Redefine the user reports for RMF, using the ERBRMFU utility.
2. Reinstall your user exits, ensuring that they run in 31-bit mode.
3. Customize the RMF Monitor III CLISTs.

   If the RMF CLISTs and default naming conventions are not acceptable, you can modify the RMF CLIST ERBRMF3X, which invokes RMF, displaying the primary menu. This CLIST can be found in your RMF SERBCLS system library. Follow the instructions in the commented text to customize the CLIST to your local environment.

4. Refer to *z/OS RMF User's Guide*, for a description of all customization steps that are required for new RMF customers as well as for installations that are migrating from a previous release.

8.5.5.7.4 XL C/C++ Compiler customization considerations

The following topics describe areas you may consider for tailoring for the XL C/C++ compiler.

8.5.5.7.4.1 Tailor the TSO environment: If you want to use TSO or compile a program with the XL C/C++ Compiler, refer to Figure 44 on page 149 for information on the data sets, which must be concatenated. Modify the TSO/E logon PROC by concatenating the listed data sets to the corresponding DDNAMEs.

8.5.5.7.4.2 Tailor REXX EXECs

For information on tailoring REXX EXECs, refer to the *z/OS XL C/C++ User's Guide*.

8.5.5.7.4.3 Customized default options for XL C/C++ Compiler

For information about customizing default options for the XL C/C++ compiler, see *z/OS XL C/C++ User's Guide*.

8.5.5.7.4.4 Customization of the Performance Analyzer for CICS

To ensure that CICS/ESA can communicate with the Performance Analyzer, you must add Performance Analyzer entries to the CICS processing program table (PPT). To do so, update the CICS system definition (CSD) file using the program definitions in member CTVSCSD in the Performance Analyzer data set SCTVJCL. PTF UQ35263 must be installed to obtain the member. This member contains the necessary input to the CICS System Definition (CSD) file utility program to define the Performance Analyzer routines in the PPT. The group name for the Performance Analyzer routines is EQA.

Note: If you intend to use the Performance Analyzer under CICS, add the SCTVMOD data set to the DFHRPL DD concatenation in your CICS startup job stream and ensure that Language Environment is available under CICS (refer to 8.5.5.3.2, “Language Environment customization considerations” on page 158 for more information).

8.5.5.8 Customization considerations for Wave 1F: The following sections describe customizations considerations for Wave 1F.

8.5.5.8.1 BDT customization considerations

Perform the following steps to activate BDT; these steps are described in more detail in *z/OS BDT Installation*. All samples are in the SBDTSAMP library.

- If a JES3 environment:
  - Add CONSOLE and SYSID statements to the JES3 initialization stream.
  - If installing SNA/NJE, add or modify the NJERMT statement in the JES3 INISH stream.
- Update SYS1.VTAMLST (see sample BDT$VTAM) to define a Communications Server for z/OS SNA Services application to support BDT.
- Allocate BDT and TQI data sets (see sample BDT$ALOC).
- Run the BDTTQBCH utility program (see sample BDT$TQFM) to format the:
  - TQI checkpointing and bitmap data sets

  - Message data set for each processor.
- Create the BDT initialization stream:
  - See sample BDT$FTF for a JES2 or JES3 file-to-file only installation.
  - See sample BDT$NJE for a JES3 SNA NJE only installation.
  - See sample BDT$MIX for a JES3 file-to-file and SNA NJE installation.
- Create user exits, as described in *z/OS BDT Installation*. Assemble and link-edit the BDT user exits into the proper load libraries. If you are a migrating customer, you must reassemble all user exit routines.

8.5.5.8.2 DFSORT customization considerations

Refer to *z/OS DFSORT Installation and Customization* for more information and instructions on DFSORT customization.

8.5.5.8.2.1 Change DFSORT installation options

Starting from z/OS V1R10, you can change DFSORT installation options by creating and using ICEPRMxx members in concatenated PARMLIB. This method simplifies the management of DFSORT installation options, particularly in multisystem environments.

You can activate the ICEPRMxx members that you want to use by issuing an appropriate START ICEOPT command from the console, or by including a START ICEOPT command in an appropriate COMMNDxx member in PARMLIB. Before you activate your ICEPRMxx members by using a COMMNDxx member, it is recommended that you test the ICEPRMxx members by activating them from the console. ICEOPT issues messages to the console to indicate the status of your changes to the installation defaults, and errors that it detects. Be sure to review these console messages and the return code from ICEOPT.

It is recommended that you use ICEPRMxx members to change DFSORT installation options because they are easier to use and more flexible than the old method using the ICEMAC macro and SMP/E USERMODs. However, the old method is still supported.

In a multisystem environment, you can only activate ICEPRMxx parmlib members that contain DFSORT installation options in z/OS V1R10 or later. For z/OS V1R10 or later, the installation options in the activated ICEPRMxx members are merged with the ICEMAC installation options at run time.

Give the user ID that is assigned to the ICEOPT started task READ access to the ICEPRMxx PARMLIB members. For example,

```
ADDSD 'SYS1.PARMLIB' UACC(NONE)
PERMIT 'SYS1.PARMLIB' CLASS(DATASET) ID(userid) ACCESS(READ)
```

Note: The ICEOPT procedure requires READ authority to all of the data sets in the data sets in the concetenated PARMLIB.

See *z/OS DFSORT Installation and Customization* for information about ICEPRMxx members, the START ICEOPT command, ICEOPT messages and return codes, and the syntax for options that you can specify using the ICEPRMxx members.

Note: You can control who can issue the START ICEOPT command by using MVS Console Security. See *MVS Planning: Operations* for information about controlling who can issue the START ICEOPT command.

8.5.5.8.2.2 Change DFSORT SVC name

Sample jobs ICESVREC and ICESVAPP set the name of the SVC that DFSORT uses. You can change the default SVC name by either specifying an alternate routing code or assigning a user SVC number. With slight modification, you can use ICESVREC and ICESVAPP sample jobs in SICESAMP to create this SVC.

- To specify SVC 109 with the alternate routing code:
  - Install an SMP/E USERMOD that creates a copy of IGX00017 named IGX00038; then load it into LPA or MLPA.
  - In the sample job ICESVREC, change the name ICESVC to IGX00038.
  - Modify the DFSORT default installation options to specify SVC=(,ALT). You can perform this modification any time after you apply the DFSORT element by using the ICEPRMxx PARMLIB member (recommended), or by using the ICEMAC macro and an SMP/E USERMOD. See the "Changing the Installation Defaults" topic in *z/OS DFSORT Installation and Customization*.
- To assign a user SVC number:
  - Select an available type-3 SVC number nnn in the range 200-255.
  - Install an SMP/E USERMOD to create a copy of IGX00017 with a different name. For example, you could name the copy ICESVC.
  - Create or modify an IEASVCxx member of parmlib to add the SVC. If you used the name ICESVC for the copy of IGX00017, and chose 255 as the SVC number, the entry would look like this:

    ```
    SVCPARM 255,REPLACE,TYPE(3),EPNAME(ICESVC)
    ```

  - Then load a copy of IGX00017 into LPA or MLPA.
  - Modify the DFSORT default installation options to specify SVC=nnn, where nnn is the SVC number that you selected. You can perform this modification any time after you apply the DFSORT element by using the ICEPRMxx PARMLIB member (recommended), or by using the ICEMAC macro and an SMP/E USERMOD. See the "Changing the DFSORT Installation Defaults" topic in *z/OS DFSORT Installation and Customization*.
- Load the initial program.

Note: DFSORT can be tested without making the SVC available as long as you do not write DFSORT SMF records (type 16 records).

Without the SVC available, DFSORT performance with input, output, or work data sets on cached DASD devices (for example, IBM 3990 Model 3) may be degraded. Always use the SVC if you will be running benchmarks.

The DFSORT SVC is available after it is loaded in LPA. You can load the SVC in LPA by using one of the following methods:

- Add the SVC load module to MLPA by specifying it in the IEALPAxx PARMLIB member that is used for IPL.

- On a supported release of z/OS, add the SICELPA library to the LPA list in the LPALSTxx PARMLIB member that is used for a cold start (CLPA) IPL.

For more information about loading modules in LPA, see *z/OS MVS Initialization and Tuning Guide*, *z/OS MVS Initialization and Tuning Reference*, and *z/OS MVS System Commands*.

8.5.5.8.2.3 Replacing IEBGENER with ICEGENER (optional)

You can achieve more efficient processing for jobs set up to use the IEBGENER system utility by using the ICEGENER facility of DFSORT. Qualifying IEBGENER jobs are processed by the equivalent (though not identical), but more efficient, DFSORT copy function. If, for any reason, the DFSORT copy function cannot be used (for example, when IEBGENER control statements are specified), control is automatically transferred to the IEBGENER system utility. DFSORT transfers control to IEBGENER using the name for IEBGENER from the environment installation module (ICEAM2 or ICEAM4) or time-of-day installation module (ICETD1-4) used for the run. The ICEGENER facility can be used either for selected IEBGENER jobs or automatically for all IEBGENER jobs.

You can invoke ICEGENER in one of the following ways:

- For selective use, substitute the name ICEGENER (or the alias SORTGENR) for the name IEBGENER in any jobs you choose.
- To make ICEGENER the system's default "IEBGENER", install an SMP/E USERMOD to create an alias of “IEBGENER” for ICEGENER. With slight modification, you can use the ICEGAREC and ICEGAAPP sample jobs in SICESAMP to create this alias. Then place the ICEGENER load module ahead of IEBGENER in the system's search order for programs. For example, you can place the ICEGENER load module with its new alias in LPA. All jobs and programs that call IEBGENER will then execute ICEGENER instead. If ICEGENER needs to call IEBGENER, it will do so automatically. For information about placing modules in LPA and Dynamic LPA, and about placing libraries ahead of the LINKLIB library in the link list, see *z/OS MVS Initialization and Tuning Reference*.

For notes and restrictions pertaining to the use of ICEGENER, see *z/OS DFSORT Installation and Customization*.

8.5.5.8.2.4 Make DFSORT programs available

After you install DFSORT, make its programs available on the system. To activate the ICEPRMxx members using the START ICEOPT started task command of DFSORT, ensure that the ICEOPT PROC of DFSORT is in PROCLIB. See the “System Planning Considerations” chapter in *z/OS DFSORT Installation and Customization* for information about using the START ICEOPT command.

To run DFSORT resident, complete the following actions:

- Load the modules from SORTLPA into LPA or Dynamic LPA.
- Add SIEALNKE before the SICELINK to the link list.

To run DFSORT nonresident, perform either of these actions:

- Add SIEALNKE before the SICELINK and SORTLPA (in this order) to the link list (this assumes SORTLPA is removed from the LPA list).
- Use a JOBLIB or STEPLIB for SIEALNKE, SICELINK and SORTLPA (in this order).

If you plan to record DFSORT SMF type-16 records or use IBM cached DASD devices, the DFSORT SVC must be accessible:

- Load the DFSORT SVC from SICELPA into LPA or MLPA.
- IPL to make the SVC available.

Notes:

1. For information about adding modules to LPA, Dynamic LPA, and the link list, and for information about assigning user SVC numbers, see *z/OS MVS Initialization and Tuning Guide* and *z/OS MVS Initialization and Tuning Reference*.
2. You can combine SORTLIB with SICELINK. If you combine these libraries, you will only need to supply a SORTLIB DD statement for tape work data set sort or Conventional merge applications that use dynamic linkedit of user exits. You will have to specify SORTLIB=SYSTEM as an installation default by using an ICEPRMxx member of PARMLIB (recommended), or by using the ICEMAC macro and an SMP/E USERMOD. See the "Changing the Installation Defaults" topic in *z/OS DFSORT Installation and Customization*.

8.5.5.8.2.5 Invoking DFSORT from a REXX CLIST

DFSORT can be invoked from a REXX CLIST. If DFSORT is available in the program search order, you can invoke DFSORT with:

```
ADDRESS LINKMVS ICEMAN
```

If DFSORT is not available in program search order, you can invoke DFSORT with:

```
"CALL 'SYS1.SICELINK(SORT)'"
```

SYS1.SICELINK is the name of your SICELINK library used for installation.

8.5.5.8.3 HCM customization considerations

For information about setting up TCP/IP communication between HCM on the workstations and HCD on the z/OS host or for information about downloading the workstation code, see *z/OS and z/VM HCM User's Guide*.

8.5.5.8.4 IBM Knowledge Center for z/OS customization considerations

For information on configuring, customizing and setting up RACF security for IBM Knowledge Center for z/OS, refer to *IBM Knowledge Center for z/OS Configuration and User Guide* for V2R5. Note that the default tree structure created by the Knowledge Center post-installation configuration scripts has changed. As a result, see *z/OS Upgrade Workflow* for actions required if migrating Knowledge Center data from V2R2.

8.5.5.9 Customization considerations for Wave 1G

8.5.5.9.1 RACF updates for zFS

To use zFS, you must create RACF definitions. For detailed information about the required group, user, and started tasks, refer to one of the appropriate books:

- *z/OS File System Administration*

8.5.5.9.2 Infoprint Server Customization Considerations

Detailed steps to get the program into operational status are defined in *z/OS Infoprint Server Customization*.

8.5.5.9.2.1 Initialize NetSpool Message Log Data Set

If you are using NetSpool, you may continue to use your existing message log data set, or you may record messages in the common message log. If you are customizing NetSpool for the first time, use of the common message log is recommended. If you want to also use the optional NetSpool message log data set, SYS1.SAMPLIB member APIMIJCL can be used to allocate and initialize the NetSpool message log data set. Refer to *z/OS Infoprint Server Customization* for information on using the common message log.

