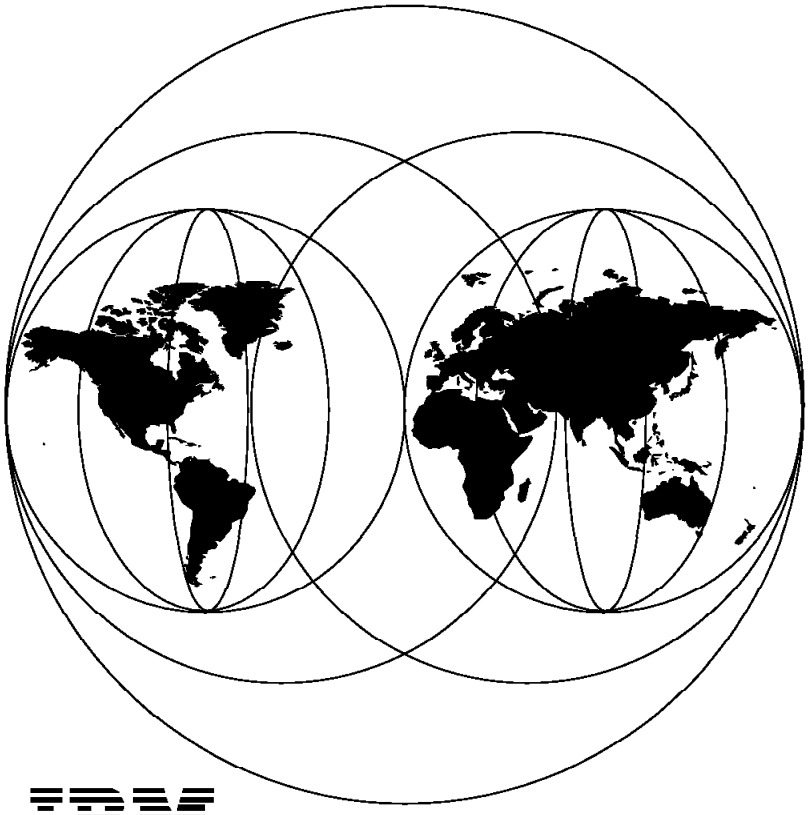
SG24-4834-00

OS/390 Release 2 Implementation
MVS, SMP/E, SDSF, and RMF

January 1997



IBM

International Technical Support Organization
Poughkeepsie Center

International Technical Support Organization

SG24-4834-00

OS/390 Release 2 Implementation
MVS, SMP/E, SDSF, and RMF

January 1997

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix E, “Special Notices” on page 239.

First Edition (January 1997)

This edition applies to Release 2 of OS/390, Program Number 5645-001 and to all subsequent releases and modifications until indicated in new editions.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

This redbook describes the new functions in OS/390 Release 2. It was written for systems programmers who need to understand these functions and the considerations for implementing them. Several practical examples are presented to demonstrate the use of these new functions. Some knowledge of OS/390 and MVS/ESA is assumed.

The OS/390 Release 2 ServerPac installation process is described by showing the process steps used in the installation of the ITSO OS/390 Release 2 system.

An enhancement for up to 10 additional data sets may be concatenated to SYS1.PARMLIB at IPL. An installation may, if desired, dynamically change the logical parmlib by switching to another set of parmlib data sets through use of a new operator command. New enhancements have been made in LNKLST and LPALST. In addition, a parmlib symbolic pre-processor tool, which is an ISPF-based interactive dialog, is available to show you what your system symbolics and parmlib members would be, given a hypothetical hardware and software configuration.

A new enhancement to Global Resource Serialization uses a star topology as an alternative to the current ring implementation. A check list is provided that makes an inventory of the analysis and activities to be followed while planning a migration to a GRS star configuration. An appendix describes a tool that monitors supervisor calls 56 and 48 (ENQ/RESERVE/DEQ) and collects data about the resources serialized and the requesters. Also, an appendix describes a sample exit ISGGREX0 based on the ISGGREXS in the SYS1.SAMPLIB. The sample exit detects volumes shared by systems outside the GRS complex and prevents RESERVE conversion for those volumes.

Also discussed in this book are the enhancements to the coupling facility structures and the programming and operator interfaces, enhancements to SMP/E to reduce the complexity of an OS/390 installation with the ServerPac, the new functions available with SDSF 1.6, the changes in OS/390 Release 2 to all monitors of RMF, and a Small Programming Enhancement (SPE) in support of the CFLEVEL=2 functions of the coupling facility.

How This Redbook Is Organized

This redbook contains 262 pages. It is organized as follows:

- Chapter 1, “OS/390 Release 2”

  This chapter provides an overview of OS/390 Release 2 features and functions.

- Chapter 2, “OS/390 Release 2 ServerPac Installation”

  This chapter describes the OS/390 Release 2 ServerPac installation process. It is a description of the process steps used in the installation of the ITSO OS/390 Release 2 system.

- Chapter 3, “Parmlib Concatenation Support”

  This chapter describes the enhancements for additional parmlib data sets at IPL time. Up to 10 additional data sets may be concatenated to SYS1.PARMLIB at IPL. An installation may, if desired, dynamically change

the logical parmlib by switching to another set of parmlib data sets through use of a new operator command. In addition, a new logical parmlib service allows MVS, vendor, and installation programs to allocate and free the logical parmlib without specifying the names of the concatenated data sets.

- Chapter 4, “LNKLST and LPALST Enhancements”

  This chapter describes the enhancements that have been made in OS/390 Release 2 for the LNKLST and LPALST.

- Chapter 5, “Symbolic Pre-Processor”

  This chapter describes the parmlib symbolic pre-processor tool, which is an ISPF based interactive dialog. This tool can show you what your system symbolics and parmlib members would be, given a hypothetical hardware and software configuration.

- Chapter 6, “OS/390 Release 2 Global Resource Serialization”

  This chapter describes the enhancements to Global Resource Serialization to use a star topology as an alternative to the current ring implementation.

- Chapter 7, “Coupling Facility Failure Policy”

  This chapter describes the enhancements to the coupling facility structures and the programming and operator interfaces.

- Chapter 8, “SMP/E Enhancements”

  This chapter describes the enhancements to SMP/E to reduce the complexity of an OS/390 installation with the ServerPac.

- Chapter 9, “SDSF”

  This chapter describes the new functions available with SDSF 1.6.

- Chapter 10, “RMF Enhancements in OS/390 Release 2”

  This chapter describes OS/390 Release 2′s changes to all monitors of RMF.

- Chapter 11, “Small Programming Enhancement for CFLEVEL=2”

  This chapter describes a Small Programming Enhancement (SPE) for OS/390 Release 2 in support of the CFLEVEL=2 functions of the coupling facility. This SPE is also known as CF Batched Unlocks.

- Appendix A, “GRS Star Implementation Checklist”

  This appendix describes a check list that makes an inventory of the analysis and activities to be followed while planning a migration to a GRS star configuration.

- Appendix B, “ENQ/DEQ/RESERVE Analysis Aid Reports”

  This appendix describes a tool that monitors supervisor calls 56 and 48 (ENQ/RESERVE/DEQ) and collects data about the resources serialized and the requesters.

- Appendix C, “ISGGREX0 Sample Exit”

  This appendix describes a sample exit ISGGREX0. The attached ISGGREX0 exit is based on the ISGGREXS in the SYS1.SAMPLIB. The sample exit detects volumes shared by systems outside the global resource serialization complex and prevents RESERVE conversion for those volumes.

- Appendix D, “SMP/E BUILDMCS Output Elements”

  This appendix describes each of the output elements associated with the SMP/E BUILDMCS process.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Poughkeepsie Center.

