

Operating **S**ystem **E**nvironment **M**anager

For z/OS

Installation Guide

Version 6.0

SC31-6908-00

Limits of Liability and Disclaimer of Warranty

Trident Services and E.S.A. Software makes no warranty of any kind, expressed or implied, with regard to the programs or documentation. Trident Services and E.S.A. Software shall not be liable in any event for incidental or consequent damages in connection with or arising out of the furnishing, performance, or use of these programs.

Information in this manual is subject to change without notice and does not represent a commitment on the part of the vendor. The software described in this manual is furnished under a license agreement, and may be used or copied only in accordance with the terms of that agreement.

Copyright Notice

IBM Operating System Environment Manager (OSEM) for z/OS. Licensed materials - Property of IBM.
5799-HAX

(c) Copyright IBM Corp 2005. All rights reserved.

(c) Copyright E.S.A. Software 1990-2005. All rights reserved.

No parts of this publication may be copied or distributed, transmitted, transcribed, stored in a retrieval system, translated into any human or computer language, or disclosed to third parties without the express written permission of IBM Corp or E.S.A. Software.

The following are trademarks of IBM Corp:

DFHSM
DFSMS
IBM
OS/390
RACF
z/OS

The following are trademarks of Computer Associates International:

CA-ACF2
CA-TOPSECRET
CA-1
EZ-Proclib

First Edition (April 2005)

This edition applies to Operating System Environment Manager for z/OS (OSEM for z/OS) Version 6 Release 0 Modification 0 (Program Number 5799-HAX).

Table of Contents

Installation Planning	PLAN-1
CPU Serial Number	PLAN-1
Obtain CPU Number	PLAN-1
Display CPU Number	PLAN-1
Current User Exits	PLAN-1
OS/EM Reload Function	PLAN-2
Third Party Exits	PLAN-2
JES2 Job Routing Option	PLAN-2
RACF User Exits	PLAN-2
SMP/E	PLAN-2
Dataset Naming Convention	PLAN-3
Tape Allocation Rules	PLAN-3
Command Checking	PLAN-4
Job Classes	PLAN-4
Job Class Checking	PLAN-4
OS/EM IEFUSI Option	PLAN-4
Possible S71A Abends	PLAN-5
OS/EM and ISPF Installation-Wide Exits	PLAN-5
Installation	INST-1
System Requirements	INST-1
Installation	INST-2
Step 1: Load the Installation Library	INST-2
Step 2: Execute the OS/EM Installation Dialogue	INST-2
Step 3: Load the Pre-built SMP/E Environment	INST-3
Step 4: Define Security (optional)	INST-5
Step 5: Define Subsystem Name OSEM	INST-5
Step 6: Define Procedure OSEM	INST-7
Step 7: Define OS/EM Load Library to LINKLST	INST-7
Step 8: Authorize OS/EM Executable Load Library	INST-8
Step 9: Copy OS\$START SAMPLIB Member	INST-8
Step 10: Add the ISPF Interface	INST-8
Step 11: Create ISPF Tables for OS/EM Version 6.0	INST-9
Step 12: Upgrade OS/EM Tables to OS/EM Version 6.0 Tables	INST-9
Step 13: Enter Authorization Code for OS/EM	INST-10
Step 14: Build Initialization Member for the Authorization Code	INST-10
Step 15: JES2 EXIT Implementation	INST-10
New OS/EM Users	INST-10
IBM Command Conversion Routine	INST-11
Users Upgrading From Prior Versions of OS/EM	INST-11
Step 16: Set up PARMLIB Members	INST-12
New OS/EM Users	INST-12
Users Upgrading From Prior Versions of OS/EM	INST-12
Step 17: Update HSM Parmlib Member ARCCMDxx	INST-13
Step 18: Update SMF Parmlib Member SMFPRMxx	INST-13
Step 19: Password Authentication (RACF users only)	INST-13
Step 20: IPL System	INST-14

Step 21: HSM Optimizer Reports	INST-14
OS/EM Maintenance	MAINT-1
Applying OS/EM Maintenance	MAINT-1
Applying JES Maintenance	MAINT-2
Using Secondary JES2 Subsystems	MAINT-2
Readers's Comment Form	READER-1

Installation Planning

This section provides a list of important topics, which you will need to consider prior to installing OS/EM. The intent is to assist you in assessing the impact, if any, that OS/EM will have on your installation.

CPU Serial Number

You must supply Trident Services with the CPU serial number of each machine that OS/EM will run on. This number is available from IBM (or other manufacturer if your CPU is not from IBM) and consists of a six digit number. The four low-order numbers are the important ones.

Obtain CPU Number

If you are unsure of your CPU serial number, you may enter the following command on an MVS master console to obtain it.

Display CPU Number

```
D M=CPU
IEE174I 12.42.25 DISPLAY M 618
PROCESSOR STATUS
ID CPU SERIAL
0 + 0114BA3746
1 + 1114BA3746
2 + 2114BA3746
CPC ND = 009672.R31.IBM.02.000000052227
CPC ID = 00
+ ONLINE - OFFLINE . DOES NOT EXIST
CPC ND CENTRAL PROCESSING COMPLEX NODE DESCRIPTOR
CPC ID CENTRAL PROCESSING COMPLEX IDENTIFIER
```

In this sample display, the serial number of ID 0 is '0114BA'. The last 4 characters are the CPU model number. OS/EM uses the last 4 characters of the serial number, in this case '14BA'.

Current User Exits

Every user exit currently in use at your installation should be thoroughly documented as to its use and the location of both source and load members. All such exits should be functioning properly and should not modify any control blocks not specifically allowed at the exit point. Further, your exits should do nothing more than what is explicitly allowed in the relevant IBM documentation.

If any of these exits can be replaced by using OS/EM's optional facilities, determine if the optional facilities meet all your needs. If your exits contain more functions than those supplied by OS/EM, consider recoding your exits leaving only those functions not supplied by OS/EM.

You will also need to determine at what point you want the optional OS/EM functions to be invoked, either before your own exits or after all your exits have been processed. It is entirely possible that a conflict can develop between your own exits and the optional functions. For example, the optional functions supplied for IKJEFF10 modify the JOB statement. If your IKJEFF10 exit depends on certain information in a certain format on the JOB statement, the OS/EM optional processing might result in a different result for your exit.