8.5.5.9.2.2 Initialize IP PrintWay basic mode Operational Data Sets

This step is required **only** for IP PrintWay basic mode. If you are using (or migrating to) IP PrintWay extended mode, you do not need these data sets.

For IP PrintWay basic mode, the transmission queue and message log data sets must be allocated and primed. The following jobs are provided in SYS1.SAMPLIB to perform this function:

ANFDEAL	Allocates IP PrintWay transmission queue data set
ANFQINIT	Initializes IP PrintWay transmission queue data set
ANFMIJCL	Allocates and initializes IP PrintWay message log data set

If you used IP PrintWay in a previous z/OS release, you may use your existing transmission queue and message log data sets.

8.5.5.9.2.3 Customize the UNIX environment for Infoprint Server

The customization steps described in this section are sufficient for a default configuration. Only the lpd and printer inventory daemons are started in the default configuration. If you want to use any of the following functions, additional customization is required:

- Infoprint Central
- Infoprint Server subsystem
- IP PrintWay Extended Mode
- IPP Server
- NetSpool
- Transform manager

Refer to *z/OS Infoprint Server Customization* for additional customization for these functions.

1. Create the configuration files for the Infoprint Server daemons:

A sample job is provided in SYS1.SAMPLIB(AOPCPETC) to copy the sample configuration files from /usr/lpp/Printsrv/samples to /etc/Printsrv. Review the comments in the job to determine which configuration files are required for your installation, and then run the job to create the configuration files.

The defaults provided in the sample aopd.conf file will start the printer inventory manager and the LPD. If you want to modify any of the defaults, refer to *z/OS Infoprint Server Customization*. If you are migrating from a previous level of Infoprint Server, and/or you are using NetSpool or IP PrintWay Extended Mode, you must update the aopd.conf file to start the appropriate daemons.

Note: Infoprint Server will default to search for its configuration file in `/etc/Printsrv/aopd.conf`. If you change the name or directory for this file, see *z/OS Infoprint Server Customization* for information on how to configure your environment.

2. Modify `/etc/profile` to update the following environment variables:

   - PATH

     Add `/usr/lpp/Printsrv/bin` to the PATH environment variable.

     Note: The `/usr/lpp/Printsrv/bin` directory must occur **before** `/bin` in the PATH environment variable so the InfoPrint Server (Print Interface and IP PrintWay extended mode) version of the lp, lpstat, and cancel commands will be invoked.
   - LIBPATH

     Add `/usr/lpp/Printsrv/lib` to the LIBPATH environment variable.
   - MANPATH

     Add `/usr/lpp/Printsrv/man/%L` to the MANPATH environment variable.

     Note: The `/usr/lpp/Printsrv/man/%L` directory must occur **before** `/usr/man/%L` in the MANPATH environment variable so the InfoPrint Server (Print Interface and IP PrintWay extended mode) version of the lp, lpstat, and cancel man pages will be displayed.
   - NLSPATH

     Add `/usr/lpp/Printsrv/%L/%N` to the NLSPATH environment variable.

   Infoprint Server may optionally use additional environment variables to set defaults for operation. Refer to *z/OS Infoprint Server Customization* for detailed information on the usage of these parameters.

3. Review the aopstart EXEC for possible updates. For security reasons, Infoprint Server daemons use some variables only if they are set in the aopstart EXEC. Daemons do not use the same variables set in the `/etc/profile` file. For information about how to set variables in the aopstart EXEC, refer to *z/OS Infoprint Server Customization*.

8.5.5.9.2.4 Establish security for Infoprint Server

1. Define or update RACF profiles for Infoprint Server security.

   A sample CLIST is provided in SYS1.SAMPLIB(AOPRACF) that can be used to define the security environment for Infoprint Server. This CLIST must be modified before it is run. All commands are commented out by default. The CLIST can be used to perform the following functions:

- Define two RACF groups for print administrators and operators. The default names for these groups are:

  AOPADMIN RACF group for administrators who maintain the Printer Inventory.

  AOPOPER RACF group for operators who will control starting and stopping the Infoprint Server daemons.

  Note: You may use one RACF group for both AOPADMIN and AOPOPER if your installation's security policy does not require a distinction between operators and administrators. For more information on security considerations for Infoprint Server, refer to *z/OS Infoprint Server Customization*.
- Create a RACF profile in the PRINTSRV class named AOP.ADMINISTRATOR to control who has access to the Printer Inventory. You must name this profile AOP.ADMINISTRATOR. Alternatively, you can use your existing AOPADMIN profile in the FACILITY class. You can also create an optional profile named AOP.CONFIGURATION in the PRINTSRV class. Refer to *z/OS Infoprint Server Customization*, for information about why it is desirable to use the PRINTSRV class profile rather than the previously-defined FACILITY class profile.
- Define the JCL startup procedures in the STARTED class and define a user ID to be associated with these started procedures.

2. Customize permissions:

   Run the aopsetup shell script to define permissions, specifying the names of the operator group and administrator group you defined in the previous step. To run the aopsetup shell script, you must run it from a user ID that has UID=0 or has read access to the BPX.SUPERUSER facility class.

   Note: The aopsetup shell script requires access to the directory defined as base-directory in the aopd.conf configuration file. The default for base-directory is /var/Printsrv. Therefore, you must have access to the /var/Printsrv directory before you can run aopsetup. If you have your target system's /var file system mounted to your driving system, you may run aopsetup from your driving system. If you do not have access to the target system's /var from your driving system, you must run aopsetup from the target system. If you want to run aopsetup from the driving system, refer to the comments in the aopsetup shell script for information on the proper syntax to invoke the script.

   To run the aopsetup shell script using the default names listed in the previous step, enter:

   ```
   /usr/lpp/Printsrv/bin/aopsetup AOPOPER AOPADMIN
   ```

   You can run aopsetup from an rlogin shell, from an OMVS session, or using BPXBATCH.

8.5.5.9.2.5 Make updates to the Communications Server IP Profile

Print Interface requires the use of TCP port 515 in a default configuration. Ensure dataset hlq.PROFILE.TCPIP does not reserve port 515 for another application. You may need to ensure the ports listed below are not reserved, if you want to use these functions of Infoprint Server:

IPP server	port 631

See *z/OS Communications Server: IP Configuration Reference* for more information.

8.5.5.9.2.6 Customize IP PrintWay and NetSpool

If you are using the IP PrintWay or NetSpool components of Infoprint Server, additional customization is required.

- For NetSpool, logical printers must be defined to VTAM with APPL definition statements.
- For IP PrintWay Basic Mode, the functional subsystem must be defined to JES.
- For IP PrintWay Extended Mode, job selection criteria must be defined in the printer inventory.

Refer to *z/OS Infoprint Server Customization* for instructions on customizing IP PrintWay and NetSpool. If you are migrating from previous levels of Infoprint Server, see *z/OS Upgrade Workflow* for migration considerations.

8.5.5.9.2.7 Customize HTTP Server for Infoprint Central

If you want to use the print management functions provided by Infoprint Central for the web, you must customize the IBM HTTP Server - Powered by Apache, and update the `conf/httpd.conf` and `bin/envvars` files to include configuration information for Infoprint Server. See *z/OS Infoprint Server Customization* for instructions on customizing the HTTP Server for use with Infoprint Server.

8.5.5.9.2.8 Download client code to workstation

The Infoprint Port Monitor allows Windows users to print to any z/OS printer defined in the Printer Inventory, using standard print-submission methods from Windows applications that support printing. The Infoprint Port Monitor is located in a subdirectory of the `/usr/lpp/Printsrv/win/` directory, depending on locale. It is downloaded to the workstation using FTP. The directory locations for each NLV are shown below:

English	`/usr/lpp/Printsrv/win/En_US/`
Japanese	`/usr/lpp/Printsrv/win/Ja_JP/`

Refer to *z/OS Infoprint Server User's Guide* for instructions to download the client code to the workstation.

8.5.5.9.3 Network File System customization considerations

Detailed steps to get the program into operational status are defined in *z/OS Network File System Guide and Reference*. Omitting these steps will prevent successful operation of the program.

8.5.5.9.4 Configuring IBM z/OS Management Facility

To activate IBM z/OS Management Facility, you must configure the element after installing it as described in this program directory. For information about configuring IBM z/OS Management Facility, see IBM z/OS Management Facility Configuration.

8.5.5.10 Customization for Wave 2: Once you have successfully SMP/E APPLY'd JES2, JES3 and SDSF, you must customize the elements.

8.5.5.10.1 JES2 Initialization considerations: You will need to update the following PARMLIB and PROCLIB members prior to IPL of the new JES2 system:

- LNKLST*xx* to include libraries SHASLNKE and SHASMIG
- Logon procedures used for IPCS setup to include the JES2 libraries SHASMIG, SHASPARM, and SHASPNL0
- PROG*xx* to APF-authorize libraries SHASLNKE and SHASMIG
- JES2 cataloged procedures to include the library SHASLNKE, if the procedures contain a STEPLIB and the library names have been changed.

For details on how to create or update the initialization deck, see z/OS JES2 Initilaization and Tuning Reference.

8.5.5.10.2 JES3 Initialization considerations: You will need to update the following PARMLIB and PROCLIB members prior to IPLing the JES3 system:

- LNKLST*xx* to include JES3 libraries SIATLIB, SIATLINK, and SIATMIG (if STEPLIBs are not used in the JES3 cataloged procedures or you are using the JES3 DLOG facility).
- LPALST*xx* to include JES3 library SIATLPA
- PROGxx to APF-authorize libraries SIATLIB, SIATLINK, SIATMIG, and SIATLPA.
- JES3 cataloged procedures to include the libraries SIATLIB, SIATLINK, and SIATMIG if the procedures contain a STEPLIB and the library names have been changed.
- Logon procedures used for IPCS setup to include JES3 libraries SIATCLI0, SIATMSG0, SIATPNL0, and SIATTBL0.
- IFAPRD00 to include JES3. Ensure the correct program number is specified.

JES3 enforces the system name (specified in member IEASYM*xx* or IEASYS*xx* in PARMLIB) for the current system matches a name specified on the NAME parameter of a MAINPROC statement in your JES3 initialization deck.

For details on how to create or update the initialization deck, see z/OS JES3 Initialization and Tuning Reference.

8.6 Step 6: Verify installation of Wave 0, Wave 1 and Wave 2

After you have successfully completed the following steps, you should verify the installation of the Wave 0, Wave 1 and Wave 2 elements:

- SMP/E APPLYed the Wave 0, Wave 1 and Wave 2 elements,
- Performed the Post-APPLY work, and
- Finished the minimal customization documented in 8.5, “Step 5: Customize Wave 1 and Wave 2” on page 126,

The Wave 0, Wave 1 and Wave 2 installation verification procedures (IVPs) should be run from your **target** system, unless specifically noted.

8.6.1 IPL the z/OS system

Attention

Do not IPL the new release in a production environment until you have tested the new release with a simulated production load that includes all applications and all non-IBM products, which ensures service level agreements can be met.

Do not IPL in a shared resource environment unless you have installed service for any applicable toleration and coexistence PTFs on lower z/OS releases that will coexist with z/OS V2R5. To identify the required coexistence PTFs that must be installed on lower z/OS releases, receive the latest HOLDDATA and run the SMP/E MISSINGFIX command with a fix category (FIXCAT) of "IBM.Coexistence.z/OS.V2R5."

Do not IPL the z/OS V2R5 system with a root file system used with previous z/OS releases. Invoking programs contained in previous file systems, may result in unpredictable behavior.

Starting with z/OS V2R1, you cannot use earlier levels of JES2, JES3, or SDSF. Do not IPL the z/OS V2R5 system with older levels of JES2 or JES3 used with previous z/OS releases.

8.6.1.1 File system execution: When a z/OS system is loaded with the initial program, the root file system must be mounted in read-only mode during execution. After the Wave 1 elements were installed during SMP/E APPLY processing, customization of some elements requires updates to the target system /etc and /var file systems. These updates must have been completed at this point.

8.6.1.2 BPXISETS and BPXISETD: Before IPLing, you will need to convert the /etc and /var directories to symbolic links. First unmount the zFS data sets on the /etc and /var directories. Use the BPXISETS REXX exec found in SAMPLIB to convert the /etc and /var directories to symbolic links. To submit the REXX exec in the background, you can use the BPXISJCL provided in SAMPLIB.

If, for any reason, you require the /etc or /var symbolic links to be removed and the /etc or /var directories recreated, use the BPXISETD REXX exec from SAMPLIB. Again, the BPXISJCL job can be used to submit this in the background.

Ensure the separate zFS data sets for your target /tmp and /dev directories are created in preparation to IPL your z/OS V2R5 system. The BPXPRMxx member should contain the following mount statements:

```
ROOT   FILESYSTEM('root_FS_data_set')
       TYPE(xxx)  MODE(RDWR)

MOUNT  FILESYSTEM('etc_FS_data_set')
       MOUNTPOINT('/etc')
       TYPE(xxx)  MODE(RDWR)

MOUNT  FILESYSTEM('var_FS_data_set')
       MOUNTPOINT('/var')
       TYPE(xxx)  MODE(RDWR)

MOUNT  FILESYSTEM('tmp_FS_data_set')
       MOUNTPOINT('/tmp')
       TYPE(xxx)  MODE(RDWR)

MOUNT  FILESYSTEM('dev_FS_data_set')
       MOUNTPOINT('/dev')
       TYPE(xxx)  MODE(RDWR)
```

Figure 46. Mount Statements Required in BPXPRMxx

Notes:

1. 'root_FS_data_set' and 'etc_FS_data_set' are the file systems in which z/OS V2R5 was just installed.
2. 'var_FS_data_set' is the file system that you created in 6.2, “Step 1: Separating File System Data Sets for z/OS V2R5” on page 31.
3. 'tmp_FS_data_set' and 'dev_FS_data_set' are the file systems that you just created.
4. On the ROOT and MOUNT statements shown in Figure 46, change TYPE(xxx) to the appropriate TYPE, such as TYPE(ZFS), that is applicable to your installation.