Bob Haimowitz	ITSO Poughkeepsie
Paul Rogers	ITSO Poughkeepsie
Ron Northrup	IBM South Africa
Ralph Rudd	IBM South Africa
Tapio Koskinen	IBM Finland
Benno Aladjem	IBM Spain
Franco Meroni	IBM Italy
Friedemann Hofmann	IBM Informationssysteme GmbH
Joaquim e Silva	Standard Bank South Africa

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. OS/390 Release 2

OS/390 is a set of MVS base, open, client/server, and applications-enabling functions orderable with one program number and delivered as one product.

OS/390 consists of base elements that deliver essential operating system functions. Most of the elements are products that have been available for some time and you have probably been running some of them. To distinguish the element from its prior or preceding product, the term root product can be used. The OS/390 level of an element can be any of the following:

- A repackaging of the root product
- The root product with some additional function
- The root product unchanged

In addition to the base elements, OS/390 has optional features that have an affinity to the base. There are two types of features:

- One type of feature is shipped with OS/390 whether you order the feature or not.

  Features of this type support dynamic enablement. If you order the features, they are shipped enabled; otherwise they are shipped disabled. If you later want to use a disabled feature, let IBM know and enable it dynamically through parmlib.

- The second type of feature does not support dynamic enablement and is not shipped with OS/390 unless you specifically order one or more features in addition to the base.

IBM′s direction is to have all optional priced features capable of being dynamically enabled and disabled and hence shipped with the base.

1.1 Exclusive and Non-Exclusive Elements and Features

Some elements and features contain new function that is available only within OS/390. Such an element or feature is termed exclusive; new function is exclusive to OS/390.

Other elements exist within OS/390 and also as separately orderable products. New function can be obtained within OS/390 or by ordering the separate product. Such elements are termed non-exclusive.

IBM′s direction is to make functional enhancements only within OS/390. Hence the number of exclusive elements and features will increase.

OS/390 Release 2 contains the following elements. Only those elements that are also available as stand-alone products are listed with the release level used in OS/390.

The new elements added in OS/390 Release 2 are:

- 3270 PC File Transfer Program
- FFST/MVS Version 1 Release 2
- DFSORT Release 13 (optional)
- SDSF Release 1.6 (optional)

- GDDM-PGF Version 2 Release 1.3 (optional)
- Softcopy Print

  OS/390 Release 2 includes a new Softcopy Print element that allows customers to print softcopy books at their locations. Softcopy print is another step in making OS/390 an information independent system, one in which users can meet all their information needs, from online information to hardcopy, through the system itself, independent of other sources.

  Softcopy print consists of an integrated subset of function from Print Services Facility (PSF) V2R2, Document Composition Facility (DCF) V1R4, and BookMaster R4, and some fonts in the APF Font Collection V1R1. This subset of function allows you only to print input files generated by the BookManager PRINT or COPY functions. You can print any IBM or non-IBM online book that can be viewed with BookManager READ.

1.2 OS/390 Release 2 Functions

The following functions are integrated into OS/390 Release 2.

1.2.1 Year 2000

OS/390 Release 2 is Year 2000-enabled. It provides the platform necessary to get started with a Year 2000-ready release, which helps to simplify many of the tasks of migrating each product separately to its Year 2000 compliant release level.

1.2.2 Parallel Sysplex Connectivity Enhancements

In MVS/ESA SP 5.1.0, Cross-system Extended Services (XCF) provided support to manage and use coupling facility structures. The coupling facility selection process has been enhanced through new Coupling Facility Failure Policy (CFFP) support, which allows both the exploiting application and the installation a greater amount of control in specifying their connectivity requirements. These enhancements provide:

- Improved coupling facility selection during structure allocation
- A way for applications to define their connectivity requirement during the initial connect
- For a rebuilt structure, a way of ensuring improved or equivalent connectivity to the systems in the sysplex

The enhancements affect the programming interface used by applications when connecting to a coupling facility structure and the programming/operator interface used when rebuilding a coupling facility structure.

The installation currently can specify its most important systems in a sysplex by assigning system weights in the Sysplex Failure Management (SFM) policy. If the installation SFM policy is active, CFFP now uses these weights as part of its selection algorithm when choosing a coupling facility in which to allocate or to rebuild a structure. Using these weights ensures:

- For an application connecting to a structure, that the application will not be impacted by having its structure allocated in a coupling facility that does not meet the application′s connectivity requirements
- For an installation, that the overall connectivity of the sysplex is influenced by the aggregate SFM weight of all systems in the sysplex

Structure rebuild processing enables the application or the installation to rebuild the structure in the same or another coupling facility. The CFFP enhancements to rebuild processing allow for an evaluation of whether the rebuild process should occur. If the rebuild will result in a reduction of system connectivity to the structure, the rebuild will not take place unless the application or the operator has specified that the rebuild should continue.

1.2.3 Global Resource Serialization Enhancements

OS/390 Release 2 introduces Global Resource Serialization (GRS) based on a star topology which uses the Coupling Facility (CF) for managing global resource serialization in a sysplex environment. This is an alternative to the Ring implementation currently used today. Using the CF, contention for global resources is maintained in a CF lock structure. Each system now maintains only its local view of the global resources. This implementation provides improved availability, as follows:

- Whenever a system fails, no action needs to be taken by the remaining GRS processing elements in the sysplex. This allows for much quicker recovery time for the systems remaining in the sysplex, as there is no longer a need to clean up the resource queues or determine a new ring topology for the remaining systems.
- GRS response time is improved by mapping ENQ/DEQ requests to XES lock structure requests, eliminating the need for and the overhead associated with a ring serialization mechanism.
- New CTRACE options have been added to provide more granular information for problem determination.

1.2.4 Dynamic Exit Facility Enhancements

Problem determination capability for an exit defined to the dynamic exit facility is being enhanced through an extension to the DISPLAY PROG,EXIT,.... command. An example of the command is as follows:

```
DISPLAY PROG,EXIT,EXITNAME=xx,DIAG
```

The diagnostic information that can now be displayed for an exit routine includes the entry point, load point, module length, and the jobname/stoken associated with the module when it was added.

1.2.5 PARMLIB Concatenation Support

OS/390 provides the capability to concatenate up to 10 additional data sets to SYS1.PARMLIB at IPL. An installation may, if desired, dynamically change the logical parmlib by switching to another set of parmlib data sets through use of a new operator command. In addition, a new logical parmlib service allows MVS, vendor, and installation programs to allocate and free the logical parmlib without specifying the names of the concatenated data sets.

1.2.6 LNKLST and LPALIST Enhancements

A new enhancement to the PROGxx parmlib statement allows libraries other than SYS1.LINKLIB, SYS1.MIGLIB, and SYS1.CSSLIB to be at the beginning of the LNKLST concatenation and allows SYS1.LPALIB to be at the beginning of the LPALST concatenation.

The LNKLST and LPALIST can be up to 255 extents when DFSMS 1.3 functions are present. LNKLST can be defined via PROGxx, rather than LNKLSTxx, and a conversion EXEC CSVLNKPR is provided.

1.2.7 SMP/E Enhancements

SMP/E is providing ease-of-use enhancements and support to reduce the complexity of OS/390 installation with the ServerPac.

OS/390 Release 2 provides support for program objects on RELFILEs by having SMP/E recognize the type of library (PDS or PDSE) from which the original RELFILE was produced and allocate the SMPTLIB data set corresponding to that RELFILE with the same DSNTYPE.

The GIMOPCDE member, which SMP/E used by default to determine valid OPCODES during the scanning of JCLIN, has been removed from PARMLIB. Instead, the default set of OPCODE definitions will be shipped with OS/390 Release 2 SMP/E.

1.2.7.1 BUILDMCS Command

A new BUILDMCS command provides a process for copying a product from one set of target and distribution zones and libraries to another set of target and distribution zones and libraries. The BUILDMCS command generates function SYSMOD images of specified FMIDs. These images incorporate all service and user modifications currently installed for the specified FMIDs. The function SYSMOD images can then be used as input to the RECEIVE, APPLY, and ACCEPT commands to install the product into a new set of target and distribution zones and libraries.