OS/EM Reload Function

In order to use the RELOAD function provided in the ISPF Interface, all user exits need to be defined to the Interface using the Basic Exits Functions. Although OS/EM will automatically find and load all user exits which have the standard IBM module names and are in the Linklist whether or not they have been defined to the Interface, the list of modules eligible to be reloaded is built from your entries in the Basic Exits Functions. If you prefer not to enter this information, please refer to "RELOAD Command" on page RELOAD-1 in the OS/EM Reference Manual for instructions on reloading modules manually.

Note: This information will be added automatically to the interface during the upgrade process or during the 'REBUILD' function from the 'MAINTENANCE' section of the interface.

Third Party Exits

Many products, such as job schedulers and SYSOUT archival systems, depend on exits to accomplish their work. You should know at which exit points these supplied exits are invoked, and where they currently reside.

Many of these products also require the same exit point. You should determine in what order they should be invoked if OS/EM is to load and enable them.

JES2 Job Routing Option

The OS/EM optional feature ***Job Routing*** allows an installation to route jobs to specific LPARs within a MAS. The Job Routing Communications dataset must be on DASD shared by each LPAR within the MAS. Additionally the Job Routing function must be enabled on each LPAR within a MAS concurrently. Failure to do so will result in jobs not being allowed to execute on LPARs where Job Routing is active if they have been through the interpreter on a LPAR without Job Routing. Conversely, LPARs within the MAS without Job Routing active may select jobs for execution without the specified resources.

RACF User Exits

At installations where it is not permissible for a 3rd party product to manage RACF user exits, you may specify to OS/EM to use **SAF** as the security system in use (see Step 7 of the installation instructions.) This will disable OS/EM from managing RACF exits as well as disallowing use of any OS/EM feature for RACF such as Surrogate Password Control, while still allowing calls to RACF for validation of other controls such as JOBCLASSCHECK (see Step 6 of the installation instructions.)

SMP/E

You should determine which of your exits have been installed via the SMP/E USERMOD process. Once OS/EM has been installed, SMP/E knowledge of such exits is no longer necessary. OS/EM dynamically loads its interface modules during initialization; and these, in turn, dynamically load your user exit modules. Therefore, after successful installation, you should remove the exits from SMP/E by the RESTORE

function and link them into an OS/EM accessible library with different names (e.g. MYUJI), then update the OS/EM initialization parms to load the new names.

Dataset Naming Convention

If your installation does not currently have a consistent dataset naming convention, one should be developed to make full use of the OS/EM DASD allocation built-in facility. At a minimum, the naming convention should be able to distinguish between production and test datasets. Ideally, your convention should be able to segregate the various types of datasets in your installation, datasets used primarily for online applications, those datasets used primarily for batch, etc.

Think in terms of classes of data. That is, you might have datasets used for online applications. These datasets should be allocated for the least amount of volume, channel, and control unit contention. Such datasets would constitute a class in OS/EM terms, a dataset name group.

To ensure that these dataset name groups get the processing you want them to have, volumes should be pooled. If you have volumes that have multiple physical and logical paths, these volumes should be reserved for those datasets that have critical access requirements for online datasets. By listing these volumes in a particular OS/EM volume group and designating the proper dataset name group to this volume group, you can ensure that all such datasets will be properly allocated.

If your installation has an existing dataset naming convention, determine if an adequate number of dataset name groups can be created to accomplish your allocation goals. You may find that, without too much re-working of your existing convention, you can group datasets in a satisfactory manner.

OS/EM can also be of benefit in migrating to a new dataset naming convention. You can define a couple of dataset name groups representing a subset of the new convention, establish one or two volume groups for these dataset name groups, and as you allocate datasets under the new names, they will be placed on these reserved volumes. Once volumes are free of datasets under the old names, they can then be used to establish new volume groups with dataset name groups representing another subset of the new convention. Migration can then be done in small, easy steps.

- Remember, OS/EM DASD allocation neither imposes, nor demands, any specific naming convention. OS/EM dataset name groups can consist entirely of discrete dataset names. However, maintenance of such dataset name groups would be problematic. And, without some means of functionally grouping datasets, your allocation rules would be very general at best and may not achieve the desired results.

Tape Allocation Rules

Most likely your installation already has rules concerning the use of tape devices by test jobs. These may be codified in some standards manual; or an informal set of rules, sometimes enforced by operators when there are not enough tape devices to go around. Whatever the rules, you should give some thought as to how you want to implement this OS/EM function. Remember, this function is intended primarily as a means to enforce your installation's standards concerning tape usage by test jobs. The wrong MAXTAPE value applied to a production job class would certainly not win you any friends in your operations group.

Also remember that tape devices are counted as allocation proceeds; therefore, jobs may cancel that did not cancel before (due to the job getting the devices, and possibly forcing other production jobs to wait until the devices are deallocated). You should give thought to running this function in WARN mode so that violations will be apparent to users before their jobs start canceling.

Command Checking

Determine which users, if any, will be allowed to submit operating system and JES2 commands in JCL. The simplest approach would be to severely limit this ability, say to a few system programmers, and RACF define resources COMMAND.* and JES2.\$* only; and only permit the appropriate userids. However, if you have other requirements, just ensure that the appropriate resource definitions are in place before invoking this function.

Job Classes

Many OS/EM optional functions may be applied on a job class basis. This means that your installation should have a well-defined set of rules for job class usage. Any rules you develop should account for the following:

- If you run CICS, or other transaction processing system, as a batch job, you should have separate production and test classes. Typically, the production class(es) will have a higher dispatching priority than almost anything else in the system. The test class(es) will be much lower (give thought to placing these classes at a lower priority than TSO. If these test systems hang, they can lock out your TSO users).
- If you are going to control tape usage for test jobs, you need enough classes defined to handle the various allocation rules you develop. You probably will want to tie job turnaround time to the number of tape devices the job uses. That is, a quick turnaround job such as a compile would normally not require any tape devices. A job that requires three, or more, tape devices could be given a long turnaround time.
- The number of production job classes should be held to a minimum. It is not the number of job classes that determines how many production jobs can run, but the number of initiators assigned to these classes.