If IBM z/OS Management Facility element is installed in a separate file system, ensure that the BPXPRMxx PARMLIB member is updated to add a mount statement for the file system used for the installation of the IBM z/OS Management Facility element to ensure the file system will be mounted during IPL of the z/OS V2R5 target system.

Update the BPXPRMxx PARMLIB member to add a mount statement for the separate file system in which IBM z/OS Liberty Embedded is installed to ensure that the file system will be mounted during IPL of the z/OS V2R5 system target system.

Update the BPXPRMxx PARMLIB member to add a mount statement for the file system in which the z/OS Container Extensions element is installed to allow the file system to be mounted during IPL of the z/OS V2R5 system.

To IPL the system, ensure you use the target SYSRES from the Wave 1 installation. In addition, the z/OS V2R5 zFS data sets in which you installed Wave 0, Wave 1 and Wave 2 elements, should be used as the root file system and the file system mounted on /etc directory when IPLing the system.

Be sure that you have synchronized your security databases; see 8.1.1, “Set up User and Group IDs Required for Installation” on page 72.

If you needed to define a user ID alias table for your security setup be sure to add the USERIDALIASTABLE keyword in your BPXPRMxx parmlib member specifying your table location. See *z/OS UNIX System Services Planning* for details.

IEA299I may be received during system initialization. This message is issued only if a conditional resource initialization module (RIM) is not found. Processing continues because z/OS V2R5 is not dependent on the function of the indicated RIM. However, subsystems, program products, or applications might require conditional RIMs. Check the installation procedures for any subsystem that fails for mention of the indicated RIM. Any RIMs shipped by these products must reside in the SYS1.NUCLEUS library.

The following messages might be issued one or more times, depending on which products you have installed and which I/O device types are defined to your system during IPL:

```
IEA093I MODULE IEANUC01 CONTAINS UNRESOLVED WEAK EXTERNAL REFERENCE <name>.
```

References to modules with any of the following names can be ignored:

```
AOMATTN   AOMATTNT  CBRATTN   IARYGGTS  IECTCATN  IECTCQSC
IRDVATT1  ISTZFMAA  ISTZFMAB  IXCIOATX
```

```
IEA093I MODULE IGGDDT01 CONTAINS UNRESOLVED WEAK EXTERNAL REFERENCE <name>.
```

All such messages for unresolved external references in IGGDDT01 can be ignored. For more information, see APAR II03282.

```
IEA093I  MODULE IGC116 CONTAINS UNRESOLVED WEAK EXTERNAL REFERENCE IGX03007.
```

8.6.2 Verify installation of z/OS V2R5 Wave 0, Wave 1 and Wave 2 FMIDs

To verify that the z/OS V2R5 elements are installed, you should make at least the following **minimal checks:**

- Initialize the system.
- Initialize JES.
- Submit a job and check its output. (You can run the BCP installation verification job provided in member IVPJOBS in your SAMPLIB data set.)
- If CICS or IMS is installed, initialize a region and sign on to a terminal.

For **more complete checks**, you can perform the following tasks:

- Run critical production jobs.
- Run a mixture of jobs, for example:
  - Jobs that produce large amounts of SYSOUT
  - Jobs that submit other jobs to the internal reader
  - Jobs with many SYSOUT data sets

  - Jobs with many DD statements
  - Jobs that perform dynamic allocations
- Test NJE lines.
- Test RJE/RJP lines.
- Send output to JES- and PSF-controlled printers.
- Issue TSO/E LOGON, TRANSMIT, RECEIVE, SUBMIT, STATUS, CANCEL, and LOGOFF commands.
- Ensure all exit routines and local modifications perform correctly.
- Communicate with all networks without IP.
- Test critical functions in applications.
- Check for completeness of accounting records.
- Test all non-IBM product functions.
- Ensure that performance goals stated in service level agreements can be met.

To verify JES3 is installed, you should make at least the following minimal checks:

Attention

- For JES3, if you are in a global/local complex, each system must be a member of the same sysplex.
- Do not IPL the new release in a production environment until you have tested the new release with a simulated production load that includes all applications and all non-IBM products, and ensures service level agreements can be met.

Minimal check:

- Initialize the system.
- Initialize JES.

  Note: For JES3, you should see at least the following message:

  ```
  *IAT3100 JES3 z v.r.m SYSTEM COLDSTART ON yyyy.ddd
  ```

- Submit a job.
- Check the job's output.

More complete checks:

- Perform all the JES3 restarts (for example, COLD, HOT, HOT with ANALYSIS, WARM, WARM with REPLACE). All restarts should be performed with various jobs in different stages of execution (for example, OUTSERV, C/I, MAIN).
- Bring up a JES3 global/local environment. Verify all systems connect properly. Submit jobs to all processors and verify the jobs run successfully.
- In the global/local environment, perform a DSI to switch global processors. Verify both processors connect properly. Once the DSI is complete, submit jobs to run on each processor. Verify the jobs run to completion.
- Perform a DSI again to switch the global back to the original processor.
- Bring up a C/I FSS address space if you plan to use one in production. Verify jobs run successfully through the C/I FSS.
- Run a mixture of jobs, for example:

- Jobs that produce large amounts of SYSOUT
  - Jobs that submit other jobs to the internal reader
  - Jobs with many SYSOUT data sets
  - Jobs with many DD statements
  - Jobs that perform dynamic allocations
  - Jobs that use or create dynamic output descriptors
  - DJC net jobs
- Test miscellaneous JES3 functions:
  - Inquiry and Modify commands
  - Dump Core utility
  - Display DSP
- Test NJE lines.
- Test RJE/RJP lines.
- Send output to JES- and PSF-controlled printers.
- Issue TSO/E LOGON, TRANSMIT, RECEIVE, SUBMIT, STATUS, CANCEL, and LOGOFF commands.
- Ensure all exit routines and local modifications perform correctly.
- Run all production jobs.
- Communicate with all networks.
- Test critical functions in applications.
- Check for completeness of accounting records.
- Test all non-IBM product functions.
- Ensure performance goals stated in service level agreements can be met.

Figure 47 (Page 1 of 2). Checklist for the IVPs

Check Box	Section, Step Description	Your Notes
	8.6.2.1, “IVP jobs for Wave 0” on page 189	
□	8.6.2.1.1, “Run the High Level Assembler Installation Verification Procedure” on page 190	
□	8.6.2.2, “Run the BCP and ISPF Installation Verification Procedure” on page 190	
	8.6.3, “IVP jobs for Wave 1A” on page 193	
□	8.6.3.2, “Run the Language Environment Installation Verification Procedure” on page 196	
	8.6.4, “IVP jobs for Wave 1B” on page 196	
□	8.6.4.1, “Run the DFSMS Installation Verification Procedure” on page 196	
□	8.6.4.6, “Run the z/OS UNIX System Services Setup Verification Procedures” on page 213	
	8.6.5, “IVP jobs for Wave 1C” on page 213	

Figure 47 (Page 2 of 2). Checklist for the IVPs

Check Box	Section, Step Description	Your Notes
□	8.6.5.1, “Run the FFST Installation Verification Program” on page 213	
□	8.6.5.2, “Run the GDDM Installation Verification Procedures” on page 214	
□	8.6.5.3, “Run the ICKDSF Installation Verification Procedure” on page 220	
	8.6.6, “IVP jobs for Wave 1D” on page 221	
□	8.6.3.1, “Run the Communications Server IP Services Installation Verification Procedures” on page 193	
□	8.6.8.2, “Run the High Level Assembler Toolkit Installation Verification Program” on page 228	
	8.6.7, “IVP jobs for Wave 1E” on page 222	
□	8.6.7.1, “Run the Runtime Library Extensions Installation Verification Procedures” on page 222	
□	8.6.7.4, “Run the XL C/C++ Installation Verification Procedures” on page 222	
□	8.6.7.3, “Run the RMF Installation Verification Procedure” on page 222	
□	8.6.6.1, “Run the Security Server (RACF) Installation Verification Procedures” on page 222	
	8.6.8, “IVP jobs for Wave 1F” on page 224	
□	8.6.8.1, “Run the DFSORT Installation Verification Procedures” on page 224	
□	8.6.10.1, “Run the installation verification procedure for SDSF” on page 231	
	8.6.9, “IVP jobs for Wave 1G” on page 229	
□	8.6.9.1, “Run the z/OS File System Installation Verification Procedures” on page 229	
□	8.6.9.2, “Run the Infoprint Server Installation Verification Procedures” on page 229	
□	8.6.9.3, “Run the Network File System Installation Verification Procedures” on page 230	

8.6.2.1 IVP jobs for Wave 0

The following sections describe IVP jobs for Wave 0.

8.6.2.1.1 Run the High Level Assembler Installation Verification Procedure

1. Assemble the Verification Program

   HLASM provides a sample job, ASMWIVP, found in the SASMSAM1 library. This job is provided to verify the element has installed correctly. This job assembles the source statements in member ASMASAMP found in the SASMSAM1 library.

2. Run the Verification Program (optional)

   The program being assembled is not intended to be run; however, if it is run, it sets a return code of zero and returns to the caller.

8.6.2.2 Run the BCP and ISPF Installation Verification Procedure

Because the BCP and ISPF are used to verify some of the installation verification procedures in Wave 1A, the BCP and ISPF should be verified now rather than in 8.6.8, “IVP jobs for Wave 1F” on page 224.

To verify the BCP was installed properly, you should run IVPJOBS found in your SAMPLIB library. Refer to the comments in the job for instructions, expected condition codes, and expected output.

Note: The command statement labeled “LISTACTV” issues a Console Management “DISPLAY ACTIVE, LIST” COMMAND. It will also generate a WTO which should be responded to immediately. The system will not accept any new work until the operator has replied to the IEFC166D message.**:** The system will not process any more work until this system prompt is responded to.

8.6.2.2.1 ISPF

Log on to TSO using a LOGON procedure that includes the z/OS V2R5 ISPF data sets. The data sets are allocated to the appropriate DDs, as are documented in Figure 44 on page 149.

Enter `ISPSTART` *nlslang* at the TSO/E READY prompt (where *nlslang* is the language you are using). The IBM-supplied default is ENGLISH. The ISPF Primary Option Menu is displayed with the ISPF logo in a pop-up window. The logo contains a copyright date, as shown in Figure 48.

```
Licensed Materials - Property of IBM
5650-ZOS Copyright IBM Corp. 1980, 2021.
US Government Users Restricted Rights -
Use, duplication or disclosure restricted
by GSA ADP Schedule Contract with IBM Corp.
```

Figure 48. ISPF Copyright Logo

Enter the command `PANELID` on the Option line. The panel ID ISP@MSTR should appear in the upper left corner of the screen.

Enter `X` on the Option line to return to TSO/E READY.

Enter `ISPF` *nlslang* or `PDF` *nlslang* at the TSO/E READY prompt (where *nlslang* is the language you are using). The IBM-supplied default is ENGLISH.

The ISPF Primary Option Menu is displayed with the ISPF logo in a pop-up window. The logo contains a copyright date, as shown in Figure 48.

Execute the steps listed in Figure 49 to verify the basic ISPF functions have been properly installed. After entering the command `PANELID`, the panel name appears in the upper left corner of the screen.

Figure 49. Installation Verification Functions

In command line of panel...	Enter	Panel displayed
ISPF Primary Option	PANELID	ISR@PRIM
ISR@PRIM	0	ISPISMMN
ISPISMMN	=1	ISRBRO01
ISRBRO01	=2	ISREDM01
ISREDM01	=3.1	ISRUDA1
ISRUDA1	=3.2	ISRUDA2/ISRUDA2S*
Note: *ISRUDA2S if SMS is active.		
ISRUDA2/ISRUDA2S	=3.3	ISRUMC1
ISRUMC1	=3.4	ISRUDLP
ISRUDLP	=3.12	ISRSSNEW
ISRSSNEW	=3.13	ISRSEPRI
ISRSEPRI	=3.14	ISRSFSPR
ISRSFSPR	=4.1	ISRFP01
ISRFP01	=5.1	ISRJP01
ISRJP01	=6	ISRTSO/ISRTSOA
Note: ISRTSOA if requested by the configuration table.		
ISRTSO/ISRTSOA	=7	ISPYXD1
ISPYXD1	=X;9	ISRDIIS
ISRDIIS	=11	ISRWORK
ISRWORK	=10	FLMDMN

8.6.2.2.2 ISPF SCLM

This procedure allocates, assembles, and link-edits a test ISPF SCLM project and then uses this project to save and build an assembler source module. To execute this procedure, you need the following information:

- **A high-level qualifier to be used to define the temporary project definition.**

  This forms the prefix for the data sets allocated by this procedure. You can use your user ID as the high-level qualifier (*hlq*). Seven data sets will be allocated:
  - *hlq*.PROJDEFS.ACCTDB

  - *hlq*.PROJDEFS.BLDLIST
  - *hlq*.PROJDEFS.INFO
  - *hlq*.PROJDEFS.LOAD
  - *hlq*.PROJDEFS.OBJ
  - *hlq*.PROJDEFS.SOURCE
  - *hlq*.PROJDEFS.SYSPRINT

  These data sets must not previously exist. A minimum of nine 3390 DASD tracks are required for the allocation. At the end of the procedure you are given the option to delete these data sets.
- **The name of the SCLM macro library.**

  The SMP/E DDDEF for SISPMACS points to this data set.
- **A volume serial for a VSAM data set.**

  The volume used for the allocation of the *hlq*.PROJDEFS.ACCTDB data set.
- **The location of the High Level Assembler module ASMA90.**

  This location will be either in normal MVS search sequence (STEPLIB, LNKLST, LPA, etc.) or the High Level Assembler data set name (SASMMOD1).