This enhancement helps facilitate the migration to a new system using the new system-replace vehicle, ServerPac. The BUILDMCS command makes it easier for customers to carry forward no-longer-marketed IBM products or SMP/E-supported ISV products rather than having to re-install them from scratch. They can either be copied to the new system environment or copied to a separate SMP/E environment, depending on product requirements.

To help prepare for the installation of OS/390 Release 2, the BUILDMCS command is also available for use in Release 8.1 of SMP/E via APAR IR32072 (PTFs UR45833, UR45834, and UR45836).

1.2.7.2 FMIDSET Specification on SELECT Operand

OS/390 Release 2 SMP/E provides additional granularity for FMIDSET specification on the SELECT operand of the APPLY, ACCEPT, RESTORE, and RECEIVE commands, which will allow a user to install sets of FMIDs.

1.2.7.3 Provide FIND Command Support in SMP/E Dialogs

OS/390 Release 2 provides FIND primary command support in the SMP/E base dialogs that allows a user to quickly determine where a specified character string appears within a table display panel.

1.2.7.4 Granularity for HOLDSYS on BYPASS Operand of APPLY/ACCEPT

OS/390 Release 2 SMP/E provides additional granularity for the HOLDSYSTEM option of the BYPASS operand of the APPLY and ACCEPT commands, which will allow a user to install specific SYSMODs that have a SYSTEM hold for a specific reason ID.

1.2.8 Hardware Configuration Definition (HCD) Enhancements

Besides the support for new hardware, HCD provides enhancements that make it easier to configure the OS/390 system and to create accurate configuration documentation:

- IODF reports support more granularity for IODF Compare reports (including limitation on LPAR level) and improves the readability of the reports. A user gets a better overview of configuration changes, and this is especially useful for configuration planning tasks. In addition, all IODF reports now show device number ranges.
- HCD reduces the number of messages during Build Production IODF, and this message collection results in a significantly shorter message list providing a better overview of changes.
- The Dynamic Activation changes provide information about devices, control units, and channel paths that are to be deleted, added, or changed during dynamic activation.
- Centralized IOGEN documentation is provided in a new report type that shows IOGEN information per device type, based on Unit Information Module (UIM) information. Together with other reports already provided by HCD, this information is useful for configuration planning purposes.
- HCD provides year 2000 support to now show the extended date format YYYY-MM-DD in all HCD reports and in the activity and message log file.

1.2.9 RMF Enhancements

With OS/390 R2, RMF introduces the following enhancements:

- The new RMF Postprocessor Cache Activity report provides cache statistics on a subsystem basis as well as on a detailed device-level basis. The control units supported are the current models in the 3990 control unit family, 3990-3 and 3990-6, and the RAMAC array subsystem 9394.
- The Postprocessor Coupling Facility Activity report has been enhanced significantly by providing data in a more usable format and by reporting additional data on coupling facility processors, paths, and structures.
- There are two new Monitor III reports that have been adapted from Monitor II reports:
  - Channel Path Activity report
  - I/O Queuing Activity report
- RMF supports the new parmlib concatenation capability introduced with OS/390 R2. This support allows for more flexibility in managing parmlib members, and better handling of the changes that can occur in members when a product level is upgraded.
- With OS/390 R2, GRS introduces an alternative Star concept to keep track of global resources shared between systems. RMF supports this new concept, and displays the GRS mode in the Enqueue Activity report.

- To handle the switch over to the next millennium, RMF expands the year to four digits where necessary in support of the year 2000.

Chapter 2. OS/390 Release 2 ServerPac Installation

Starting with OS/390 Release 1, IBM introduced the IBM ServerPac for OS/390 to install the new OS/390 package. The ServerPac is the recommended system replacement vehicle and comes free with your OS/390 license. There is no Custom Built Installation Process Offering (CBIPO) package for OS/390.

For non-OS/390 products that you omit from the ServerPac, you must order a CBPDO and install it after the ServerPac installation. OS/390 consists of base elements that deliver the basic operating system functions.

In addition to the base elements, OS/390 provides two types of optional features. One type is shipped with OS/390 whether you order the feature or not. This feature can be dynamically enabled through a parmlib member. The second type of feature does not support dynamic enablement and is not shipped with OS/390 unless you order it specifically in addition to the base.

For more information about the elements and features in OS/390 Release 2, see *OS/390 Up and Running Release 2*.

With the ServerPac, OS/390 is delivered in a dump-by-data set format and provides a system which can be IPLed after the installation process. All IVPs are checked by IBM prior to shipment. During the OS/390 systems integration test, all IBM subsystems will have been tested.

In this chapter, the following sections describe the installation of OS/390 Release 2 using an IBM ServerPac for OS/390.

- ServerPac contents
- Which publications to use
- Installation prerequisites
- Installation planning
- Installation steps
- Installation summary

2.1 ServerPac Contents

The shipment for an IBM ServerPac contains the following items:

- Customized deliverables (depending on your order) on 3480 cartridges
- A tape with related installation material (RIM)
- A service tape
- A WEB server tape
- Diskettes for LanServer, LanRes, TCP/IP Offload, and VisualLift ADE
- The following documents are available with ServerPac:

  ServerPac Guide and Worksheet

  MVS CB Offering User Guide

  MVS Customized Offerings

2.2 Publications Used for Installation

The following publications were used during the ServerPac installation:

IBM ServerPac for OS/390 Guide and Worksheet

ServerPac Dump by Data Set Format Installation Guide

CustomPac Installation Dialog Reference Manual

CustomPac Installation Dialog Messages Book

OS/390 Up and Running Release 2

OS/390 Software Management Cookbook

OS/390 Up and Running Release 2 describes OS/390 and takes you through the necessary planning and installation steps. This book is intended for MVS users planning to migrate to OS/390. It also gives you an overview of the installation tasks involved when using an IBM OS/390 ServerPac.

You should also refer to the PSP buckets for

Upgrade **CUSTOMPAC** subset **SERVERPAC**

before you start with the Installation.

The product program directories are in the data set IPO1.PGMDIR.

Before going into production, check the product PSP buckets to obtain the latest maintenance level.

2.3 Installation Prerequisites

Before you start the installation, check to ensure that the following minimum hardware and software prerequisites are installed:

Hardware Any hardware that can run MVS such as a processor complex, tape drives capable of reading tape cartridges, DASD drives, and terminals.

Software A driving system software such as a running MVS system.

The following products must be installed on the driving system:

Table 1. Driver System Program Products	
Program Product	**Minimum Level**
MVS/SP	V3.1.3
MVS/DFP or	V3R1
DFSMS/MVS	V1R1
ICKDSF	R16
ISPF/PDF	V3R3 or higher
TSO/E	V2R4
VTAM	V3R4.2
SMP/E	V1R8.1 with PTF UR44006
HLASM	V1R2 with PTF UN89088

Considerations for the following program products are:

ICKDSF ICKDSF is also provided by the RIM tape.

IEBCOPY IEBCOPY is used to restore the tapes to DASD. It is strongly recommended to have IEBCOPY at a current level to avoid problems when using the IEBCOPY COPYMOD function. Refer to your *ServerPac Dump by Data Set Installation Guide* to find out the minimum service level for IEBCOPY.

SMP/E SMP/E V1R8.1 is needed on the driving system if you want to apply any maintenance, or to install customization with SMP/E.

HLASM HLASM V1R2 is required to assemble any source code.

Always refer to current editions of *CustomPac Installation Dialog Reference Manual* and *CustomPac Dump by Data Set Installation Guide* for driving system requirements.

2.3.1 Installation Planning

Since the ServerPac process is a system replacement, you should carefully plan the installation. Review your current environment as follows:

- The system layout
- The catalog structure
- Data set naming conventions
- Security software considerations

Depending on your order, the system target and DLIB data set may exceed more than one 3390-3 volume. You should define your new system layout to be prepared for future ServerPac installation and easy cloning of your system.