However, you should have at least one class reserved for those production jobs that consume excessive resources and single-thread such jobs through this class. We also recommend that you reserve one production class for those jobs that absolutely need to be executed when they are submitted. Such jobs would be the SMF dump job; and, if your CICS journal is on disk, the job that dumps a full disk journal to tape.

Job Class Checking

Some of the optional OS/EM control functions allow you to check that a specified job class is valid for the job. These classes must be defined to your security system using the classname FACILITY (or IBMFAC for CA-TOPSECRET) and resource JOBCLASS.x where x is the desired class. If the resource is not defined, OS/EM will by default allow access to the resource, so no Job will fail.

OS/EM IEFUSI Option

The optional OS/EM IEFUSI function is applied based on the weight assigned to each selection criteria type: program name, job class or job name. Before applying storage limits, determine, at the very least, your installation's use of job classes. If, for example, you start CICS as a job, you would not want to specify the CICS job class with the wrong storage limits. In this case, it would be better to specify DFHSIP as a program name and assign it to a REGION keyword with the appropriate limits.

Possible S71A Abends

When using SMF exit IEFU83, including the OS/EM extended function Catalog Account Control, it is possible that SVC dumps will be produced for S71A abends. These abends are normally suppressed by the Catalog Address Space where they occur, but when CAS is called by IEFU83 a dump may be produced as OS/EM's SMF Controller may detect the abend. The same problem may be encountered without OS/EM as the SMF ESTAE routine IEEMB830 would then detect the abend.

IBM recommends that you set a SLIP trap in IEASLP00 in SYS1.PARMLIB to suppress the dump. An example for ABEND71A is:

```
SLIP SET,C=71A, ID=X71A, A=NODUMP, END
```

This information is documented in IBM APAR OW13864.

OS/EM and ISPF Installation-Wide Exits

To use the OS/EM support of the ISPF installation-wide exits, you must have set the EXITS keyword on the ISPMTAIL macro to YES, or by using the ISPCCONF command to change the ISPDFLTS setting 'Enable ISPF Exits'.

For detailed instructions, refer to the ISPF Planning and Customizing manual, section Tailoring ISPF Defaults.

Also be sure that you do not have a version of ISPEXITS available in a STEPLIB in any ISPF logon procedure, otherwise ISPEXITS will be loaded from the STEPLIB and not from the OS/EM load library. Any exit (and associated data area) coded in your ISPEXITS module must be defined to OS/EM via the Basic Exits function.

Installation

OS/EM is distributed as a pre-built SMP/E environment. The installation process is performed through an ISPF dialogue that guides the user to:

1. Define and load the SMP/E datasets
2. Define and load the OS/EM target and distribution libraries
3. Generate the appropriate JES offset tables
4. Define and build the OS/EM executable libraries and control datasets

Additional installation steps involve setting up system environment and other related subsystems (e.g. RACF, DFSMSHsm) to support OS/EM operation.

Some changes to the system PARMLIB will be required to enable OS/EM operation.

System Requirements

This section identifies the system requirements for installing and running OS/EM.

- z/OS 1.4 or later
- An available Type 3 SVC number for JES3 only, which you choose, or let OS/EM dynamically choose one

Installation

OS/EM is distributed via CD. This CD contains a binary distribution file named **OSM0600.DIST.TERSE**. As the name implies, the data is in the 'TERSED' format using the TRSMAN utility. Users who do not have the TRSMAN utility installed on their system can obtain it from the IBM Technical Support website:

<http://techsupport.services.ibm.com/390/trsmain.html>

Download the distribution file into a sequential dataset with the following attributes:

```
RECFM=FB,LRECL=1024,BLKSIZE=6144,SPACE=(CYL,(30,5))
```

The distribution dataset contains a pre-built SMP/E environment which is loaded into the pre-allocated datasets. An ISPF dialogue is provided to guide the user through the installation process (this is similar to a ServerPac installation procedure). Follow these instructions carefully to avoid errors when attempting to load the SMP/E environment.

Step 1: Load the Installation Library

The Installation Library contains the ISPF dialogue and JCL skeletons that are used to define and load all the OS/EM datasets. Extract the Installation Library using the following JCL:

```
//jobname JOB (acct info), 'name', CLASS=A, MSGCLASS=X
//*
//UNPKDIST EXEC PGM=TRSMAN, PARM=UNPACK
//SYSPRINT DD SYSOUT=*
//INFILE DD DSN=osem.dist.dataset, DISP=OLD
//OUTFILE DD DSN=&&PDS, DISP=(NEW, PASS),
//          UNIT=SYSDA, SPACE=(CYL, (15, 30, 15), RLSE)
//*
//EXTRPREP EXEC PGM=TRSMAN, PARM=UNPACK
//SYSPRINT DD SYSOUT=*
//INFILE DD DSN=&&PDS(PREP), DISP=(OLD, DELETE)
//OUTFILE DD DSN=h1q.OSEM.PREP, DISP=(, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(27920, (20, 15, 15), RLSE),
//          VOL=SER=SYS000
```

- Replace the JOB statement with one that meets your installation's requirements.
- Replace the DSN= for the INFILE dataset in the UNPKDIST job step to point to the downloaded OS/EM distribution dataset.
- Replace the DSN= for the OUTFILE dataset in the EXTRPREP job step to meet your installation's requirements, and specify an appropriate DASD esoteric name and volume serial number.

Step 2: Execute the OS/EM Installation Dialogue

Execute the Installation Dialog by entering the following TSO command:

```
ex 'h1q.OSEM.PREP(OS$DISK)' 'h1q.OSEM.PREP'
```

Ensure that dataset names specified in the above command is the same that was used in installation step 1.

The following menu will be displayed:

```

OS/EM ----- OS/EM Installation Preparation ----- Version 6.0
Option ==>

          1  Set Variables
          2  Allocate DLIB/TLIB Datasets
          3  Allocate SMP/E Datasets
          4  Allocate Executable Datasets
          5  Load SMP/E Environment
          6  Create JES Offset Table
          7  Copy SMP/E to Executable

UserID   -SPRCB
System ID-PROD
Time     -14:54
Terminal -3278
PF Keys  -12

F1=Help  F2=Split  F3=Exit  F7=Up    F8=Down  F9=Swap  F12=Cancel

```

Figure 1. OS/EM Installation Menu

Step 3: Load the Pre-built SMP/E Environment

The pre-built OS/EM SMP/E environment is loaded by executing the selection items in the main menu in sequence.