After you have obtained the required information, follow these steps to perform the installation verification:

1. From the ISPF Primary Option Panel, select Option 10, SCLM.
2. From the SCLM panel, select Option 6, TSO or SCLM Commands.
3. Enter the command: `FLMIVP`
4. Answer the prompts with the information gathered during preparation.
5. At this point, the procedure will:

   - Allocate the necessary data sets. (If the data set allocation fails, diagnose the problem, take corrective action, and then rerun the FLMIVP procedure.)
   - Create, assemble, and link-edit the SCLM project definition
   - Test the installation by executing SAVE and BUILD commands
   - Indicate if the installation test passed or failed. If the test failed, the messages reported to the screen and the contents of the *hlq*.PROJDEFS.SYSPRINT and *hlq*.PROJDEFS.BLDLIST data sets can be used to determine the cause of the failure. If you cannot determine the cause of the failure, contact the IBM Support Center.
   - Prompt you to determine if the data sets should be deleted or retained.

   If errors occur during the SCLM installation verification procedure, use the FLMIVPC EXEC to delete the data sets allocated by FLMIVP.

8.6.3 IVP jobs for Wave 1A

There are currently no installation verification procedures for the following elements in Wave 1A:

- Communications Server for z/OS SNA Services
- TIOC
- Cryptographic Services ICSF
- System SSL

8.6.3.1 Run the Communications Server IP Services Installation Verification Procedures

Before beginning the installation verification procedures for Communications Server IP Services, SNA(VTAM) must be started in order for Communications Server IP Services to initialize successfully. To start VTAM from the MVS operator console, issue the following command:

```
S VTAM_procname,,,(LIST=start_list_id)
```

Expected results: The following messages will be displayed on the MVS operator console:

```
IST020I VTAM INITIALIZATION COMPLETE FOR CSV2R5
IST1349I COMPONENT ID IS 5695-11701-250
```

Before beginning the installation verification procedures for Communications Server IP Services, verify you have all the required values as listed beginning on page 161.

1. Set up TSO user's TCPIP.DATA

   On the TSO user ID from which the IVP will be executed, the SYSTCPD DD card of the TSO logon procedure must point to the data set specified by *TCPIP_DATA_name*. Either change the TSO logon procedure (and log off and on again), or do the following steps to override the SYSTCPD definition for the rest of the TSO logon session:

   - In TSO ready mode or from TSO option 6, drop the existing definition by entering the following command:

     ```
     FREE FI(SYSTCPD)
     ```

   - Establish the new SYSTCPD definition with this command:

     ```
     ALLOC FI(SYSTCPD) DS('TCPIP_DATA_name')
     ```

2. Start the SYSLOG daemon

   - From the MVS operator console, issue:

     ```
     S SYSLOGD_procname
     ```

   - Expected results:

     ```
     FSUM1248 SYSLOGD_procname INITIALIZATION COMPLETE
     ```

3. Start TCP/IP

   - From the MVS operator console, issue

```
S TCPIP_procname
```

- Expected results:

  The following messages will be displayed on the MVS operator console:

  ```
  EZZ4202I Z/OS UNIX - TCP/IP CONNECTION ESTABLISHED FOR TCPIP_procname

  EZB6473I TCP/IP STACK FUNCTIONS INITIALIZATION COMPLETE

  EZAIN11I ALL TCPIP SERVICES FOR PROC tcpproc_name ARE AVAILABLE

  EZD1314I TCP/IP AND EXTENDED SERVICES ARE NOW INITIALIZED FOR STACK:
  tcpstackname.
  ```

  Note: EZD1314I is newly issued by default in z/OS V2R5.

4. Start the FTP daemon

   - From the MVS operator console, issue

     ```
     S FTPD_procname
     ```

   - Expected results:
     - The *FTPD_procname* task will end after forking another task. The forked task (for example, FTPD1), will remain running.
     - The following message will be displayed on the MVS operator console:

       ```
       EZY2702I SERVER-FTP: INITIALIZATION COMPLETED AT hh:mm:ss ON mm/dd/yy
       ```

5. Test the following commands from TSO:

   - Issue:

     ```
     ping loopback
     ```

   - Expected results:

     ```
     CS VyRx: Pinging host LOOPBACK  (127.0.0.1).
     Ping #1 response took n.nnn seconds.
     ```

   - Issue

     ```
     netstat home
     ```

   - Expected results:

     ```
     MVS TCP/IP NETSTAT CS VyRx   TCPIP NAME:
     TCPIP_procname  hh:mm:ss

     Home address list:

     Address          Link             Flg

     -------          ----             ---

     127.0.0.1        LOOPBACK         P
     ```

Note: If you are using the sample profile without any changes, then you will not see the 'P' under the Flg header.

6. Set up environment variables required in the z/OS UNIX Shell

   - Set the following environment variables in the z/OS UNIX Shell for the user ID on which the IVP will be executed. From the z/OS UNIX shell, issue the following commands:

     ```
     export X_SITE="//'dyn_alc_hlq.HOSTS.SITEINFO'"
     export X_ADDR="//'dyn_alc_hlq.HOSTS.ADDRINFO'"
     export RESOLVER_CONFIG="//'TCPIP_DATA_name'"
     ```

7. Test commands from the z/OS UNIX shell

   - The following steps may require superuser authority to execute.
   - Ensure you have write access to your local directory. Then, in the z/OS UNIX shell, copy the verification shell script to your local directory. For example:

     ```
     cp /usr/lpp/tcpip/samples/ivp.scp /your_target_directory/ivp.scp
     ```

   - Run the shell script, directing output to a file.

     ```
     ivp.scp > ivp.out
     ```

   - The following tests are done by the shell script. The script will display the expected results and the actual results.

     ```
     oping loopback
     onetstat -h
     ```

   - Review the ivp.out file, ensuring the actual results match the expected results.

8. Test the following commands from the MVS operator console

   - Issue the following DISPLAY command:

     ```
     display tcpip,TCPIP_procname,netstat,home
     ```

   - Expected results:

     ```
     EZZ2500I NETSTAT CS VxRy TCPIP_procname

     HOME ADDRESS LIST:

     ADDRESS          LINK             FLG

     127.0.0.1        LOOPBACK         P

     1 OF 1 RECORDS DISPLAYED
     ```

9. Run the IVPFTP batch job to verify that FTP has been correctly set up

   - This job will get a sample file, FTPTESTP, from the *target_lib_hlq*.SEZAINST partitioned dataset and store it into the /tmp directory. It will then get the file from the /tmp directory hfs and store it in the *IVP_configuration_dsn* partitioned data set as FTPTESTG.

- Copy IVP test job to your chosen JCL dataset: `target_lib_hlq.SEZAINST(IVPFTP)` to `IVP_jcllib_dsn(IVPFTP)`
- Fill in any required parameters on the //JOB statement.
- Change the //SYSFTPD DD card to set the DSN variable to `IVP_configuration_dsn(FTCDATA)`, or accept the default of `'TCPIVP.TCPPARMS(FTCDATA)'` for the FTP client FTP.DATA file.
- Change the //SYSTCPD DD card to set the DSN variable to `TCPIP_DATA_name`, or accept the default of `'TCPIVP.TCPPARMS(TCPDATA)'`.
- Change the input values for *'userid password'* to match the TSO user ID and password to be used for the FTP client. For example:

  ```
  USER1 USER1PW
  ```

- Change the PUT statement to specify the appropriate SEZAINST dataset for sample dataset to be moved: 'tcpip' to *target_lib_hlq*.
- Change the GET statement to specify the appropriate location for the retrieved dataset: 'TCPIVP.TCPPARMS' to *IVP_configuration_dsn* (or accept the default of 'TCPIVP.TCPPARMS')
- Submit batch IVPFTP

  Expected results:
  - The job should complete with condition code zero.
  - The /tmp directory should contain a file called ftptestp.
  - The *IVP_configuration_dsn* should contain member FTPTESTG.
  - Both the new file and the new partitioned dataset member should contain the one line:

    ```
    The File Transfer Protocol is working!!!
    ```

8.6.3.2 Run the Language Environment Installation Verification Procedure: To verify that Language Environment is installed properly, run CEEWIVP in your SCEESAMP library. If the z/OS XL C/C++ compiler uses 64-bit virtual memory, ensure that MEMLIMIT > 0. The actual value of MEMLIMIT might vary with each customer. See the comments in the job for instructions, expected condition codes, and expected output.

Note: As of z/OS V1R8, the IPA Link step of the z/OS XL C/C++ compiler uses 64-bit virtual memory, which requires sufficient storage above the 2 GB bar (2 GB address line). For information about using MEMLIMIT to ensure that sufficient storage above the 2 GB bar is available for the IPA Link step, see 8.6.7.4, “Run the XL C/C++ Installation Verification Procedures” on page 222.

8.6.4 IVP jobs for Wave 1B

8.6.4.1 Run the DFSMS Installation Verification Procedure: To test the successful installation of DFSMS, run the following IVPs found in SAMPLIB:

```
DFPS1IVP  DFPS2IVP  DFPS3IVP  DFPX1IVP  DFPX2IVP  DFPX3IVP
```

Note: The members with the prefix 'DFPS' use SMS managed volumes and datasets. The members with the prefix 'DFPX' use non-SMS managed volumes and datasets.

8.6.4.1.1 DFSMSdfp OAM Installation Verification Procedure: It is recommended that you run the OAM Installation Verification Program (CBRSAMIV) at initial installation and at migration. CBRSAMIV verifies that OAM object support is successfully installed and operational. It ensures that the proper environment exists for applications to utilize the OSREQ Macro Application Programming Interface (API).

For detailed information about how to run the IVP program (CBRSAMIV) and invoke the OSREQ API, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

8.6.4.2 DFSMSrmm Installation Verification Procedures

DFSMSrmm Samples Provided in SAMPLIB

- EDGIVPPM Sample Parmlib for Use in the IVP
- EDGIVP1 IVP Job 1 to Initialize Tape Volumes
- EDGIVP2 IVP Job 2 to Use Tape Volumes

This topic helps you prepare for and run the DFSMSrmm installation verification procedures (IVP). You can use the IVP to ensure that the DFSMSrmm functional component has been successfully installed by SMP/E and can be activated on your system. The IVP does not test all the functions in DFSMSrmm but validates that the key interfaces are in place.

8.6.4.2.1 Preparing to run the IVP: Before you run the IVP, you need to activate some of DFSMSrmm's functions. This topic lists and describes the steps you should perform to set up DFSMSrmm for the IVP. Chapter 2. Implementing DFSMSrmm in *z/OS DFSMSrmm Implementation and Customization Guide* contains all the steps needed to install DFSMSrmm.

If this is first time you are setting up DFSMSrmm, follow all the steps described in this topic. If your system has previously been set up for use with DFSMSrmm, you might not need to perform all the steps listed here. Evaluate your installation setup to determine which steps you can omit.

1. Install DFSMSrmm with SMP/E.

   Ensure that DFSMS including DFSMSrmm is SMP/E applied.

   Once you have used SMP/E to install DFSMSrmm, IPL your system without performing any implementation tasks and have DFSMSrmm take no part in removable media management. The ability to run without DFSMSrmm is especially helpful if you are running another tape management product in production.

2. Update SYS1.PARMLIB members.

   For detailed instructions, see "Step 5: Updating SYS1.PARMLIB Members" in *z/OS DFSMSrmm Implementation and Customization Guide*. At a minimum, you should update IEFSSN*xx*. Also update IFAPRD*xx* member in SYS1.PARMLIB to enable DFSMSrmm".

3. Update the procedure library.

Refer to "Step 8: Updating the Procedure Library" in *z/OS DFSMSrmm Implementation and Customization Guide* for detailed instructions. Use member EDGDFRMM of SYS1.SAMPLIB as a sample DFSMSrmm procedure.

4. Assign DFSMSrmm a RACF user ID.

   Perform this step if you want to use a specific RACF user ID for DFSMSrmm during the IVP. When running on a system with RACF installed, assign DFSMSrmm a RACF user ID by adding a profile in the STARTED class as described in "Step 9: Assigning DFSMSrmm a RACF User ID" in *z/OS DFSMSrmm Implementation and Customization Guide*. You can use the DFSMSrmm procedure name that you created in Step 3 as the RACF user ID but any installation-selected RACF user ID is acceptable. As data sets are created for use by the DFSMSrmm procedure, add the RACF user ID to the access list for the data sets. "Table 6. Data Sets Requiring Access by the DFSMSrmm RACF User ID" in *z/OS DFSMSrmm Implementation and Customization Guide* lists the data sets that the DFSMSrmm procedure should be able to access.

   If you are using an equivalent security product, review the RACF-related information to determine the changes that might be required to run DFSMSrmm with that product.