Use the worksheets from Appendix B in *IBM ServerPac for OS/390 Guide and Worksheet*, and define where the following new data sets should reside:

- Target data sets
- DLIB data sets
- Master catalog and user catalog

- Dialog and order data sets

The following should also be defined for your new system:

- Data set names
- Alias names and user catalog relationships
- New standards if necessary

2.4 Installation Steps

This section describes the installation steps which are provided by the installation dialog. The *IBM ServerPac for OS/390 Guide and Worksheet* leads you through the dialog-based installation.

Because ServerPac is one of the CustomPac offerings, you might see the term CustomPac in this documentation, as well as in the ServerPac documentation and on the dialogs.

2.4.1 Load RIM Tape

The installation of the CustomPac dialogs is described in *IBM ServerPac for OS/390 Guide and Worksheet*. You must install the CustomPac dialogs from the RIM tape. Once they are installed, the CustomPac dialogs do not have to be reinstalled with every order. They may be updated when initiated by IBM whenever you get a new order. Version checking invokes the update of the dialogs during the CustomPac RECEIVE function.

Throughout the installation of the dialogs, you are requested to define a CustomPac qualifier or the HLQ of your master data sets. Since the dialogs are permanently installed at your installation, you should not specify the IBM-supplied order number in the CustomPac qualifier.

2.4.2 Installing the CustomPac Dialogs

The installation of CustomPac dialogs is described in *IBM ServerPac for OS/390 Guide and Worksheet* and *CustomPac Installation Dialog Reference Manual*.

The RIM tape contains the following sample procedures, JCL, jobs, and CLISTs:

LOADRIM LOADRIM is the JCL to unload files from tape and the setup of the installation dialog. When you edit the LOADRIM sample JCL, you can choose the name of the master data sets, the unit name of your tape drives, and the VOLSER of the DASD which receives the installation dialog data sets. For the master and order data sets, you should use different HLQs.

SETUP SETUP is a sample LOGON procedure which includes the CustomPac dialog ISPF libraries.

CPPCSAMP The CPPCSAMP sample CLIST can be used to set up the environment instead of modifying the LOGON procedure. CPPCSAMP uses LIBDEFs and should be the preferred method to allocate the CustomPac libraries and start the dialog. CPPSAMP can be used after invoking ISPF.

CPPCINIT With the CPPCINIT CLIST, you can set up the environment from native TSO.

PRTDOC PRTDOC is a sample job to print the CustomPac installation dialog reference manuals.

2.4.2.1 CustomPac Dialog Considerations

All Information related to the CustomPac installation dialogs can be found in *CustomPac Installation Dialog Reference Manual*.

HELP (PF1) is available on any panel. The HELP key is a very useful online help facility that explains panel functions in detail. Some panels have PRIM and LINE commands available. Using the HELP key allows you to get a description and example of how to use the commands.

Chapter 4 of the *IBM ServerPac for OS/390 Guide and Worksheet* publication describes the tailoring options that are provided with the CustomPac dialogs.

2.4.3 Receiving the ServerPac Order

Invoke the CustomPac CLIST **CPPCSAMP** to start the dialog. Receiving the order means you will copy the order from tape to DASD.

```
CustomPac ---------------- (C) IBM Corporation 1990-1996 ---------------
OPTION ==>  R

CustomPac INSTALLATION - Select The Option Of Your Choice

       The HLQ of your MASTER data set is : SYSPOK.MASTER


       R   RECEIVE      - Receive an Order

       I   INSTALL      - Install Orders

         ==> 0T000014 (Order Nbr OR blank for ALL NEW Orders)

       O   ORDER        - Order Information
```

Figure 1. CustomPac Installation Panel

Panel Activities

R receives the order. This unloads the control tables and installation jobs from the shipment tapes to your DASD. Selecting option **R** selects the Order Receive panel shown in Figure 2 on page 12.

2.4.4 Order Receive Panel

After completion of the Receive option, a batch job is generated and submitted to download the order installation libraries from the shipment tape to DASD.

```
CustomPac ---------------------- Order Receive ------------------------
COMMAND ==>

ORDER DETAILS

        Order Number  ==> OT000014

        TAPE VolSer   ==> R0014A TAPE Unit  ==> 3490

        Order HLQ     ==> SYSP0K.OT000014

        DASD VolSer   ==> T0TTS2 DASD Unit  ==> SYSDA


        Do You Want To Use VB Clists       ==> N

        Edit JOB Stream Before Submitting  ==> N
```

Figure 2. Order Receive Detail Panel

Panel details

Order Number This is your specific IBM-supplied order number, which is listed on the cover of the order documentation.

TAPE VolSer This is the volume serial number of the RIM tape.

TAPE Unit This is the unit type of your tape drives.

Order HLQ This is the HLQ used to allocate the order installation data sets. It is recommended to include the order number as part of the qualifier.

DASD VolSer This is the VOLSER of the DASD where your order data sets are to be restored.

DASD Unit This is the unit type of your DASD units and is defaulted to SYSDA.

After you press the Enter key, the generate job stream panel appears, as shown in Figure 3 on page 13.

```
CustomPac ----------------------- Order Receive ------------------------
COMMAND ==>

GENERATE JOBSTREAM

   Enter JOBCARDs

 > //FHOFMA   JOB (XXXX),'PROGRAMMER NAME',
 > //         CLASS=A,MSGCLASS=X,
 > //         NOTIFY=FHOFMA
 > //*

   Installation  ISPLLIB ==> ISP.V4R2M0.SISPEXEC
                         ==>
       ISPF      ISPMLIB ==> ISP.V4R2M0.SISPMENU
                         ==>
     Libraries   ISPPLIB ==> ISP.V4R2M0.SISPPENU
                         ==>
                 ISPSLIB ==> ISP.V4R2M0.SISPSENU
                         ==>
                 ISPTLIB ==> ISP.V4R2M0.SISPTENU
                         ==>
```

Figure 3. Generate Job Stream Panel

Enter the job card information relating to your installation standards. Change the ISPF library names to your current ISPF environment.

After pressing the Enter key, you enter a panel where you can specify additional job card information for loading of the RIMs. Depending on whether you have previously indicated that you wanted to edit the job stream before submission, you can now review and edit the generated job that is to receive the order, then submit it.

After successful completion of the job, your order data sets are copied from the RIM tape to your DASD.

Once you have selected an order for processing, an enqueue is issued against the order number. This ensures that only one person can work on an individual order at any one time.

After you finish the order receive function, place an **I** on the CustomPac installation panel, shown in Figure 1 on page 11, to start the installation of the order. On the Order installation panel, shown in Figure 4 on page 14, you can select the order you want to install. If this is your first ServerPac installation, only one order number can be selected.

```
CustomPac --------------- ORDER INSTALLATION ----------------- Row 1 to 1 of 1
COMMAND ==>                                                    SCROLL ==> PAGE

PRIM Cmds:(? SET L F N P REFresh STatus SORT VErbose VERsion)
LINE Cmds:<Copy Delete Edit Finalise Insert Products Report Select>

   ORDer    PROFile   SYStem         PAC       Prod       Change  Change
S  Number   Number     Name    SREL  TYPE      DATE    St  USER   DATE
- -------- --------- --------- ---- ------- ---------- -- ------- ----------
S OTOOOO14            OS390R2  Z038 EXP     01/08/1996 S  FHOFMA  20/08/1996
************************ Bottom of data ************************************
```

Figure 4. Order Installation Panel

2.4.5 Installation Dialog

After selecting a ServicePac order, the main installation dialog panel is invoked, as shown in Figure 5.

```
CustomPac --------------- (C) IBM Corporation 1990-1996 ----------------
OPTION ==> C

Installation

Order ( OTOOOO14 ) The Following Functions MUST be Executed in Sequence

   C    Configure        Select Configuration for Installation and Merge
   *    Variables        Define Installation Variables
   *    Zones            Define Zone Configuration
   *    Modify           Modify System Layout
   *    Alias            Define Alias to Catalog Relationships
   *    SSA              Define SSA to Catalog Relationships
   *    Installation     Select and Submit Installation Jobs
   *    Save             Save Used Configuration
   *    Update           Update Order Inventory Status
   DI   Display          Display a list of data set names
```

Figure 5. Installation Dialog Panel

When this panel is shown for the first time, the only option which may be selected is option **C**. Each of the following functions now marked with an ***** become available after the previous function has successfully finished.