1. **Set Variables** This option defines the parameters to be used for subsequent installation steps. Set the values according to your installation's naming conventions.

```

OS/EM ----- OS/EM Installation Variables ----- Version 6.0
COMMAND ==>

      Library Type           High Level Qualifier       Product Qualifier
SMP/E Libraries             ==> MVS$SMP                 ==> OSM0600
Distribution Libraries      ==> SYSA                   ==> OSM0600
Target Libraries           ==> SYS2A                  ==> OSM0600
Executable Libraries       ==> SYS2                     ==> OSEM

COBOL SCEELKED Library DSN ==> SYS1.SCEELKED
JES SMPMTS Library DSN    ==> MVS$SMP.ZOSV106.SMPMTS
RELFILE DSN Prefix        ==> SYS2A.REL
TRSMAN Library DSN        ==> SYS1.IBM.LINKLIB

Unit and Volume Names:
SMP/E Unit                 ==> SYSDA                 Volume ==> SMS001
DLIB Unit                  ==> SYSDA                 Volume ==> SMS002
TLIB Unit                  ==> SYSDA                 Volume ==> SMS003
Executable Unit            ==> SYSDA                 Volume ==> SMS004
Work File Unit             ==> SYSDA

Distribution File Name      ==> OSEM600.D050101.PACK

F1=Help  F2=Split  F3=Exit  F7=Up    F8=Down  F9=Swap  F12=Cancel

```

Figure 2. OS/EM Installation Variables

Notes:

- Do not use quotes (') when specifying dataset names.

- The SMP/E Macro Temporary Storage dataset (SMPMTS) is used for generating the JES Offset tables. If your installation does not have an SMPMTS dataset, specify SYS1.MACLIB.

Subsequent installation steps will submit jobs. For each job that will be submitted, the following panel will be displayed:

```

OS/EM ----- ALLOCATE DLIB/TLIB DATASETS ----- Version 6.0
COMMAND ==>

Select from the following:
  1 - Submit the Job
  2 - Browse the Job
  3 - Edit the Job

Job Statement Information:  Verify before proceeding

==> //SPRCBA JOB (ACCT),ALLOCATION,
==> //          MSGCLASS=X,CLASS=A,NOTIFY=&SYSUID,REGION=0M
==> //*
==> /*

Select option 1 to submit job, or press PF3 (END) to exit.

F1=Help   F2=Split   F3=Exit   F7=Up     F8=Down   F9=Swap   F12=Cancel

```

Figure 3. OS/EM Installation Job Submission

Alter the JOB statement to meet your installation's standards and then select option 1 to submit the job. You may browse or edit the JCL prior to submission using options 2 and 3.

2. **Allocate DLIB/TLIB Datasets** This step will allocate the OS/EM Distribution and Target libraries (load libraries, ISPF libraries, sample library). You should receive a return code of zero for this job.
3. **Allocate SMP/E Datasets** This step will allocate the SMP/E datasets for the OS/EM system. You should receive a return code of zero for this job.
4. **Allocate Executable Datasets** This step will allocate the datasets that OS/EM will use for execution. You should receive a return code of zero for this job.
5. **Load SMP/E Environment** This step will load the SMP/E environment from the OS/EM distribution tape / dataset. You should receive a return code of zero for this job.
6. **Generate JES Offset Table** You will be presented with a panel where you can choose to generate the offset table for JES2 or JES3. You should receive a return code of zero or one (for JES2) and zero (for JES3) for this job.

Important - If your installation uses multiple JES systems with different release or maintenance levels, the JES offset table must be generated on each of these systems.

7. **Copy SMP/E to Executable** This step will build the OS/EM executable datasets from the SMP/E environment. You should receive a return code of zero for this job.

Once all installation steps have been completed, terminate the dialogue by pressing PF3.

Step 4: Define Security (optional)

Many of the OS/EM optional functions such as the Job Class Checking, Account Number Checking, DDNAME Checking, JCL parameter checking, MVS Operating System and JES2 commands are controlled by appropriate definitions to your installation's security system (RACF, CA-TOPSECRET, or CA-ACF2). If you intend to use these functions, the proper definitions must be in place before OS/EM can enforce the use of these resources. If resource checking (RACROUTE) for a particular resource is requested by OS/EM and the resource is not defined to the installation's security system, OS/EM will by default allow the use of the requested resource if using RACF, or disallow the use of the requested resource if using ACF2.

All such definitions are done by establishing the proper resources within the class FACILITY (CA-TOPSECRET users will use class IBMFAC). Job class resources take the form JOBCLASS.x where x is the desired jobclass. Users would then be permitted to the proper jobclass. For example, if JOBCLASSCHECKING is in effect, and resource JOBCLASS.A is defined, users would have to be permitted to this resource before they could submit jobs to JOBCLASS A. READ authority is sufficient.

- If JOBCLASSCHECKING is in effect, but the resources are not defined to your security system, **submission will be allowed** for RACF users.

MVS Operating system and JES2 command submittal authority is also defined within class FACILITY (or IBMFAC). The resource takes the form COMMAND.cmd for operating system commands, and JES2.\$cmd for JES2 commands.

For operating system commands, cmd is the long form of the command. For example, COMMAND.MOUNT would be specified as the resource name for establishing authority to issue MVS MOUNT commands (do not code COMMAND.M).

The commands \$VS, \$ADD and \$TRACE must be fully designated; all other commands must be a single letter. The proper resource designation for \$A for example would be JES2.\$A, a generic resource for JES2 commands would be JES2.\$*, this covers all JES2 commands.

To permit the user to issue any JES2 command, define the resource as JES2.* and permit the user to this resource. Read authority is sufficient.

- Two special resources are allowed for command authorization: COMMAND.* and JES2.*. These resources establish global resources that users can be permitted to. These special resources would be useful where you want to restrict the authority to issue such commands via submitted JCL, but some users should not be restricted. Use of these resources will mean that each and every command does not have to be defined.
- If command checking is in effect and no resources are defined, no user will be permitted to issue commands in submitted jobs.