5. Define parmlib member EDGRMM*xx*.

   Refer to "Step 10: Defining Parmlib Member EDGRMMxx" in *z/OS DFSMSrmm Implementation and Customization Guide* for detailed instructions. Use member EDGIVPPM of SYS1.SAMPLIB as a sample parmlib member.

6. Specify DFSMSrmm options.

   Refer to *z/OS DFSMSrmm Implementation and Customization Guide* for information on tailoring the DFSMSrmm sample parmlib member EDGIVPPM to specify DFSMSrmm options for the IVP.

   During the IVP, DFSMSrmm runs in record-only mode. DFSMSrmm records information about tape volumes, but does no validation. You can tailor EDGIVPPM to specify that DFSMSrmm run in warning mode or protect mode if you want DFSMSrmm to validate volumes.

   If you are running the IVP on a system with no other tape management system you can select any mode: record-only, warning or protect. If there is a possibility of accidental use of the wrong tape volumes, we suggest that you run in protect mode. However, if you run the IVP on a system where others are using tape including the use of scratch tapes, be aware that DFSMSrmm rejects all scratch tapes not defined to it while running in protect mode. See "Defining System Options: OPTION" in *z/OS DFSMSrmm Implementation and Customization Guide* for information about DFSMSrmm modes of operation.

7. Create the DFSMSrmm control data set.

   Refer to *z/OS DFSMSrmm Implementation and Customization Guide* for detailed instructions. You can use the sample JCL in member EDGJMFAL in SYS1.SAMPLIB to allocate a control data set. Ensure that the control data set name is the same as that specified in the parmlib member EDGRMMxx that you created. Initialize the control data set by running the EDGUTIL utility. You can use the sample JCL in member EDGJUTIL in SYS1.SAMPLIB. Set the rack and bin count fields to 0.

8. Create the journal.

Refer to *z/OS DFSMSrmm Implementation and Customization Guide* for detailed instructions. You can use sample JCL in member EDGJNLAL in SYS1.SAMPLIB to allocate a journal data set.

9. Make the DFSMSrmm ISPF Dialog available to users.

   Refer to *z/OS DFSMSrmm Implementation and Customization Guide* for detailed instructions.

10. Restart z/OS with DFSMSrmm implemented.

    You are ready to start the system with DFSMSrmm implemented. Refer to *z/OS DFSMSrmm Implementation and Customization Guide* for information on conditions that determine if you need to IPL the system to restart z/OS with DFSMSrmm implemented. Perform this step so that the changes you made to IEFSSNxx and other parmlib members when you performed Step 2 take effect.

11. Start DFSMSrmm.

    Refer to "Step 19: Starting DFSMSrmm" for detailed instructions. When you start DFSMSrmm, if it issues message EDG0103D, reply 'RETRY'. If you do not reply 'RETRY', DFSMSrmm will not record any tape usage activity.

DFSMSrmm is activated and you are ready to run the IVP.

8.6.4.2.2 Running the IVP: To run the IVP, perform these steps:

1. You need three tape volumes that do not have any data on them, and a single tape unit online to your system. Ask your tape librarian to externally label these volumes EDG000, EDG001, and EDG002 for your testing.

   Ensure that the tape volumes you use are suitable for use with DFSMSrmm during the IVP. For example, if you have an existing tape management system, check that the volumes are either not managed by it or are designated for use with DFSMSrmm for testing.

2. Ensure that TSO help information has been correctly installed by entering this command from a TSO terminal:

   ```
   HELP RMM
   ```

   DFSMSrmm lists help information for the RMM TSO subcommand, including a list of subcommands, function, syntax, and operands.

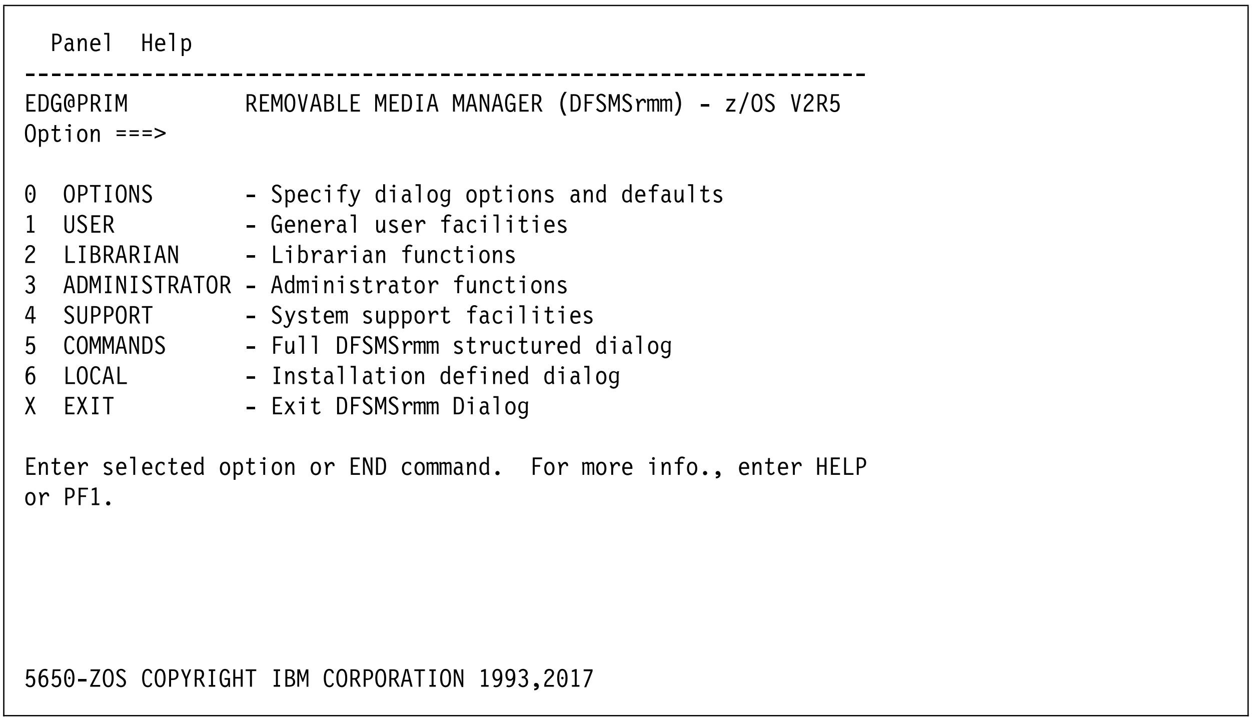
3. Add some shelf locations to DFSMSrmm by entering this RMM TSO command from a TSO terminal:

   ```
   RMM ADDRACK RMM000 COUNT(10)
   ```

4. Add some volumes to DFSMSrmm by using the DFSMSrmm ISPF dialog. Enter this command from a TSO terminal:

   ```
   %RMMISPF
   ```

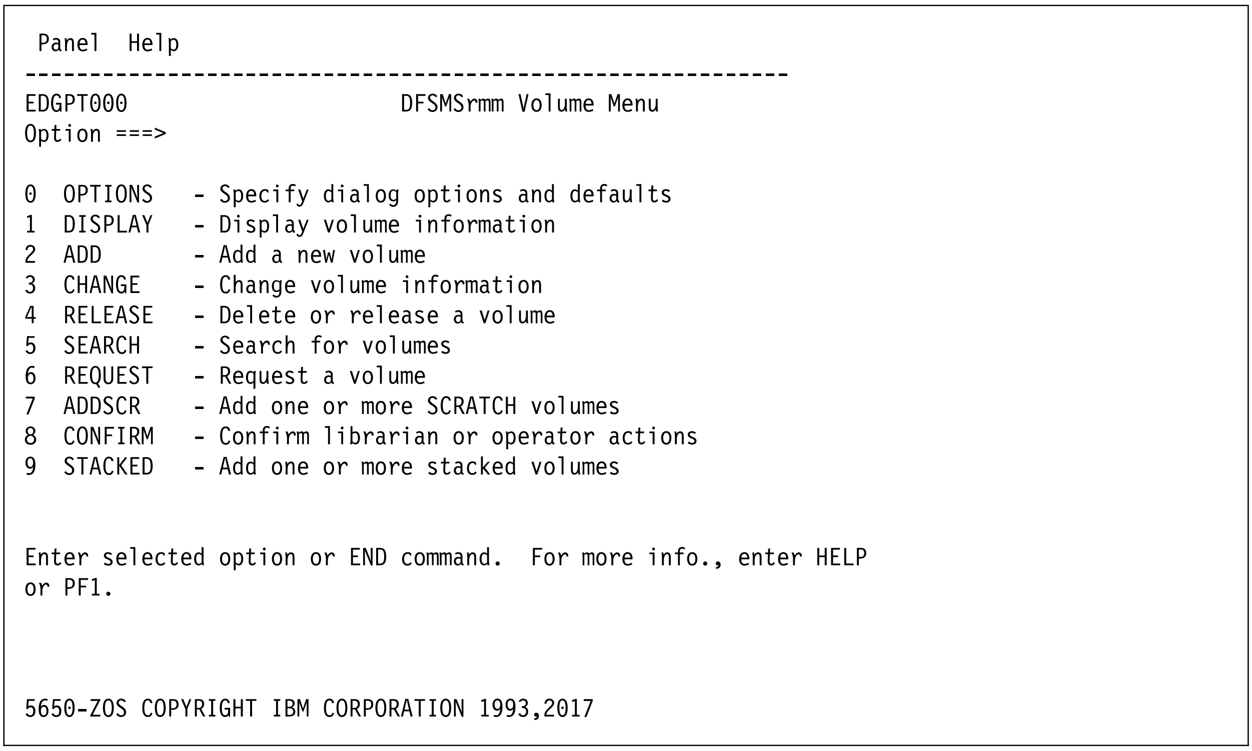
   DFSMSrmm displays the DFSMSrmm ISPF dialog primary option menu as shown in Figure 50 on page 200.



```
  Panel  Help
-----------------------------------------------------------------
EDG@PRIM         REMOVABLE MEDIA MANAGER (DFSMSrmm) - z/OS V2R5
Option ===>

0  OPTIONS       - Specify dialog options and defaults
1  USER          - General user facilities
2  LIBRARIAN     - Librarian functions
3  ADMINISTRATOR - Administrator functions
4  SUPPORT       - System support facilities
5  COMMANDS      - Full DFSMSrmm structured dialog
6  LOCAL         - Installation defined dialog
X  EXIT          - Exit DFSMSrmm Dialog

Enter selected option or END command.  For more info., enter HELP
or PF1.





5650-ZOS COPYRIGHT IBM CORPORATION 1993,2017
```

Figure 50. DFSMSrmm Primary Option Menu

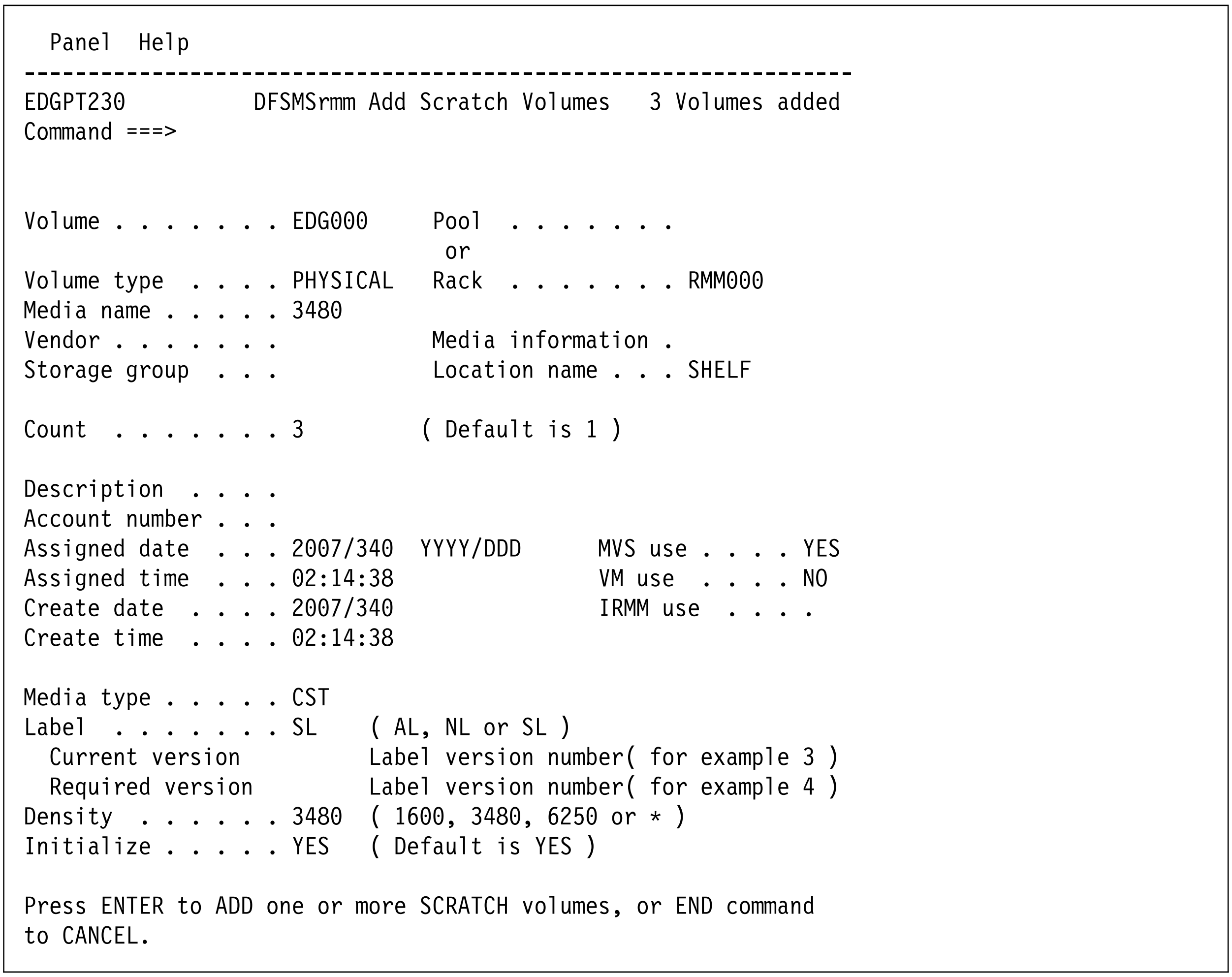
Enter VOLUME on the option line to display the DFSMSrmm Volume Menu, as shown in Figure 50. DFSMSrmm displays a panel, as shown in Figure 51 on page 201.



```
 Panel  Help
------------------------------------------------------------
EDGPT000                     DFSMSrmm Volume Menu
Option ===>

0  OPTIONS   - Specify dialog options and defaults
1  DISPLAY   - Display volume information
2  ADD       - Add a new volume
3  CHANGE    - Change volume information
4  RELEASE   - Delete or release a volume
5  SEARCH    - Search for volumes
6  REQUEST   - Request a volume
7  ADDSCR    - Add one or more SCRATCH volumes
8  CONFIRM   - Confirm librarian or operator actions
9  STACKED   - Add one or more stacked volumes


Enter selected option or END command.  For more info., enter HELP
or PF1.



5650-ZOS COPYRIGHT IBM CORPORATION 1993,2017
```