Panel Options

C Option C on this panel allows you to select a configuration for merging an initial installation. If this is your first CustomPac installation, the Create Configuration panel appears, as shown in Figure 6 on page 15.

DI Option DI allows you to display any data set names and is similar to the PDF Option 3.4 function.

2.4.6 Selecting a Configuration for the Order

```
CustomPac ------------ CREATE Configuration ( OTOOOO14 ) ----- Row 1 to 1 of 1
COMMAND ==> CR                                                 SCROLL ==> PAGE

Select Configuration

PRIM Cmds:(? SET L F N P SORT CReate)
LINE Cmds:(Select)

hp2.S CONFiguration                      Comment
- ---------------------------------- --------------------------------
* SYSPOK.OTOOOO14                    Always Selected for Order
- ---------------------------------- --------------------------------
  SYSPOK.D96219                      MASTER Configuration
- ---------------------------------- --------------------------------
  ** NO MERGABLE ORDER CONFIGURATIONS
******************************** Bottom of data ************************
```

Figure 6. Create Configuration Panel

Before you start the installation, you must select and create a configuration. On the Create Configuration panel, you can see the master configuration and, if available, other saved configurations. If there is no previous configuration, you cannot merge with the current order.

Enter a **CR** in the command line to create a work configuration.

Type an **S** in front of the configurations you want to merge, if applicable.

2.4.7 Define the Installation Variables

The Define Installation Variable panel can now be selected by typing a **V** option on the Installation panel shown in Figure 5 on page 14. This takes you to the Installation Variables panel shown in Figure 7 on page 16.

```
CustomPac ----------- Installation Variables ( OTOOOO14 ) -- Row 1 to 14
COMMAND ==>                                                  SCROLL ==> PAGE

Variable Selection List                                        SHOW( —C

PRIM Cmds:(? SET L F N P CANcel SAVE SHow VARname)
LINE Cmds:(Browse Delete Edit Insert Repeat Ship)

 S     Synonym            STA  Contents
 - --- -----------------  ---  ----+----0----+----0----+----0----+----0-
   ==> GEN SYSTEMPAC
       ASSEMBLER NAME       D  ASMA90
       ASSEMBLER NAME       D  ASMA90
       VVDS SPACE           D  CYLINDERS(2 2)
       SMPTLIB PREFIX       D  SYSPOK.SMPTLIB
       ASSEMBLER NAME       D  ASMA90
       SMPTLIB UNIT         D  3390
       SMPTLIB VOLSER       D  OS3CAT
            ...
       UNIT OLD CATVOL      D  3390
       UNITNAME OLD DASD    D  SYSDA

   ==> MVS OLD VOL
       VOLSER OLD RES       P  OS3R2T
       VOLSER OLD CAT       D  TOTCAT

   ==> GENERAL DSN
       AUTH. LINKLIB        D  SYS2.LINKLIB
       RIM LOAD LIB         P  SYSPOK.OTOOOO14.LOADLIB
       WORK DASD UNIT       D  3390
       WORK DASD VOLSER     D  TOTTS4
       ORDER HLQ            P  SYSPOK.OTOOOO14
            ...
```

Figure 7. Installation Variables Panel

Panel Activities

Verify the current contents and enter or change any values by overtyping in the Contents column if a value is either missing or invalid.

The installation variables can be different for each order. They are stored in the installation variables table (IVT), which is a CustomPac-generated ISPF table shipped with your order. The installation variables are briefly described in Appendix B (Worksheets for ServerPac Installation) of the *IBM ServerPac for OS/390 Guide and Worksheet* publication.

It is recommended that you read and use the worksheets before changing any installation variable values.

The variable for AUTH.LINKLIB may be an existing authorized library of your installation site.

You may use the VAREdit command on some panels to change the installation variables later.

2.4.8 Defining SMP/E ZONE Names

Back on the Installation Dialog panel (see Figure 5 on page 14), you may now select option **Z** to define your SMP/E zone configuration. This panel is shown in Figure 8.

```
CustomPac ----------- Define ZONE Information ( OTOOOO14 ) --- Row 1 to 1 of 1
COMMAND ==>                                                   SCROLL ==> PAGE

PRIM Cmds:(? SET L F N P SORT CANcel SAVE SHIP)
LINE Cmds:(NONE)

ALL DLIB Zones will be allocated in CSI : SMPE.OSR2DZN.CSI
ALL TLIB Zones will be allocated in CSI : SMPE.OSR2TZN.CSI

For (Sub)System  : MVS   ( Z038 )

                 NickName        Dlib Zone        Target Zone
                 --------        ---------        -----------
                   100            MVSD100           MVST100
                   200            MVSD200           MVST200
```

Figure 8. Define ZONE Information Panel

Panel Activities

This panel is displayed even if you do not plan to change the shipped zone names. You can change the zone names to the names you want.

The reason for having more than one target and DLIB zone is that you cannot have incompatible products together in one SMP/E zone, such as COBOL/II and OS/COBOL.

Figure 8 shows two DLIB and target zones. The nickname is used to pair them together.

The naming conventions for the SMP/E zone names shipped with your order are described in *IBM ServerPac for OS/390 Guide and Worksheet* under Defining Zone Configuration Table in Chapter 3. Refer also to the SystemPac/MVS SMP/E Zone Structure section in *ServerPac Dump by Data Set Format Guide*.

Use the SHIP command with caution, because it restores all DLIB and target zone names to their shipped value.

2.4.9 Define System Layout

The Installation Dialog panel (see Figure 5 on page 14), allows you now to select **M** for modifying your target system layout. This displays the Modify System Layout panel shown in Figure 9 on page 18.

```
CustomPac ------------ Modify System Layout ( OTOOOO14 ) -- Row 1 to 14
COMMAND ==>                                                SCROLL ==> PAGE

SUMMARY Of Products

PRIM Cmds:(? SET F L N P SORT ALL CANcel SAVE SHIP DEVT SUML SUMP)
LINE Cmds:<Dslist Select>

        S  Product                                VSLname   data sets
        -  ---------------------------------------- --------  --------
           AD/CYCLE C/370        1.02.0           ADCE120       21
           AD/CYCLE CODE/370 ENU 1.02.0           ACCE120       13
           ADSM FOR MVS          2.01.0           DSMB210       10
           AOC OPC  AUTO         1.04.0           AOCO140       14
           AOC/CICS AUTO         1.04.0           AOCC140       19
           AOC/IMS AUTO          1.04.0           AOCI140       23
           AOC/MVS               1.04.0           AOCB140       43
           AON/MVS BASE          1.01.0           AONE110       18
            ...
           OS/390 BCP                             O32B120       94
        D  OS/390 ISPF                            O32F120       33
           OS/390 JES2                            OS3J120       15
           OS/390 LE 1.02.00                      OSLE120       38
           OS/390 OE APPL SVCS                    O32E120        4
           OS/390 SMP/E                           O3SM120       11
           OS/390 SOFTCOPY                        OSSP120       23
           OS/390 SOM R/T                         O32S120       21
           OSA/SF FOR MVS        1.02.0           OSAB120       18
           RACF SECURITY SERVER 1.1.0             OS3V110       10
           REXX COMP/370         1.03.0           REXC130       18
            ...
```

Figure 9. Modify System Layout Panel

Panel Activities

Defining the target system layout is one of the most important steps during the order installation. You should use the Modify System Layout Sample Worksheet from Appendix B of *IBM ServerPac for OS/390 Guide and Worksheet* before you enter any information on the Modify System Layout panel.