Finally, use of the OS\$CNTL command can be controlled via your security system. Again, the class is FACILITY|IBMFAC and the resource is OS\$CNTL.sysid.function, where sysid is the four character SMF id of your system, and function is either ALLOC, CODE, DASD, HSM, ISPF, JES2, JES3, POOL, QUERY, RACF, RELOAD, SMF, SVC, or TSO. READ access is sufficient.

Note: The resource OS\$CNTL.sysid.RACF must be defined before any RACF commands may be issued. Protection of other commands is optional.

Step 5: Define Subsystem Name OSEM

OS/EM runs as a MVS subsystem. Therefore, the subsystem name must be defined. Edit your SYS1.PARMLIB member IEFSSNxx to add OSEM as the subsystem name. The member IEFSSN60 in the OS/EM SAMPLIB can be copied into your current IEFSSN member in SYS1.PARMLIB or copy IEFSSN60 into SYS1.PARMLIB as member IEFSSNOS and update the IEASYS member parameter SSN=(00,OS).

Notes:

- The OS/EM subsystem **must** be defined **after** the JES subsystem entry.

```
SUBSYS SUBNAME(OSEM)
  INITRTN(OS$IPL)
  INITPARM('subs,MSG=x,SVC=nnn,SEC=yyyy,CLASS=zzzzzzzz,SMF=xx')
                                           ,SMF=(xx,yyy)')
                                           ,SMF=(,yyy)')
```

OSEM is the sub-system name, OS\$IPL is the execution module that establishes the OS/EM environment.

The additional entries are the initialization parameters that are

The INITPARM string is passed to the subsystem initialization routine. This string contains the following parameters:

subs The OS/EM Job Entry Subsystem to control

JES2
JES3

Note: If the sub-system parameter is not specified, OS/EM will look at the JESCT to determine the primary Job Entry sub-system in use (JES2 or JES3). If the primary Job Entry sub-system cannot be determined from the JESCT, OS/EM will search the active LPA to determine if HASPSSSM for JES2, or IATSI34 for JES3 is available to set the primary Job Entry sub-system. If neither the HASPSSSM nor IATSI34 is in the active LPA, OS/EM will load the interfaces for both JES2 and JES3.

x The message support required during OS/EM Sub-system initialization

0 Errors Only
1 Default - Includes Started/Ended and Interface module loaded messages
2 All messages

nnn The SVC number you want OS/EM to use with JES3 only. If you omit this parameter (SVC=nnn), as in the example, OS/EM starts scanning for an available SVC number and will use the first available number. The OS/EM scan starts at SVC number 255 and works backwards.

yyyy Defines the security system installed at your installation. You must code this operand, along with the CLASS parameter, if you intend to use any of the OS/EM functions that depend on security system definitions.

RACF - Resource Access Control Facility
TOPS - CA-TOPSECRET
ACF2 - CA-ACF2
SAF - System Authorization Facility

Note: SAF may be coded to disable OS/EM's ability to manage RACF exits.

zzzzzzzz Resource class for authorization checking

FACILITY - for RACF
FACILITY - for CA-ACF2
FACILITY - for SAF
IBMFAC - for CA-TOPSECRET **or**
DATASET - for CA-TOPSECRET

xx Suffix of SMFPRM member to be invoked after OS\$IPL executes

yyy SMF number to use for audit records.

Note: See SMF Record Format in the appendix for the format of these SMF records.

Step 6: Define Procedure OSEM

The OSEM procedure is automatically invoked by the OS/EM subsystem initialization. This procedure initializes the OS/EM environment.

The SAMPLIB member OSEM contains a sample procedure. Copy this member into a PROCLIB pointed to by your Master Scheduler JCL and update it to reflect your dataset naming.

```
//OSEM      PROC
//OS$INIT  EXEC PGM=OS$INIT,TIME=(0,15)
//SYSTSPRT DD DISP=OLD,DSN=xhlq.OSEM.IPL.REPORT
//SYSMDUMP DD DISP=OLD,DSN=xhlq.OSEM.DUMP
//SYSTSIN  DD DSN=xhlq.OSEM.PARMLIB(TIME),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(CODEINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(JES2INIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(JES3INIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(MVSINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(DASDINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(DSNINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(HSMINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(JCLINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(JOBINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(JOBRINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(MISCINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(PREFINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(VOLINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(POOLINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(RACFINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(RSTRINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(SVCINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(TIMEINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(TPSHINIT),DISP=SHR
//          DD DSN=xhlq.OSEM.PARMLIB(QUERY),DISP=SHR
//*
```

Ensure that 'xhlq' is set to the dataset name high-level qualifier for the OS/EM executable datasets.

In this example, the SYSTSIN DD statement is a concatenation that points to the initialization members that build the OS/EM environment. This is the way in which initialization members are built by the ISPF interface.

Since OSEM runs as a started task, you must define OSEM to your security system giving it SYS1 authority. Otherwise OS/EM initialization will fail.

Step 7: Define OS/EM Load Library to LINKLST

Perform one of the following procedures:

1. If your IEASYSxx PARMLIB member uses the PROG= parameter:
 - Copy the OS/EM SAMPLIB member PROGL6 into your system PARMLIB.
 - Update PROGL6 to point to the LNKLST name used in your system and modify the DSNAME to match the OS/EM executable load library name.
 - Update IEASYSxx to include the PROGL6 member in the list (e.g. PROG=(00,L1,L6))

2. If your IEASYSxx PARMLIB member uses the LNK= parameter:
 - Copy the OS/EM SAMPLIB member LNKLST60 into your system PARMLIB. Update the member to ensure that the executable load library name is correct.
 - Update IEASYSxx to include the LNKLST60 member in the list (e.g. LNK=(00,60))

Notes:

- Use caution if your site uses both the LNK= and PROG= parameters. A LNKLST ACTIVATE statement in a PROGxx member will cause the LNKLSTxx member(s) to be ignored.