Figure 51. DFSMSrmm Volume Menu

Select option 7, ADDSCR, and press ENTER. DFSMSrmm displays the DFSMSrmm Add Scratch Volumes panel shown in Figure 52 on page 202. Complete the details as shown in the panel and press ENTER:



```
  Panel  Help
------------------------------------------------------------------
EDGPT230         DFSMSrmm Add Scratch Volumes   3 Volumes added
Command ===>


Volume . . . . . . . EDG000     Pool  . . . . . . .
                                 or
Volume type  . . . . PHYSICAL   Rack  . . . . . . . RMM000
Media name . . . . . 3480
Vendor . . . . . . .            Media information .
Storage group  . . .            Location name . . . SHELF

Count  . . . . . . . 3         ( Default is 1 )

Description  . . . .
Account number . . .
Assigned date  . . . 2007/340  YYYY/DDD      MVS use . . . . YES
Assigned time  . . . 02:14:38                VM use  . . . . NO
Create date  . . . . 2007/340                IRMM use  . . . .
Create time  . . . . 02:14:38

Media type . . . . . CST
Label  . . . . . . . SL    ( AL, NL or SL )
  Current version          Label version number( for example 3 )
  Required version         Label version number( for example 4 )
Density  . . . . . . 3480  ( 1600, 3480, 6250 or * )
Initialize . . . . . YES   ( Default is YES )

Press ENTER to ADD one or more SCRATCH volumes, or END command
to CANCEL.
```

Figure 52. DFSMSrmm Add Scratch Volumes Panel

DFSMSrmm displays the message `3 volumes added` in the right hand corner of the screen.

Exit the DFSMSrmm ISPF dialog by entering **=X** on the command line.

5. Initialize tape volumes by editing and submitting the JCL in member EDGIVP1 in SYS1.SAMPLIB. Mount the three tape volumes requested by this job in the sequence EDG002, EDG001, and EDG000.

   Ensure that the job completes with a return code of zero and the expected messages in EDGIVP1 are in the job output.

6. Write data to tape volumes by editing and submitting the JCL in member EDGIVP2 in SYS1.SAMPLIB. Mount the three tape volumes requested by this job in the sequence EDG000, EDG001, and EDG002. Use the three volumes initialized in Step 5.

Ensure that all steps of the job complete with a return code of zero. Message IEC502E is issued when the job finishes with the second volume, EDG001. Check that the message in the SYSLOG contains the text RACK=RMM001 on the right hand side as follows:

```
IEC502E RK ddd,EDG001,SL,jjjjjjjj,WRITE22 - RACK=RMM001
```

7. To display data set information that is recorded by DFSMSrmm, enter these RMM TSO subcommands:

```
RMM LISTDATASET 'RMMIVP.TEST1' VOLUME(EDG000) SEQ(1)
RMM LISTDATASET 'RMMIVP.TEST2' VOLUME(EDG000) SEQ(2)
RMM LISTDATASET 'RMMIVP.TEST3' VOLUME(EDG001) SEQ(1)
RMM LISTDATASET 'RMMIVP.TEST4' VOLUME(EDG001) SEQ(2)
RMM LISTDATASET 'RMMIVP.TEST4' VOLUME(EDG002) SEQ(1)
```

   DFSMSrmm displays data set information, as shown in Figure 53 on page 204.

```
Data set name = RMMIVP.TEST1
Volume        = EDG000         Physical file sequence number = 1
Owner         = TEST                       Data set sequence = 1
Create date   = 2015/089 Create time = 12:50:01 System ID = TEST
Expiration date       = 2015/094          Expir. time = 12:50:01
         set by       = OCE_DEF     Original expir.date  =
LASTREF Extra Days    = 0          WHILECATALOG          = OFF
Block size            = 80         Block count           = 10
Data set size(KB)     = 1
Physical size(KB)     = 0          Compression           = 0.00
Percent of volume     = 0          Total block count     = 10
Logical Record Length = 80         Record Format         = FB
Date last written     = 2015/089   Date last read     = 2015/089
Job name              = EDGIVP2    Last job name      = EDGIVP2
Step name             = WRITE11    Last step name     = WRITE11
Program name          = IEBDG      Last program name     = IEBDG
DD name               = SEQOUT     Last DD name          = SEQOUT
Device number         = 0B50       Last Device number    = 0B50
Management class      =            VRS management value  =
Storage group         =            VRS retention date    =
Storage class         =            VRS retained          = NO
Data class            =            Closed by Abend       = NO
                                   Deleted               = NO
VRSEL exclude         = NO         Catalog status        = YES
Primary VRS details:
       Name           =
       Job name       =            Type                  =
       Subchain NAME  =            Subchain start date   =
Secondary VRS details:
       Value or class =
       Job name       =
       Subchain NAME  =            Subchain start date   =
Security Class   =            Description   =
BES key index    = 0

Last Change information:
Date             = 2015/089    Time = 12:50:01  System  = 3090
User change date =             Time =           User ID = *OCE
```

Figure 53. Sample Data Set Information

To cleanup after running the IVP or to prepare to rerun the IVP, issue these commands to remove information from the DFSMSrmm control data set.

```
RMM DELETEVOLUME EDG000 FORCE
RMM DELETEVOLUME EDG001 FORCE
RMM DELETEVOLUME EDG002 FORCE
RMM DELETERACK RMM000 COUNT(10)
```

When you have completed running the IVP, you can return the three volumes to your tape library.

8.6.4.3 DFSMShsm Installation Verification Procedures: The DFSMShsm installation verification procedure (IVP) is an optional procedure that verifies that the DFSMShsm product is correctly installed and can be started and stopped using a minimum of DASD resources.

When the System Modification Program/Extended (SMP/E) installs the DFSMShsm product modules onto an MVS system, it places the ARCIVPST member into SYS1.SAMPLIB. This ARCIVPST member is the job that you run for the IVP procedure.

Note: If you have not performed an initial program load for your MVS system with the CLPA (create link pack area) option, do it before you run DFSMShsm.

8.6.4.3.1 Setup requirements: The following information is required to run the DFSMShsm IVP:

- The catalog, with its associated alias, which must be defined before attempting to run the IVP.
- The name of the IVP user ID.
- A high-level qualifier for the data sets that are required by the IVP.
- A volume serial number and unit type of a device containing one free cylinder for a temporary user catalog.
- A name for a temporary user catalog for the migration control data set (MCDS).
- The name of the system master catalog.
- The job control parameters for each job. (Usually this is defined as HSM.)
- The volume serial number and unit type of a device containing one free cylinder for a temporary MCDS.
- The version of JES (either JES2 or JES3) on the system.

  Note: In an SMS environment, no consideration is given to whether data sets allocated by the IVP are SMS or non-SMS managed.

8.6.4.3.2 Steps for running the Installation Verification Procedure: The following steps are used to prepare for and run the DFSMShsm installation verification procedure:

1. Edit the job ARCIVPST inserting the correct parameters in the job statement. See "HSMIVP" in *z/OS DFSMShsm Implementation and Customization Guide* for a listing of ARCIVPST.

   ARCIVPST is an IEBUPDTE job, HSMIVP, that creates the partitioned data set DFSMSHSM.IVP.CNTL. Be sure to change the job control statement before running this job. For an example listing of the HSMIVP job, refer to the samples provided in SYS1.SAMPLIB.

2. After the edit, run the job in ARCIVPST. The result of this job is a partitioned data set named DFSMSHSM.IVP.CNTL, containing the member HSMIVP1. The partitioned data set is allocated on one cylinder of SYSDA.

3. Edit member HSMIVP1 in the DFSMSHSM.IVP.CNTL data set. Globally change the HSMIVP1 parameters by substituting the values determined in 8.6.4.3.1, “ Setup requirements.” For a listing of HSMIVP1 parameters, see "HSMIVP1" in *z/OS DFSMShsm Implementation and Customization Guide*.

   This job allocates data sets on your system for the DFSMShsm IVP. For an example listing of the HSMIVP1 job, refer to the samples provided in SYS1.SAMPLIB. Note that you must ensure that you globally change the following values before submitting this job.

?UCATUNIT Defines a unit type for the temporary user catalog.

?UCATVOL Defines the volume serial number of the volume for the temporary user catalog.

?UCATNAM Defines the name of the temporary user catalog for the MCDS.

?MCATNAM Defines the name and password of the system master catalog.

?JOBPARM Defines the job control parameters that are used for each job.

?HIQUAL Defines the high-level qualifier for the data sets used by the IVP. Usually this is defined as HSM. This value must be between 1 and 7 characters; the first character must be alphabetic. This value must be different from the value that is specified for the high-level qualifier on the control data set. Ensure that this IVP parameter is unique so that it does not conflict with any other alias currently in use.

?CDSUNIT Defines a unit type for the temporary MCDS.

?CDSVOL Defines the volume serial number of the volume for the temporary MCDS.

?JESVER Defines the version of JES on the system as either JES2 or JES3.

4. Run the HSMIVP1 job to create the following:

   - A user catalog
   - An MCDS VSAM data set
   - A startup procedure DFSMSHSM in SYS1.PROCLIB

     Note: If a startup procedure named DFSMSHSM already exists in the SYS1.PROCLIB data set, it will be overwritten.
   - Members ARCCMD00, HSMIVP2, and UCLINCHK in the partitioned data set DFSMSHSM.IVP.CNTL

     Rule: If the HSMIVP1 job is submitted while you are editing member HSMIVP1, you must exit the edit so that the HSMIVP1 job can update the partitioned data set DFSMSHSM.IVP.CNTL.

5. After job HSMIVP1 has completed, start DFSMShsm by typing the command **S DFSMSHSM** from the system console.

6. Review the console messages created as IVP output. Scan the console for messages ARC0001I and ARC0008I informing you that DFSMShsm has started and that initialization has completed. See Figure 54 on page 208 for an example of the console screen.

   Note that you can expect to see error messages that are issued because many data sets are not allocated until later in the DFSMShsm installation when the starter set jobs are run. For a description of the messages, use LookAt or see *MVS System Messages*.

7. Run job UCLINCHK.

   The job attempts actions which fail if DFSMShsm is correctly installed; You may receive the following messages

   ```
   ARC1001I HSMIVP.UCLINCHK  RECALL FAILED, RC=0002, REAS=0000
   ARC1102I DATA SET IS NOT MIGRATED/BACKED UP
   ```

You may also receive other error messages during the IDCAMS job, depending on your system environment:

```
ALLOCATE DA('HSMIVP.UCLINCHK')
IKJ56238I DATA SET HSMIVP.UCLINCHK NOT ALLOCATED, UNKNOWN ERROR DURING
RECALL
IGD04001I UNEXPECTED CATALOG LOCATE PROCESSING ERROR - RETURN CODE 38
REASON CODE 4
IDC3003I FUNCTION TERMINATED. CONDITION CODE IS 12
```

Note: A request to mount volume "MIGRAT" is an error condition that indicates that UCLIN has not run or has not run correctly.