This panel shows you the summary of all products within your order. From here you start your customization of the individual products. You may modify the following information about the target, SMP/E, and catalog data sets:

- Product data set names, placement and attributes
- Logical volume to physical volume relationship
- Physical volumes device type, address and volser

The **PRIM Cmds** and **LINE Cmds** on this panel give you greater flexibility in defining your new environment.

Read and use the section Modify the System Layout in Chapter 3 of *IBM ServerPac for OS/390 Guide and Worksheet*. Refer to *CustomPac Installation Dialog Reference Manual* whenever you need more detailed information about the dialogs.

The *ServerPac Dump by Data Set Format Installation Guide* comes with your order and contains all information relating to the products to be installed.

If this is the first time the Summary of Products panel has been displayed, you should run the **SUMP** command. This enables you to define the physical device address of each physical volume.

Figure 10, Figure 11 on page 20, Figure 12 on page 20, and Figure 13 on page 21 show you the major functions of the Modify System Dialog.

The **Dslist** command displays the data set list by product, as shown in Figure 10.

```
CustomPac ------------- Modify System Layout ( OTOOOO14 ) -- Row 1 to 14
COMMAND ==>                                                 SCROLL ==> PAGE

data set LIST PRODUCT : OS/390 ISPF

PRIM Cmds:(? SET L F N P SORT CHange OFile OList VErbose)
LINE Cmds:(Attribs Space)

S DSName                                          F RECFM DSORG LRECL BLK
- ----------------------------------------------   - ----- ----- ----- ---
  ISP.AISPALIB                                       FB    PO      80  88
  ISP.AISPCLIB                                       FB    PO      80  88
  ISP.AISPEXEC                                       FB    PO      80  88
  ISP.AISPGENU                                       FB    PO      80  88
  ISP.AISPGMLI                                       FB    PO      80  88
  ISP.AISPGUI                                        VB    PO     255  88
  ISP.AISPHELP                                       FB    PO      80  88
    ...
```

Figure 10. Data Set List by Product Panel

If you selected product OS/390 ISPF on the Modify System Layout main panel in Figure 9 on page 18, all data sets for product OS/390 ISPF are displayed. The PRIM command CHange allows you to make global changes to data set profiles. For example, you may change the HLQ for those product data sets. Before using the CHange command, refer to the *CustomPac Installation Dialog Reference* and read Chapter 13.

The line commands A and S allow you to change data set names, logical volumes, space, and BLKSIZE definitions for a specific data set profile. The Select command entered next to a product on the Modify System Layout main panel in Figure 9 on page 18, displays a summary of all logical volumes for the selected product (as shown in Figure 11 on page 20).

```
CustomPac ------------- Modify System Layout ( OTOOOO14 ) ---- Row 1 to 1 of 1
COMMAND ==>                                                  SCROLL ==> PAGE

Logical Volume By PRODUCT

PRIM Cmds:(? SET L F N P SORT)
LINE Cmds:<Assign Dslist>

PRODUCT  : OS/390 ISPF

   S  LVol       CYLs     Largest     data sets        PVol       DEV n
   -  ------     ----     -------     ---------         ------     -----
      DLBO67      128        35           16            OS3DL1     3390-
      RESO67      126        26           17            OS3RS1     3390-
****************************** Bottom of data ************************
```

Figure 11. Logical Volume by Product Panel

The line command Assign allows you to assign all data set profiles for the selected logical volume to a different logical volume. LVol name DLBxxx stands for a DLIB Volume. LVol name RESxxx stands for a residence volume.

The line command Dslist displays all data sets for the selected logical volume. You can make global changes to the data set profiles as described with the Data Set List by Product Panel.

It is also possible to add your own user-defined data set profiles. To do this, return to the Modify System Layout Main panel, shown in Figure 9 on page 18. Then enter the PRIM Cmds ALL U and the following panel, shown in Figure 12, is displayed:

```
CustomPac ------------- Modify System Layout ( OTOOOO14 ) ---- Row 1 to 1 of 1
COMMAND ==>                                                  SCROLL ==> PAGE

data set LIST All User Defined data sets

PRIM Cmds:(? SET L F N P SORT CHange OFile OList VErbose)
LINE Cmds:(Attribs Space Insert Delete)

S DSName                                        F RECFM DSORG LRECL BLK
- ----------------------------------------      - ----- ----- ----- ---
I
****************************** Bottom of data ************************
```

Figure 12. List All User Defined Data Sets Panel

The line command I displays a panel where you can define all the information needed to allocate a data set. See the *CustomPac Installation Dialog Reference Manual* in Chapter 13 “Insert a User Defined Data Set” for details.

Before you leave the Modify Sytem Layout Main panel, shown in Figure 9 on page 18, you should enter the SUMP PRIM command, which displays a summary of the physical volumes, as shown in Figure 13 on page 21.

```
CustomPac ------------- Modify System Layout ( OTOOOO14 ) -- Row 1 to 1 of 1
COMMAND ==>                                                  SCROLL ==> PAGE

SUMMARY Of Physical Volumes

PRIM Cmds:(? SET L F N P SORT DEVT)
LINE Cmds:<Assign Dslist>

      S  PVol     CCUU   DEV name         CYLs    Largest       data setss
      -  ------   ----   --------         ----    -------       -------
         OS3CAT   OCF1   3390-3           2198       261            21
         OS3DL1   OCF2   3390-3           2780       138            60
         OS3RS1   OCF0   3390-3           3010       170            63
         OS3RS2   OFF0   3390-3            942       254            20
****************************** Bottom of data ************************
```

Figure 13. Summary of Physical Volumes Panel

When one of your physical volumes becomes overallocated, the following message appears on the panel:

```
-------------------------------------------------------------
| CPP06050O5S At least ONE PHYSICAL Volume is OVER ALLOCATED |
-------------------------------------------------------------
```

This condition is also shown by the <<<<<<< next to the physical volume names.

By using the dialogs previously described, you are able to modify the system layout and correct the overallocation of physical volumes.

Important

Use the SHIP command with care because it is powerful. This command is available on several dialog panels. It can be used to restore all profiles to their initial-ship values. You can lose all the customization you previously entered if the SHIP command is issued without considering these ramifications.

2.4.10 Define ALIAS to Catalog Relationships

```
CustomPac -------------- ALIAS to CATALOG ( OTOOOO14 ) ----- Row 1 to 14
COMMAND ==>                                                  SCROLL ==> PAGE

Define CATALOG data set Names

PRIM Cmds:(? SET L F N P SORT CANcel SAVE)
LINE Cmds:(Delete Insert Repeat)

        S  Alias    STA TARGET System Catalog DSName               Ty
        -  -------- --- ----------------------------------------  --
           ADSM         ???????.CATALOG
           AOCIMS       CATALOG.TOTICF1.VOS3RS2
           AOCOPC       CATALOG.TOTICF1.VOS3RS2
           ASMA      M  CATALOG.TOTICFM.VOS3CAT                   MCAT
           ASMT         CATALOG.TOTICFM.VOS3CAT                   MCAT
           BFS       M  CATALOG.TOTICFM.VOS3CAT                   MCAT
            ...
```

Figure 14. Define Catalog Data Set Names Panel

Panel Activities

Specify the catalog data set name for each **ALIAS**.

Before you use the panels shown in Figure 14 and Figure 15 on page 23, you should read the related sections in *IBM ServerPac for OS/390 Guide and Worksheet* and *CustomPac Installation Dialog Reference Manual* to become familiar with using system specific aliases (SSA) and the catalog structure.

Appendix B of *IBM ServerPac for OS/390 Guide and Worksheet* contains the worksheets to be used for alias-to-catalog and SSA-to-catalog specifications.

Use the Alias to Catalog panel to specify which HLQ you want to be associated with a catalog. An **M** in the **STA** column indicates that this alias name must be associated with a master catalog.

The ?????? in the TARGET System Catalog DSName field indicates that there is no catalog defined yet.

This function allows you also to insert additional user-defined alias names and catalogs.