Step 8: Authorize OS/EM Executable Load Library

Perform one of the following procedures:

1. If your IEASYSxx PARMLIB member uses the PROG= parameter:
 - Copy the OS/EM SAMPLIB member PROGOS into your system PARMLIB.
 - Update PROGOS to specify the correct executable load library name and the volume name where the library resides (as specified in installation step 2).
 - Update IEASYSxx to include the PROGL6 member in the list (e.g. PROG=(00,OS,L6))
2. If your IEASYSxx PARMLIB member uses the APF= parameter:
 - Edit the IEAAPFxx member and add the SAMPLIB member LNKLST60
 - Update the OS/EM load library definition to specify the correct executable load library name and the volume name where the library resides (as defined in installation step 2).

Notes:

- Users may use both IEAAPFxx and PROGxx to define authorized load libraries, however there are restrictions related to dynamic authorization lists. Users should consult the MVS Initialization and Tuning Reference if you intend to use both mechanisms.

Step 9: Copy OS\$START SAMPLIB Member

- Copy the member OS\$START from the OS/EM SAMPLIB into a SYSPROC library that is concatenated to your TSO logon procedure.
- Modify the symbolic parameters in the **SITE SPECIFIC PARMS** block in the OS\$START CLIST to match the names of the ISPF libraries that you will be using. If this is a first time install, the symbolic OLDTLIB should point to your new table library for OS/EM.

Note: The OS/EM CLIST library is shipped with LRECL=80 and RECFM=FB. Only CLIST libraries with like characteristics can be concatenated. If your CLIST libraries are RECFM=VB, it is recommended that you use the sample CLIST ICQSMC00 to copy these CLISTS to a RECFM=VB CLIST dataset. ICQSMC00 is shipped in the IBM library ICQ.ICQSAMP as a part of TSO/E.

Step 10: Add the ISPF Interface

Since you will use the ISPF interface on a regular basis, you should place an entry for the interface on an ISPF menu, preferably a menu that is not widely available to your regular ISPF users.

If you use **O** as the selection letter, the entry in the menu's)PROC section would be:

```
O, 'CMD(OS$START) NOCHECK'
```

You will also need to add the following line at the end of the menu's)PROC section to allow users to go directly to an OS/EM sub-menu.

```
&NXOPT = .TRAIL
```

Step 11: Create ISPF Tables for OS/EM Version 6.0

This procedure will build the initial OS/EM PARMLIB and ISPF Table library entries for your environment:

- Start the ISPF interface from ISPF option 6 by entering %OS\$START.
- Select **Installation Functions** (Option 1) from the OS/EM Primary Option Menu.
- Select **Create Tables for Version 6.0** (Option 9) from the MAINTENANCE AND INSTALLATION panel.

This process may take several minutes. When complete, the OS/EM Tables and Initialization members will be initialized.

Notes:

- For more information about this process see **Installation, Create** in the OS/EM User Guide.
- In the event that this procedure does not complete successfully, **do not simply re-run the Create option**. Refer to **Installation, Create** in the OS/EM User Guide for corrective procedures.

Step 12: Upgrade OS/EM Tables to OS/EM Version 6.0 Tables

If this is a new installation, go to step 13.

The **Upgrade** function parses a Query Report of your current OS/EM environment to determine which exits and/or optional features you are using and stores that information in the OS/EM Version 6.0 PARMLIB and ISPF Table library.

Perform the following procedure to upgrade to OS/EM Version 6.0:

- Start the **OS/EM 6.0 ISPF interface** from ISPF option 6 by entering %OS\$START (or whatever CLIST name you used in installation step 11 to differentiate the CLIST from your current OS/EM environment).
- Select **Installation Functions** (Option 1) from the OS/EM Primary Option Menu.
- Select **Upgrade to OS/EM Version 6.0** (Option 10) from the MAINTENANCE AND INSTALLATION panel.
- The **VERIFY UPGRADE** panel will be displayed. Perform the following:
 1. Enter **YES** next to **Execute Query Function ?** if the OS/EM 6.0 upgrade process is to migrate the current OS/EM settings by executing an OS/EM query

OR

 2. Enter **NO** next to **Execute Query Function ?** and next to **Use this Query Report** enter the name of a dataset that contains the output of an OS/EM Query. The upgrade will migrate the OS/EM settings that are detailed in the saved report.
 3. A list of datasets that are used by the upgrade function is displayed at the bottom of the panel. Verify that the dataset names are correct before proceeding. If the dataset names are incorrect

press **PF3** and correct the OS\$START CLIST (or whatever name was used in installation step 11) to reference the correct datasets.

The upgrade process may take several minutes. When complete, the OS/EM settings from the current environment or the saved Query Report will be migrated to the OS/EM 6.0 tables.

Notes:

- For more information about this process see **Installation, Upgrade** in the OS/EM User Guide.
- Since the upgrade function rebuilds the initialization members, it is essential to execute this function before you IPL. Otherwise, the initialization members that the install procedure places into the OS/EM PARMLIB will be empty, and no OS/EM features or user exits will be activated.

Step 13: Enter Authorization Code for OS/EM

Users upgrading from a prior release of OS/EM should skip to Step 15 because Step 12 copies the existing authorization codes into the new tables.

- Start the **OS/EM 6.0 ISPF interface** from ISPF option 6 by entering %OS\$START.
- Select **System Level Controls** (Option 1) from the OS/EM Primary Option Menu.
- Select **Authorization Codes** (Option 1) from the MAINTENANCE AND INSTALLATION panel.
- Enter the authorization code provided with the OS/EM software distribution package.

See **Authorization Code, Definition and Use** in the OS/EM User Guide for complete instructions.

Step 14: Build Initialization Member for the Authorization Code

Users upgrading from a prior release of OS/EM should skip to Step 15 because Step 12 creates the initialization member.

If you are only installing the HSM Optimizer Reports go to step 22.

- Select **Build Initialization Member** from the OS/EM Primary Menu.
- Enter a non-blank character next to the **SYSTEM** option.

See the OS/EM User Guide for complete instructions.

Step 15: JES2 EXIT Implementation

New OS/EM Users

The OS/EM AUTOINSTALL feature (new to OS/EM Version 6.0) will process the JES2 initialization parameters and automatically place the defined user exits under OS/EM control. Therefore, there is nothing that a new user must do to implement existing JES2 exits in OS/EM.