Do not progress to the next step until the UCLINCHK job has run with the result described in Step 7.

```
S DFSMSHSM
$HASP100 DFSMSHSM ON STCINRDR
IEF695I START DFSMSHSM WITH JOBNAME DFSMSHSM IS ASSIGNED TO USER IBMUSER
 , GROUP SYS1
$HASP373 DFSMSHSM STARTED
ARC0041I MEMBER ARCSTR00 USED IN DFSMSHSM.IVP.CNTL
ARC0036E I/O DISABLED FOR DFSMSHSM PROBLEM 310
ARC0036E (CONT.) DETERMINATION OUTPUT DATA SET, REAS= 1
ARC0020I ******************************
ARC0036E I/O DISABLED FOR DFSMSHSM PROBLEM 312
ARC0036E (CONT.) DETERMINATION OUTPUT DATA SET, REAS= 2
ARC0021I  DFSMSHSM LOGGING FUNCTION DISABLED
ARC0020I ******************************
ARC0035E DFSMSHSM JOURNAL IS PERMANENTLY DISABLED, 315
ARC0035E (CONT.) REASON=8
ARC0860E JOURNAL SPACE MONITORING DISABLED - RC=24. 316
ARC0860E (CONT.) MIGRATION, BACKUP, FRBACKUP, DUMP, AND RECYCLE HELD.
ARC0001I DFSMSHSM 2.5.0 STARTING HOST=1 IN 317
ARC0001I (CONT.) HOSTMODE=MAIN
IEC130I BAKCAT   DD STATEMENT MISSING
ARC0945I OPEN OF DDNAME=BAKCAT FAILED, VSAM REASON 319
ARC0945I (CONT.) CODE IS X'80'
ARC0134I BACKUP CONTROL DATA SET NOT OPENED, BACKUP 320
ARC0134I (CONT.) WILL NOT BE ENABLED
IEC130I OFFCAT   DD STATEMENT MISSING
ARC0945I OPEN OF DDNAME=OFFCAT FAILED, VSAM REASON 322
ARC0945I (CONT.) CODE IS X'80'
ARC0133I OFFLINE CONTROL DATA SET NOT OPENED, TAPE 323
ARC0133I (CONT.) SUPPORT WILL NOT BE ACTIVE
ARC6374E BCDS IS NOT DEFINED, AGGREGATE BACKUP AND 324
ARC6374E (CONT.) RECOVERY FUNCTIONS ARE DISABLED
ARC1700I DFSMSHSM COMMANDS ARE RACF PROTECTED
ARC0041I MEMBER ARCCMD00 USED IN DFSMSHSM.IVP.CNTL
ARC0100I SETSYS COMMAND COMPLETED
ARC0101I QUERY SETSYS COMMAND STARTING ON HOST=1
ARC0147I BUDENSITY=*, BUUNIT=3590-1, BU RECYCLE 355
ARC0147I (CONT.) PERCENTAGE=020%, MOUNT WAIT TIME=015 MINUTE(S),
ARC0147I (CONT.) TAPESPANSIZE(0500)
ARC0419I SELECTVOLUME=SCRATCH, 356
ARC0419I (CONT.) TAPEDELETION=SCRATCHTAPE, PARTIALTAPE=REUSE,
ARC0419I (CONT.) DISASTERMODE=NO
ARC0259I TAPEDATASETORDER=PRIORITY
ARC0408I INPUT TAPE ALLOCATION=NOWAIT, OUTPUT TAPE 358
ARC0408I (CONT.) ALLOCATION=NOWAIT, RECYCLE TAPE ALLOCATION=NOWAIT,
ARC0408I (CONT.) TAPEFORMAT=SINGLEFILE
```

Figure 54 (Part 1 of 4). Example of a z/OS V2R5 DFSMShsm Startup Screen (IVP)

```
ARC0417I TAPE INPUT PROMPT FOR BACKUPTAPES=YES
ARC0417I TAPE INPUT PROMPT FOR DUMPTAPES=YES
ARC0417I TAPE INPUT PROMPT FOR MIGRATIONTAPES=YES
ARC0442I TAPE OUTPUT PROMPT FOR TAPECOPY=NO, DUPLEX 362
ARC0442I (CONT.) BACKUP TAPES=NO, DUPLEX MIGRATION TAPES=NO
ARC0410I TAPEMIGRATION=NONE(ROUTETOTAPE(ANY)), 363
ARC0410I (CONT.) MIGDENSITY=*, MIGUNIT=3590-1, ML2 RECYCLE
ARC0410I (CONT.) PERCENTAGE=020%, TAPEMAXRECALLTASKS=01, ML2 PARTIALS
ARC0410I (CONT.) NOT ASSOCIATED GOAL=010, RECONNECT(NONE)
ARC0444I CLOUDMIGRATION RECONNECT(NONE)
ARC0411I TAPESECURITY=PASSWORD, DEFERMOUNT
ARC0412I RECYCLEOUTPUT BACKUP=**NONE**, 366
ARC0412I (CONT.) MIGRATION=**NONE**, RECYCLETAKEAWAYRETRY=(NO,
ARC0412I (CONT.) MAXRETRYATTEMPTS=12, DELAY=0300)
ARC0840I MAXRECYCLETASKS=02, RECYCLE INPUT 367
ARC0840I (CONT.) DEALLOCATION FREQUENCY BACKUP=000 MIGRATION=000
ARC0149I MONITOR STARTUP NOSPACE NOVOLUME, MCDS(080), 368
ARC0149I (CONT.) BCDS(  0), OCDS(  0), JOURNAL(  0)
ARC0150I JOURNAL=NONE, LOG=NO, TRACE=NO, SMFID=NONE, 369
ARC0150I (CONT.) DEBUG=NO, EMERG=NO, JES=2, SYS1DUMP=YES, RACFIND=YES,
ARC0150I (CONT.) ERASEONSCRATCH=NO, PDA=ON, DSSXMMODE=(BACKUP=NO,
ARC0150I (CONT.) CDSBACKUP=NO, DUMP=NO, MIGRATION=NO, RECOVERY=NO)
ARC0151I DAYS=001, ML1DAYS=060, 370
ARC0151I (CONT.) PRIMARYSPMGMTSTART=(0000 NONE),
ARC0151I (CONT.) MAXMIGRATIONTASKS=0002, INTERVALMIGRATION=YES,
ARC0151I (CONT.) MIGRATIONCLEANUPDAYS(0010 0030 0003), SDSP=NONE,
ARC0151I (CONT.) MIGRATION PREFIX=IBMUSER, SCRATCH EXPIRED DATA
ARC0151I (CONT.) SETS=NO, SECONDARYSPMGMTSTART=(0000 NONE)
ARC0267I MIGRATIONSUBTASKS=NO, ADDITIONALMIGSUBTASKS=**
ARC0272I PRIMARY SPACE MGMT CYCLE LENGTH=00 DAYS, 372
ARC0272I (CONT.) CYCLE=*NONE*, TODAY IS DAY=0, CYCLE START
ARC0272I (CONT.) DATE=00/00/00
ARC0272I SECONDARY SPACE MGMT CYCLE LENGTH=00 DAYS, 373
ARC0272I (CONT.) CYCLE=*NONE*, TODAY IS DAY=0, CYCLE START
ARC0272I (CONT.) DATE=00/00/00, ML1OVERFLOW(DATASETSIZE=002000000K,ARC02
ARC0139I MAXINTERVALTASKS=02, ONDEMANDMIGRATION=NO, 374
ARC0139I (CONT.) ODMNOTIFICATIONLIMIT=00100,
ARC0139I (CONT.) MAXSSMTASKS(TAPEMOVEMENT=01, CLEANUP=02)
ARC0374I ACCEPTPSCBUSERID=NO
ARC0152I MAXRECALLTASKS=15, 376
ARC0152I (CONT.) RECALL=ANYSTORAGEVOLUME(LIKE), MAXEXTENTS=00,
ARC0152I (CONT.) CONVERSION=NO, VOLCOUNT=*NONE*,
ARC0152I (CONT.) TAPERECALLLIMITS(TASK=00015, TAPE=00020)
ARC0153I SCRATCHFREQ=9999, SYSOUT(CLASS=A, COPIES=01, 377
ARC0153I (CONT.) SPECIAL FORMS=NONE), SWAP=YES, PERMISSION=NO,
```

Figure 54 (Part 2 of 4). Example of a z/OS V2R5 DFSMShsm Startup Screen (IVP)

```
ARC0153I (CONT.) EXITS=NONE, UNLOAD=NO, DATASETSERIALIZATION=DFHSM,
ARC0153I (CONT.) USECMS=NO
ARC0418I TAPEUTILIZATION PERCENT=0097, LIBRARYMIGRATION
ARC0418I TAPEUTILIZATION PERCENT=0097, LIBRARYBACKUP
ARC0418I TAPEUTILIZATION PERCENT=0097, UNIT=3480 380
ARC0418I (CONT.) CAPACITYMODE=**NONE**
ARC0418I TAPEUTILIZATION PERCENT=0097, UNIT=3480X 381
ARC0418I (CONT.) CAPACITYMODE=**NONE**
ARC0418I TAPEUTILIZATION PERCENT=0097, UNIT=3490 382
ARC0418I (CONT.) CAPACITYMODE=**NONE**
ARC0418I TAPEUTILIZATION PERCENT=0097, UNIT=3590-1 383
ARC0418I (CONT.) CAPACITYMODE=**NONE**
ARC0638I MAXDUMPTASKS=02, ADSTART=(0000 0000 0000), 384
ARC0638I (CONT.) DUMPIO=(1,4), VOLUMEDUMP=(STANDARD),
ARC0638I (CONT.) MAXDUMPRECOVERTASKS=01
ARC0274I BACKUP=NO, SPILL=NO, MAXDSRECOVERTASKS=02, 385
ARC0274I (CONT.) MAXDSTAPERECOVERTASKS=02
ARC0154I MAXBACKUPTASKS=02, ABSTART= (0000 0000 0000), 386
ARC0154I (CONT.) VERSIONS=002, FREQUENCY=000, SKIPABPRIMARY=NO, BACKUP
ARC0154I (CONT.) PREFIX=IBMUSER, INCREMENTALBACKUP=ORIGINAL,
ARC0154I (CONT.) PROFILEBACKUP=YES, INUSE=(RETRY=NO, DELAY=015,
ARC0154I (CONT.) SERIALIZATION=REQUIRED)
ARC0269I DS DASD BACKUP TASKS=02, DS TAPE BACKUP 387
ARC0269I (CONT.) TASKS=02, DEMOUNTDELAY=0060, MAXIDLETASKS=00, DS
ARC0269I (CONT.) BACKUP MAX DASD SIZE=000003000, DS BACKUP STD DASD
ARC0269I (CONT.) SIZE=000000250, SWITCHTAPES TIME=0000,
ARC0269I (CONT.) PARTIALTAPE=MARKFULL, GENVSAMCOMPNAMES=YES
ARC1823I MAXCOPYPOOL (FRBACKUP TASKS=0015, FRRECOV 388
ARC1823I (CONT.) TASKS=0015, DSS TASKS=0024),
ARC1823I (CONT.) FASTREPLICATION(DATASETRECOVERY=NONE
ARC1823I (CONT.) FCRELATION=EXTENT VOLUMEPAIRMESSAGES=NO
ARC1823I (CONT.) MESSAGEDATASET(NO HLQ=HSMMSG))
ARC0375I CDSVERSIONBACKUP, 389
ARC0375I (CONT.) MCDSBACKUPDSN=IBMUSER.MCDS.BACKUP,
ARC0375I (CONT.) BCDSBACKUPDSN=IBMUSER.BCDS.BACKUP,
ARC0375I (CONT.) OCDSBACKUPDSN=IBMUSER.OCDS.BACKUP,
ARC0375I (CONT.) JRNLBACKUPDSN=IBMUSER.JRNL.BACKUP
ARC0376I BACKUPCOPIES=0004, BACKUPDEVICECATEGORY=TAPE 390
ARC0376I (CONT.) UNITNAME=3590-1, DENSITY=*, RETPD=0007, NOPARALLEL,
ARC0376I (CONT.) LATESTFINALQUALIFIER=V0000000, DATAMOVER=HSM
ARC0203I CSALIMITS=YES, CSA CURRENTLY USED=0 BYTES, 391
ARC0203I (CONT.) MWE=0, MAXIMUM=0K BYTES, ACTIVE=0%, INACTIVE=0%
ARC0340I COMPACTION OPTIONS ARE: TAPEMIGRATION=NO, 392
ARC0340I (CONT.) DASDMIGRATION=NO, TAPEBACKUP=NO, DASDBACKUP=NO,
ARC0340I (CONT.) TAPEHARDWARECOMPACT=NO, ZCOMPRESS OPTIONS ARE:
```