After specifying the alias-to-catalog relationship, you may select **SSA** on the Installation Dialog panel shown in Figure 5 on page 14, which leads you to the SSA to Catalog panel.

2.4.11 Define System-Specific Alias Names

```
CustomPac --------------- SSA to CATALOG ( OTOOO014 ) -------- Row 1 to 1 of 1
COMMAND ==>                                                    SCROLL ==> PAGE

CATALOG Selection List

PRIM Cmds:(? SET L F N P SORT CANcel SAVE)
LINE Cmds:<Select>

S TARGET System Catalog DSName           SSA Name Type VOLser Alloc Def
- ------------------------------------- -------- ---- ------ ----- ---
  CATALOG.TOTICFM.VOS3CAT                MOSR2    MCAT OS3CAT   Y    Y
  CATALOG.TOTICF1.VOS3RS2                UOSR2    UCAT OS3RS2   Y    Y
```

Figure 15. SSA to Catalog Panel

Panel Activities

The ServerPac installation process uses the system specific alias (SSA) technique for data set allocation during the installation jobs. This allows you to work conveniently with new data sets that have the same name as those on your existing system, for example SYS1.LINKLIB. It is recommended to specify **Y** in the Def column for defining the SSAs. The SSAs are removed by a cleanup job after you have successfully installed the ServerPac.

This is the end of the customization steps for the ServerPac. You are now ready to run the supplied installation jobs. The Installation Jobs panel, shown in Figure 16 on page 24, is initiated by entering an **I** on the Installation Dialog panel in Figure 5 on page 14.

2.4.12 Run the ServerPac-Provided Installation Jobs

Figure 16 is an example of specific jobs and documentation steps that will be followed in the installation process.

```
CustomPac -------------- Installation JOBs ( OTOOOO14 ) --- Row 1 to 14
COMMAND ==> GEN                                          SCROLL ==> PAGE

JOB Selection List                                          SS$( EXCL

PRIM Cmds:(? SET L F N P GENskel SUMmary SS$ VAREdit)
LINE Cmds:(Backup Delete Edit Insert Log Output Select SS-block Vars)

S      Description                              STEP     MC STAtus
-- --- ---------------------------------------  -------- -- ----------
   SRC DEFAULT JOBCARD
   DOC CUSTOMPAC UTILITY ENU 1.1.1
            ...
   DOC OE APPLICATION SERVICES
            ...
   ==> PACKAGE SPECIFIC INSTALL
   DOC PACKAGE INSTALLATION STEPS
   DOC OS/390 OE SYSTEM SERVICES
            ...
   ==> PRODUCT SPECIFIC INSTALL
   DOC PRODUCT SPECIFIC INSTALL STEPS
   DOC RMF 1.2.0
   JOB ASSEM/LINK RMF FOR JES2                  HRM6602G 08 JOB14172
            ...
   DOC RMF 1.2.0
   JOB BUILDING OBJECT LIBRARY DIRECT.          HMWL610N 00 JOB17043
   DOC OS/390 1.2.0 LANGUAGE ENVIRONMEN
            ...
   JOB LINK EDIT AND RUN C PROGRAM              HMWL610R 00 JOB17052
   JOB PRELINK LINK EDIT AND RUN C              HMWL610S 00 JOB17053
            ...
   ==> POST INSTALLATION
   DOC POST INSTALLATION TASKS
            ...
   DOC OS/390 LICENSE FEATURE
   JOB ICQPOST1 TABLE COPY JOB                  HTE25D2I 00 JOB14481
   DOC VTAM FOR MVS/ESA 4.03.0
   JOB LINKEDIT APPC APPL. SUITE                HVT4301F 04 JOB14506
            ...
   ==> CUSTOMIZATION SECTION
   DOC CUSTOMIZATION TASKS
   DOC CUSTOMIZATION SECTION START
            ...
   ==> INSTALLATION VERIFICATION SECTION
   DOC RUN IVPS TO CHECK SYSTEM INSTALL.
            ...
   ==> CLEANUP JOBS
   DOC INST. CLEANUP PROCEDURES
   JOB UNCATALOG VVDS FROM DRIVING MCAT         DELVVDS  00
            ...
   ==> MIGRATION SECTION
   DOC MIGRATION
   DOC MASTER CATALOG CONVERT
   JOB CREATE MASTER CAT CONVERT STEP           CATCVTM  00 JOB17167
            ...
```

Figure 16. Installation Jobs Panel

Panel Activities

When you enter this panel for the first time, the installation jobs have still not been generated. All installation jobs are generated using ISPF file tailoring services. The GENskel command submits a batch job, which generates all the installation jobs. Each job is stored in the SCPPBENU data set that is provided through the SeverPac RECEIVE process. The Generate File Tailored Installation Jobs panel is shown in Figure 17.

```
CustomPac -------------- Installation JOBS ( OTOOOO14 ) ----------------
COMMAND ==>

GENERATE File Tailored Installation Jobs



         This function generates a BATCH job which will file tailor
         ALL Installation Jobs in one pass, and save the jobs to the
         BACKUP data set.

         If a job already exists in the backup data set

         REPLACE Job  ==> Y   (Y or N)
```

Figure 17. Generate File Tailored Jobs Panel

As shown in Figure 16 on page 24, there are three types of components:

SRC Source data such as parameter lists

DOC Documentation

JOB Executable JCL

The installation steps are grouped into the following sections:

- Package-specific installation
- Product-specific installation
- Post installation
- Additional post installation
- Customization section
- Installation verification section
- Cleanup jobs
- Migration section
- Customer specific customization

The installation jobs should be submitted in sequence. Always read the DOC section before you select and submit the related jobs.

All installation steps and jobs are also described in *ServerPac Dump by Data Set Format Installation Guide*.

After a job's completion, the job output can be seen using the Output line command. This also updates the STAtus column in the Installation Jobs panel.

The job, copying data sets to SystemPac Vols (RESTORE), may run for a long time, depending on the number of products your SeverPac order contains. You should have two tape drives and all the tape cartridges shipped with your order available before you start the RESTORE job.

Post-installation and customization is product- and installation-dependent, and should be related to your specific requirements.

You can insert your own defined jobs to these dialogs.

After the installation jobs have completed, you should be able to IPL and test your new OS/390 System. To complete the installation, you may go back to the Installation Dialog panel shown in Figure 18.

2.4.13 Save Used Configuration

After the successful installation of your ServerPac, you may save your configuration by typing **S** on the Main Installation panel, shown in Figure 18.

```
CustomPac --------------- (C) IBM Corporation 1990-1996 ----------------
OPTION ==>

Installation

Order ( OT000014 ) The Following Functions MUST be Executed in Sequence

   C    Configure        Select Configuration for Installation and Merge
   V    Variables        Define Installation Variables
   Z    Zones            Define Zone Configuration
   M    Modify           Modify System Layout
   A    Alias            Define Alias to Catalog Relationships
   SSA  SSA              Define SSA to Catalog Relationships
   I    Installation     Select and Submit Installation Jobs
   S    Save             Save Used Configuration
   U    Update           Update Order Inventory Status
   DI   Display          Display a List of data set Names
```

Figure 18. Installation Dialog Panel

The configuration is saved in a data set shown in Figure 19 on page 27.

As the last step of the installation, you should update the inventory. This is done by entering a **U** on the Main Installation panel, as shown in Figure 18.