However, since OS/EM provides centralized management of all common user exits, it makes more sense to explicitly define the user exits to OS/EM. These exits can be defined to OS/EM before or after the initial IPLs with OS/EM 6.0.

The OS/EM processing rules for JES2 exits are:

- Exits that are defined in the JES2 initialization parameters will be loaded and controlled by OS/EM. The console message OS\$2TP258 will be issued for each exit point that is defined and these exits will be indicated with **LOADED BY JES** in an OS/EM query.
- If the same exit point is defined to both OS/EM and the JES2 parms (and both are defined as active/enabled), only the OS/EM definition will be used. This permits users to migrate from JES2 to OS/EM exit definitions without the need to immediately modify JES2 parameters.

A suggested migration strategy from JES2 PARM definitions to OS/EM is:

1. IPL the system using the existing JES2 PARM exit definitions.
2. Rebuild the OS/EM ISPF Table Library and initialization members from the current environment:
 - From the OS/EM ISPF Main Menu select **Maintenance Functions** (Option 1)
 - Select **Rebuild OS/EM Tables from Query Function** (Option 8)
 - Press **Enter** (do not change any of the input fields). This process may take several minutes.
 - Return to the OS/EM ISPF Main Menu
 - Select **Build Initialization Member** (Option 8)
 - Place a non-blank character next to **JES2** under **BASIC EXIT FUNCTIONS**.
3. The exit definitions in the JES2 parameters will now be ignored in subsequent IPLs. The **EXIT(nnn)** and **LOADMOD** statements can be removed from the JES2 initialization parameters at the user's convenience.

If you are using the JES2 Pre-Initialization Exit (EXIT0) to control the start of the JES2 Initialization process, it must be Link Edited as a separate Load Module with a name other than **HASPXIT0**, and additional Exit Entry Points (such as EXIT19 or EXIT24, for example) should not be included. These should be assembled and Link Edited into a separate Load Module (or Modules) of their own and additionally defined to OS/EM, as appropriate.

IBM Command Conversion Routine

Beginning with OS/390 Version 2 Release 4, IBM has provided a compatibility and migration aid in the form of a user Exit 5 routine. Without OS/EM, this exit is invoked automatically if you do not specify your own Exit 5. Since OS/EM enables Exit 5 this user exit will not be automatically invoked. To ensure that this exit continues to be available, you **must** define it to the OS/EM Basic Exits Function. The module name of this exit is **HASX05C** with an entry point name of **HASX5CTR**. Because this exit is not MVS re-entrant, you will need to code KEY: 1 on the OS/EM entry panel.

Users Upgrading From Prior Versions of OS/EM

Prior versions of OS/EM required users to define OS/EM JES2 the interface module and the user exit points in the JES2 initialization parms. The OS/EM AUTOINSTALL feature (new to OS/EM Version 6.0) automatically adds the OS/EM JES2 Interface for all of the user exit points.

Users who are upgrading to OS/EM 6.0 **must remove** the OS/EM LOADMOD and EXIT(nnn) definitions from the JES2 initialization parms before IPLing with OSEM 6.0:

```
LOADMOD(OS$J2ITF) STORAGE=CSA

EXIT(001) ROUTINE=EXIT1, ENABLE, TRACE=NO
*
*
*
EXIT(nnn) ROUTINE=EXITnnn, ENABLE, TRACE=NO
```

Important:

Failure to remove these definitions will cause OS/EM to attempt to dynamically define these exits during JES2 initialization. These definitions will fail (load module not found) and error messages will be issued, but JES2 and OS/EM initialization will continue normally.

Step 16: Set up PARMLIB Members

New OS/EM Users

Make the following changes to the system PARMLIB:

1. **EXIT ADD definitions in PROGxx or EXITxx members.:** These members define system exit routines. OS/EM will now load and manage these exits and so you will need to remove these definitions from PARMLIB and add them to OS/EM.
 - Start the OS/EM ISPF dialog (OS\$START) and go to MVS Basic Exit Selection (2.3 from the Primary Menu) and define the user exits that are currently defined in PROGxx and/or EXITxx PARMLIB members. For more information, consult the OS/EM User's Guide.
 - If exits are defined in a **EXITxx** member, remove the **EXIT=xx** definition from the IEASYSxx member.
 - If exits are defined in a **PROGxx** member and this member contains only **EXIT ADD** definitions, remove the member suffix from the PROG= parameters in the IEASYSxx member. If the PROGxx member contains other definitions, edit the member and remove the **EXIT ADD** definitions.

Users Upgrading From Prior Versions of OS/EM

OS/EM Version 6 introduces significant architectural changes that affect how it is configured to OS/390 & z/OS.

1. **EXIT ADD definitions in PROGxx member.:** Prior versions of OS/EM required that the OS/EM exit point management modules be defined to the operating system via **EXIT ADD** control statements in a PROGxx member. OS/EM Version 6 dynamically resolves the user exit points during initialization, and so the EXIT ADD statements must be removed.
 - If the **EXIT ADD** definition statements are defined in their own PROGxx member, remove the member suffix entry from the PROG= parameter in the IEASYSxx member.
 - If the **EXIT ADD** definition statements are defined in a PROGxx member along with other control statements, edit the member and delete the **EXIT ADD** entries.

Note: If the **EXIT ADD** statements are not removed, message CSV431I will be issued at IPL time for each entry point (Module Not Found). This will not prevent OS/EM from initializing successfully.
2. **IEAFIXxx Entries** Prior versions of OS/EM required selected modules to be loaded into the Fixed Link Pack Area (FLPA) by specifying them in IEAFIXxx members. This is no longer required and these definitions must be removed from the system initialization parameters. The entries can be removed by one of the following methods:
 - If the OS/EM entries are defined in an IEAFIXxx member along with definitions for other applications, delete the OS/EM definitions from the member.
 - If the OS/EM entries are defined its own member, remove the member suffix specification from the FIX= parameter of the IEASYSxx member. If the OS/EM member suffix is the only one specified (i.e. FIX=xx), remove the FIX= parameter from the IEASYSxx member.