Figure 54 (Part 3 of 4). Example of a z/OS V2R5 DFSMShsm Startup Screen (IVP)

```
ARC0340I (CONT.) TAPEMIGRATE=NO, DASDMIGRATE=NO, TAPEBACKUP=NO,
ARC0340I (CONT.) DASDBACKUP=NO
ARC0341I COMPACT PERCENT IS 40%
ARC0339I OPTIMUMDASDBLOCKING=YES, LOGGING LEVEL=FULL, 394
ARC0339I (CONT.) LOG TYPE=SYSOUT A
ARC6008I AGGREGATE BACKUP/RECOVERY PROCNAME = DFHSMABR
ARC6009I AGGREGATE BACKUP/RECOVERY MAXADDRESSSPACE = 01
ARC6366I AGGREGATE BACKUP/RECOVERY UNIT NAME = 3590-1
ARC6368I AGGREGATE BACKUP/RECOVERY ACTIVITY LOG 398
ARC6368I (CONT.) MESSAGE LEVEL IS FULL
ARC6371I AGGREGATE RECOVERY ML2 TAPE UNIT NAME = 3590-1
ARC6372I NUMBER OF ABARS I/O BUFFERS = 01
ARC6373I ABARS ACTIVITY LOG OUTPUT TYPE = SYSOUT(A)
ARC6033I AGGREGATE RECOVERY UNIT NAME = 3590-1
ARC6036I AGGREGATE BACKUP OPTIMIZE = 3
ARC6036I AGGREGATE RECOVERY TGTGDS = SOURCE
ARC6036I AGGREGATE RECOVERY ABARSVOLCOUNT = *NONE*
ARC6036I AGGREGATE RECOVERY PERCENTUTILIZED = 080
ARC6036I AGGREGATE BACKUP/RECOVERY ABARSDELETEACTIVITY 407
ARC6036I (CONT.) = NO
ARC6036I AGGREGATE BACKUP/RECOVERY ABARSTAPES = STACK
ARC6036I AGGREGATE BACKUP ABARSKIP = NOPPRC, NOXRC
ARC1500I PLEXNAME=ARCPLEX0,PROMOTE PRIMARYHOST=NO, 410
ARC1500I (CONT.) PROMOTE SSM=NO,COMMON RECALL QUEUE BASE NAME=*****,
ARC1500I (CONT.) COMMON RECALL QUEUE TAPEDATASETORDER=*****,COMMON
ARC1500I (CONT.) DUMP QUEUE BASE NAME=**** AND MSC=***,COMMON RECOVER
ARC1500I (CONT.) QUEUE BASE NAME=**** AND MSC=***
ARC0468I EXTENDEDTTOC=N
ARC0278I CLASSTRANSITIONS EVENTDRIVENMIGRATION=Y, 412
ARC0278I (CONT.) SERIALIZATIONEXIT=N
ARC0101I QUERY SETSYS COMMAND COMPLETED ON HOST=1
ARC0101I QUERY STARTUP COMMAND STARTING ON HOST=1
ARC0143I PARMLIB MEMBER=ARCCMD00, DFSMSHSM AUTHORIZED 415
ARC0143I (CONT.) USERID=IBMUSER, HOSTID=1, PRIMARY HOST=YES, LOGSW=NO,
ARC0143I (CONT.) STARTUP=YES, EMERGENCY=NO, CDSQ=YES, CDSR=NO, PDA=YES,
ARC0143I (CONT.)  RESTART=NOT SPECIFIED, CDSSHR=NOT SPECIFIED,
ARC0143I (CONT.) RNAMEDSN=NO, STARTUP PARMLIB MEMBER=ARCSTR0
ARC0249I CELLS=(200,100,100,50,20),HOSTMODE=MAIN
ARC0101I QUERY STARTUP COMMAND COMPLETED ON HOST=1
ARC0101I QUERY CSALIMITS COMMAND STARTING ON HOST=1
ARC0203I CSALIMITS=YES, CSA CURRENTLY USED=0 BYTES, 419
ARC0203I (CONT.) MWE=0, MAXIMUM=0K BYTES, ACTIVE=0%, INACTIVE=0%
ARC0101I QUERY CSALIMITS COMMAND COMPLETED ON HOST=1
ARC0038I RESOURCE MANAGER SUCCESSFULLY ADDED.  RETURN 421
ARC0038I (CONT.) CODE=00
ARC0008I DFSMSHSM INITIALIZATION SUCCESSFUL
```

Figure 54 (Part 4 of 4). Example of a z/OS V2R5 DFSMShsm Startup Screen (IVP)

8. Prevent extraneous error messages from occurring when you stop DFSMShsm by putting the system in debug mode.

   For example, if you specify `F  DFSMSHSM,SETSYS DEBUG`, then DFSMShsm operates in debug mode.

9. Stop DFSMShsm

   For example, if you specify `F  DFSMSHSM,STOP`, DFSMShsm stops with a message ARC0002I, which indicates DFSMShsm has stopped successfully. See Figure 55 for an example of the shutdown screen.

```
F DFSMSHSM,STOP
ARC0016I DFSMShsm SHUTDOWN HAS BEEN REQUESTED
ARC0002I DFSMShsm SHUTDOWN HAS COMPLETED
$HASP395 DFSMSHSM   ENDED
$HASP150 DFSMSHSM   OUTGRP=1.1.1 ON PRT1     2 (2) RECORDS
$HASP150 DFSMSHSM   OUTGRP=2.1.1 ON PRT1     7 (7) RECORDS
$HASP160 PRT1       INACTIVE - CLASS=AJ
$HASP250 DFSMSHSM   IS PURGED
```

Figure 55. Example of DFSMShsm Shutdown Screen

   You can start the IVP again from the beginning at any time by returning to Step 5.

10. Run job HSMIVP2, which is a member of DFSMSHSM.IVP.CNTL, to remove the IVP temporary data sets from your system.

11. After running the IVP, you can delete member DFSMSHSM from the SYS1.PROCLIB data set and delete the DFSMSHSM.IVP.CNTL data set. They are no longer needed. The DFSMSHSM configuration and the parameters in the ARCCMD00 member created by the IVP are used to determine if the product is correctly installed and can be started and stopped. They are not intended for use in a fully functional DFSMShsm environment. Return to "How to Implement DFSMShsm" in *z/OS DFSMShsm Implementation and Customization Guide* for the next steps to take to implement DFSMShsm.

8.6.4.4 Run the DFSMS File System Installation Verification Procedure: When z/OS V2R5 is installed in a zFS root file system, the following command may be invoked under the z/OS UNIX shell to validate the product is installed in the zFS and is an example (it must be modified as required for your environment)

```
  zfsadm aggrinfo -long OMVS.ZFS.ZOS25.ROOT
```

In the example above, OMVS.ZFS.ZOS25.ROOT, is zFS name. The output will display information about the file system.

8.6.4.5 Run the HCD Installation Verification Procedure: To verify that HCD is installed, you should at least make the following minimal checks.

1. Issue the 'D IOS,CONFIG' command and verify the resulting IOS506I message.
2. Invoke HCD. You will receive the primary task selection panel of HCD, Hardware Configuration. Verify that it says "z/OS V2.5 HCD" at the first line of the panel. Select "What's new in this release" from the primary task selection panel. Verify it says "What's New in This Release" at the heading of the panel. Finally, scroll forward. You will see:

   ```
   What's New in This Release
   ```

   This panel tells you what changes have been made in the present release. If you have not used HCD before, select Item 8, 'Getting Started with This Dialog' on the primary selection panel for "Overview of Changes".

   For information on HCD support for new processor types and the enhanced capabilities of new processor models, select 'Query supported hardware and installed UIMs' on the primary selection panel and then 'List supported processors.'

   For information on new functions and enhancements of the present HCD release, move your cursor to a highlighted topic, then press Enter. Pressing F12 on the panel describing the selected topic returns you to this panel to select another topic.

8.6.4.6 Run the z/OS UNIX System Services Setup Verification Procedures: Refer to the z/OS UNIX System Services setup verification procedure in *z/OS UNIX System Services Planning*.

8.6.5 IVP jobs for Wave 1C

There are currently no installation verification procedures for the following elements in Wave 1C:

- MICR/OCR
- TSO/E

8.6.5.1 Run the FFST Installation Verification Program: FFST provides a program for execution to verify you have installed FFST correctly. Perform the following steps to ensure successful installation of FFST.

1. Start FFST with the procedure (EPWFFST) provided.
2. Execute the IVP provided. This is found in your SEPWSRC2 data set, member EPW12012. This job issues several EPW90xxI messages. See FFST operations in *z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT*, for detailed explanations.

This completes the testing of FFST.

8.6.5.2 Run the GDDM Installation Verification Procedures: The installation verification procedures for OS/390 GDDM consist of:

- Testing GDDM/MVS Base.
- Testing GDDM-PGF
- Testing GDDM-REXX.
- Testing GDDM Under CICS (if applicable):
  - Testing GDDM-PGF Under CICS
  - Testing the Print Utility ADMOPUC Under CICS
- Testing GDDM Under IMS (if applicable):
  - Testing GDDM-PGF Under IMS
  - Testing the Print Utility ADMOPUI Under IMS

8.6.5.2.1 Testing GDDM/MVS Base

Before performing the installation verification procedures, ensure the default User Control key is set. The default User Control key is PA3. If your keyboard does not have a PA3 key, or the PA3 key is already used by another application, set the User Control key to another value. To do this, add the following nickname user default specification to the external defaults file, or external defaults module:

```
ADMMNICK FAM=1,PROCOPT=((CTLKEY,type,value))
```

where `type` defines the type of key selected for entering User Control (1=PF key, 4=PA key), and `value` defines the number of the PF or PA key to be used. The line must start with a space in column 1.

For more information about nicknames and supplying user-default specifications, see *GDDM System Customization and Administration*.

For more information about User Control, see the *GDDM User's Guide*.

This is how to test that GDDM/MVS has been installed successfully for use under TSO:

1. Allocate the GDDM symbol and map data sets on your TSO session by typing:

   ```
   ALLOC F(ADMSYMBL) DA('GDDM.SADMSYM') SHR REU
   ALLOC F(ADMGDF)   DA('GDDM.SADMGDF') SHR REU
   ```

   If you also intend to use this step to test the GDDM TSO Print Utility, you must allocate the Master Print Queue data set:

   ```
   ALLOC F(ADMPRNTQ) DA('your-master-print-queue') SHR REU
   ```

2. Run the GDDM installation verification program:

   ```
   CALL *(ADMUGC) 'FROM(ADMTEST)'
   ```

   A picture should be displayed that shows the words 'Welcome to GDDM Version 3.2'. This picture is the ADMTEST ADMGDF from the SADMGDF data set being displayed by the ADMUGC conversion utility.

   If the terminal does not support graphics, the following message will be displayed:

   ```
   ADM0275 W GRAPHICS CANNOT BE SHOWN. REASON CODE 3
   ```

3. Go into GDDM User Control by pressing the User Control key.

The User Control panel should now be superimposed on the bottom of the display, using the language specified by the NATLANG external default.

While you are in this panel, you can also test the GDDM print utility if you have already customized it. For information about the print utility, see *GDDM System Customization and Administration*. To create a file for testing the GDDM print utility using GDDM User Control:

a. Press **PF4** for the User Control output panel.

b. Type the terminal ID of the printer to which you want to send the output, as defined to Communications Server for z/OS SNA Services, or in your external defaults module.

c. Press **PF4** to send the file to be printed. A highlighted message is displayed when the print has completed successfully.

4. Press **PF3** until you are back in TSO.

This completes the test of the graphic and alphanumeric functions of GDDM/MVS.

8.6.5.2.2 Testing GDDM-PGF

If GDDM-PGF has been enabled (see 8.5.2.3, “IFAPRDxx considerations” on page 140), verify GDDM-PGF has been installed properly by performing the following steps:

1. Use the commands suggested for a CLIST in Figure 56 on page 220 to call the Interactive Chart Utility (ICU).

   When you start, the Home Panel of the ICU is displayed in the language specified in the NATLANG external default.

   For information about using the ICU, you can access the built-in help facility by pressing the **PF1** key; you can also find more information in *GDDM-PGF Interactive Chart Utility*.

2. This is what you do:

   a. Type `0` to move to the Chart by Example panel.

   b. Type `1` on the Chart by Example panel to get data headings and titles for your chart.

   c. Type any number from 1 through 7 on the Chart by Example - Data panel to get the sample data supplied with the ICU.

   d. Display the data by pressing **PF5**, the Display key.

   e. Return to the Home panel by pressing **PF12**.

   f. Exit the ICU by pressing **PF9** twice.

This completes the tests of the graphic and alphanumeric functions of GDDM-PGF.

8.6.5.2.3 Testing GDDM-REXX

If GDDM-REXX has been enabled (see 8.5.2.3, “IFAPRDxx considerations” on page 140), verify GDDM-REXX has been installed properly by performing the following steps:

1. Allocate the GDDM map and symbol set data sets to your session by typing:

```
ALLOC F(ADMGDF) DA('GDDM.SADMGDF') SHR REU
ALLOC F(ADMSYMBL) DA('GDDM.SADMSYM') SHR REU
```

2. Type the command `EXEC 'GDDM.SADMSAM(ERXMODEL)'` from a terminal that can display graphics.
3. Press **ENTER**.

   A picture is displayed, with the large characters “GDDM-REXX” near the top of the screen.
4. Press **ENTER** again or press the PF3 key to return to TSO.

8.6.5.2.4 Testing GDDM under CICS

This section describes how to verify the installation of GDDM under the CICS subsystem.

To test GDDM under CICS, you must have allocated the required GDDM VSAM data sets, defined them to CICS, and included them in the CICS startup JCL. You must also have defined GDDM to CICS in your CICS tables or CSD.

8.6.5.2.4.1 Testing GDDM/MVS Base under CICS

Before performing the installation verification procedures, ensure the default User Control key is set. The default User Control key is PA3. If your keyboard does not have a PA3 key, or the PA3 key is already used by another application, set the User Control key to another value. To do this, add the following nickname user default specification to the external defaults file, or external defaults module:

```
ADMMNICK FAM=1,PROCOPT=((CTLKEY,type,value))
```

where `type` defines the type of key selected for entering User Control (1=PF.key, 4=PA key) and `value` defines the number of the PF or PA key to be used. The line must start with a space in column 1.

For more information about nicknames and supplying user-default specifications, see *GDDM System Customization and Administration*.

For more information about User Control, see the *GDDM User's Guide*.

This is how to test GDDM/MVS has been installed successfully for use under the CICS subsystem:

1. Type the transaction name for the GDDM/MVS installation verification program: ADMA.
2. Press **ENTER**

   A picture should be displayed that shows the words “Welcome to GDDM Version 3.2”. This picture is the ADMTEST ADMGDF, which is stored in the ADMF data set.
3. Go into GDDM User Control by pressing the User Control key.

   The User Control panel should now be superimposed on the bottom of the display, using the language specified by the NATLANG external default.

   While you are in this panel, you can also test the GDDM print utility if you have already customized it. For information about the print utility, see *GDDM System Customization and Administration*. To create a print file for testing the GDDM print utility using GDDM User Control:

   a. Press **PF4** for the User Control output panel.

   b. Type the terminal ID of the printer to which you want to send the output, as defined in your TCT.