3. **IEALPA60** Prior versions of OS/EM required selected modules to be loaded into the Modified Pageable Link Pack Area (MLPA) by specifying them in IEALPAxx members. This is no longer

required and these definitions must be removed from the system initialization parameters. The entries can be removed by one of the following methods:

- If the OS/EM entries are defined in an IEALPAXx member along with definitions for other applications, delete the OS/EM definitions from the member.
- If the OS/EM entries are defined its own member, remove the member suffix specification from the LPA= parameter of the IEASYSxx member. If the OS/EM member suffix is the only one specified (i.e. LPA=xx), remove the LPA= parameter from the IEASYSxx member.

Step 17: Update HSM Parmlib Member ARCCMDxx

This step may be skipped during an upgrade of OS/EM, however you should review your ARCCMDxx member to be sure all DFHSM exits are specified.

In order to activate the HSM Optimizer functions the DFHSM ARCCMDxx parm member must be updated to enable all the DFHSM exits.

```
SETSYS EXITON(AD BD BE CB CD CR ED IN MD MM)
SETSYS EXITON(MV M2 RD RP SA SD SK TD TE TV)
```

Step 18: Update SMF Parmlib Member SMFPRMxx

This step may be skipped during an upgrade of OS/EM, however you should review your SMFPRMxx member to be sure all SMF exits are enabled. The EXITS parameter should be specified on both the SYS and SUBSYS keywords, as the following example shows:

```
SYS (NOTYPE (16 : 19 , 99) , EXITS (IEFACTRT , IEFUAV , IEFUJI , IEFUJP , IEFUJV ,
    IEFUSI , IEFUSO , IEFUTL , IEFU29 , IEFU83 , IEFU84 , IEFU85) ,
    INTERVAL (003030) , NODETAIL)
SUBSYS (STC , EXITS (IEFACTRT , IEFUAV , IEFUJI , IEFUJP , IEFUJV , IEFUSI ,
    IEFUSO , IEFUTL , IEFU29 , IEFU83 , IEFU84 , IEFU85) )
```

Step 19: Password Authentication (RACF users only)

This step may be skipped during an upgrade.

RACF provides three types of password authentication:

1. In-house or vendor supplied exit
2. RACF masking algorithm to perform encoding
3. RACF DES algorithm to perform authentication

Prior to RACF release 2.1 the default authentication method was the masking algorithm. Beginning with RACF release 2.1, a two-step method of checking is the default, where RACF uses both the masking algorithm and the DES algorithm.

You **must** control the type of authentication used via OS/EM if you are running RACF 2.1 or above.

- If you are using your own authentication exit, you need to define the exit to OS/EM using the ISPF Facility MVS Basic Exit Support.

- If you want to use only the masking algorithm, specify ICHDEX01 as primary user exit one using the ISPF Facility MVS Basic Exit Support. This program resides in SYS1.LINKLIB and unconditionally returns with a return code of **4**, which tells RACF to use the masking algorithm.
- If you want to use only the DES algorithm, you will need to specify your own exit as the primary user exit using the ISPF Facility MVS Basic Exit Support. This exit **must** return with a return code of **8** to force the DES algorithm to be used.
- To use the two-step method of checking, which is the default beginning with release 2.1 of RACF, nothing needs to be done as this is OS/EM's default mode of operation.

Note: For more information on using the ISPF Facility MVS Basic Exit Support, see the OS/EM User Guide. For more information regarding RACF Password Authentication, refer to the RACF System Programmer's Guide.

Step 20: IPL System

IPL the system. OS/EM will initialize with the parameters you placed in the parmlib members pointed to by the OSEM procedure.

Use the ISPF interface at any time to modify processing options and/or user exits, or to build initialization members that your installation requires.

Step 21: HSM Optimizer Reports

Refer to the OS/EM Version 6.0 Users Guide initializing and creating HSM Optimizer Reports.

OS/EM Maintenance

Applying OS/EM Maintenance

OS/EM is installed and maintained with SMP/E. Maintenance is distributed in the form of individual PTFs and Cumulative Service. This maintenance is applied using the standard SMP/E RECEIVE / APPLY / ACCEPT process.

In order to implement the maintenance, the affected library members must be copied from the target libraries to the executable libraries. If the maintenance affects a large number of members it may be more prudent to simply copy all of the target libraries to their executable counterparts:

```
LINKLIB
SISPEXEC
SISPMENU
SISPPENU
SISPSENU
SISPTENU
```

PTFs that require an IPL will be indicated through the SMP/E hold data. Users should receive the HOLDDATA as well as the maintenance and the HOLDDATA must not be bypassed when performing the initial APPLY.

When applying maintenance that does not require an IPL, the following procedure is recommended after the executable libraries have been updated.

- Refresh the Library Lookaside - F LLA,REFRESH
- Reload the OS/EM system modules. If a small number of modules are affected, this can be done through the OS/EM ISPF Management Dialogue:
 1. Select **Reload Exits** at the primary menu
 2. Select **OS/EM System Modules**
 3. Place an 'S' next to the updated modules & press Enter
 4. Press **PF3** to execute the reloads
- Alternatively, OS/EM can be re-initialized by executing the OSEM started task (i.e. S OSEM,SUB=MSTR)

Applying JES Maintenance

Any time you apply maintenance to your JES system, you must remember to reassemble the OS/EM Offset Table. This is done by selecting option 6 in the Installation Dialogue that was provided with the OS/EM distribution. Refer to installation step 2 in the OS/EM Installation Guide.

After the JES offset table is generated, it will need to be copied from the OS/EM target LINKLIB into the executable LINKLIB.

Using Secondary JES2 Subsystems

Whenever you install a new version of JES2, and this release will operate as a secondary JES subsystem, you **MUST** use the ISPF interface and update the JES2 version number. Select option 6 **Set JES Name** on the Primary Option Menu and specify the version of JES you will be using. Once this is updated, select option 8 **Build Initialization Member** and select the following items:

- Basic Exit Functions
 - JES2
- Extended Functions
 - JCL
 - JOB
 - JOBR
 - TIME

Note: This process should be done before you IPL with your new JES2 system or unpredictable results may occur requiring another IPL or a restart of JES2.

Note: This process is only relevant for secondary JES2 subsystems. The JES2 release information, if specified, is ignored for the primary JES subsystem and the release information is determined from the JES2 system during OS/EM initialization.