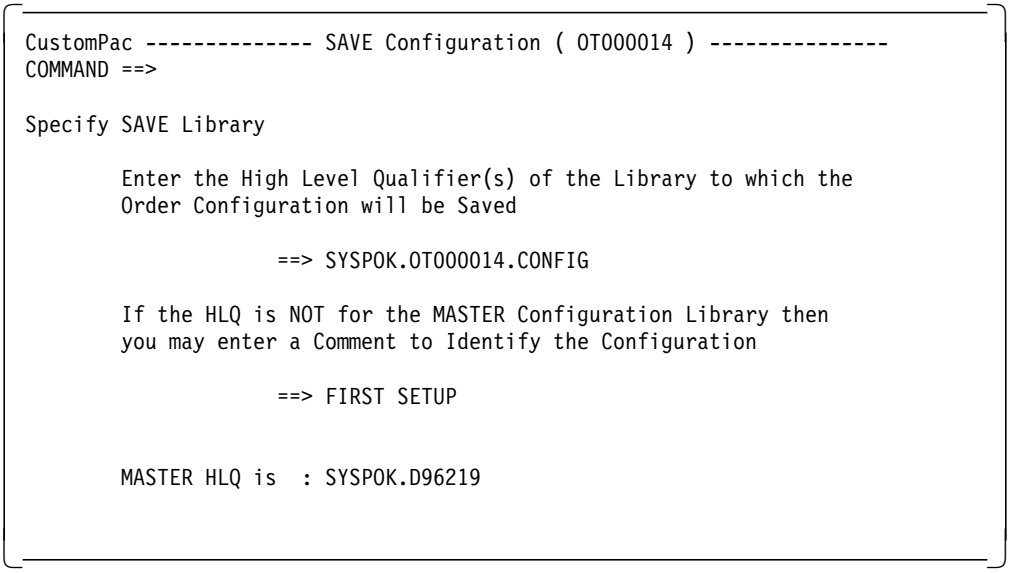


Figure 19. Save Configuration Panel

Chapter 3. Parmlib Concatenation Support

OS/390 Release 2 provides the capability of having a logical concatenation of SYS1.PARMLIB data sets, referred to as a logical parmlib. This concatenation is specified in the LOADxx parmlib member used at IPL time. OS/390 Release 2 also provides the capability to dynamically switch to another set of parmlib data sets via an MVS command. A service is also provided whereby an installation program can allocate, use, and free the logical parmlib without specifying the data set names.

3.1 Logical Parmlib Benefits

Parmlib members can be separated into different libraries and can have the appropriate RACF access to the various libraries. This means that an installation can separate the parmlib members into functional areas. The following options are available:

- Installations may provide a different library for the installation members. Installations can keep “SYS1.PARMLIB” as supplied by IBM. This means that when maintenance is done, the installation library is not affected.
- Testing changes to parmlib is simplified.
- Migration from one system level to another is easier in case the parmlib members′ syntax changes. With the concatenated parmlib, the members with the changed syntax could be in a different library on the other system.
- Installations can switch from one logical parmlib to another using the SETLOAD operator command.

3.2 Defining a Logical Parmlib

A PARMLIB statement is added to the LOADxx member. This statement allows specification of a logical parmlib. Up to ten data set names with optional volume serial numbers may be specified. If SYS1.PARMLIB is not specified in the LOADxx member, it is added by default as the last library.

The logical parmlib support has introduced the following functional changes in the system:

- The master scheduler uses the logical parmlib instead of the IEFPARM DD data set. This statement can be removed from the master JCL if you specify PARMLIB statements in the LOADxx member. It is ignored if you do not remove it when using the PARMLIB statements.
- A display command is provided for logical PARMLIB and IPL information. See 3.3, “PARMLIB Commands” on page 36.
- The operator command SETLOAD is provided to switch from one logical parmlib to another without an IPL. See 3.3, “PARMLIB Commands” on page 36.

Note: The logical parmlib data sets must be cataloged in the master catalog.

3.2.1 LOADxx Statements

For the complete syntax of the LOADxx member, refer to OS/390 V1R2.0 MVS Initialization and Tuning Reference.

The following statements are added to the LOADxx member definitions:

PARMLIB The parmlib statement identifies a data set that is to be included in the logical parmlib data set concatenation established during IPL and used by Master Scheduler Initialization and the logical parmlib allocation services. A maximum of ten unique PARMLIB statements can be accepted. Those in excess of the maximum will be ignored. The data sets are concatenated in the order they appear in the LOADxx member. SYS1.PARMLIB is concatenated as the last data set in the logical parmlib if it was not specified on a PARMLIB statement.

If no PARMLIB statements are found in the LOADxx member, then SYS1.PARMLIB is used.

HWNAME The optional HWNAME statement identifies a segment of LOADxx whose IODF, SYSCAT, SYSPARM, SYSPLEX, IEASYM, PARMLIB, NUCLEUS, and NUCLST statements are to be used if the specified name value matches the actual system hardware processor name.

The optional HWNAME statement specifies the name of a central processor complex (CPC), as defined in the hardware configuration definition (HCD). The HWNAME parameter is used as a filter to define parameters for a specified processor. When HWNAME is specified, it resets the LPARNAME and VMUSERID to their initial values. Refer to Figure 23 on page 33.

LPARNAME The optional LPARNAME statement identifies a segment of LOADxx whose IODF, SYSCAT, SYSPARM, SYSPLEX, IEASYM, PARMLIB, NUCLEUS, and NUCLST statements are to be used if the specified name value matches the actual name of the LPAR logical partition in which MVS/ESA is executing. When LPARNAME is specified, it resets VMUSERID to its initial value. Refer to Figure 21 on page 32.

VMUSERID The optional VMUSERID statement identifies a segment of LOADxx whose IODF, SYSCAT, SYSPARM, SYSPLEX, IEASYM, PARMLIB, NUCLEUS, and NUCLST statements are to be used if the specified name value matches the actual user ID of the VM/ESA guest machine in which MVS/ESA is executing.

Note: The HWNAME, LPARNAME, and VMUSERID statements provide a logical segmentation of the processor hardware name, the logical partition name, and the VM user ID. They are used as filters to make it possible to have a single LOADxx member define the IPL parameters for multiple images.

3.2.2 Single LOADxx Member Examples

There is no requirement to cluster the segmentation statements; that is, other LOADxx statements may occur between them. This non-clustering usage should be done with care. In the following examples, note the logical scope of the segmentation statements:

HWNAME From its occurrence to the next HWNAME statement, or to the end of file.

LPARNAME From its occurrence to the next HWNAME statement or LPARNAME statement, or to the end of file.

VMUSERID From its occurrence to the next HWNAME, LPARNAME, or VMUSERID statement, or to the end of file.

In Figure 20 there is a default segment, Segment 1, that is applicable to an MVS image being IPLed on any machine, in any logical partition, and under any VM/ESA user. In Figure 20, the second segment is applicable to an MVS image being IPLed on any machine, in any logical partition and under VM/ESA user V1. This means that user V1 being IPLed under VM/ESA uses parmlib member IEASYMV1 and not IEASYMXX as for the default segment.

```
* Default Segment
*                          Segment 1
NUCLEUS  1
NUCLST   XX
IEASYM   XX
SYSPLEX  WTSCPLX1 Y
IODF     ** SYS6     LO6RMVS1 01 Y
SYSCAT   TOTCAT1   CATALOG.TOTICFM.VTOTCAT
PARMLIB  SYSO.IPLPARM
PARMLIB  SYS1.OS390R2.PARMLIB
PARMLIB  SYS1.PARMLIB
* Segment for user V1
*                          Segment 2
VMUSERID V1
IEASYM   V1
```

Figure 20. MVS Image Running under VM/ESA

In Figure 21 on page 32, an LPARNAME segment has been added. The LOADxx statements in this segment, segment 3, apply to an MVS image being IPLed on any machine, in logical partition P4, and under any VM/ESA user. This means that the MVS image IPLed in logical partition P4, on any machine under any VM/ESA user, uses parmlib member IEASYMP4.

```
* Default Segment
*                          Segment 1
NUCLEUS  1
NUCLST   XX
IEASYM   XX
SYSPLEX  WTSCPLX1 Y
IODF     ** SYS6     LO6RMVS1 01 Y
SYSCAT   TOTCAT1   CATALOG.TOTICFM.VTOTCAT
PARMLIB  SYS0.IPLPARM
PARMLIB  SYS1.OS390R2.PARMLIB
PARMLIB  SYS1.PARMLIB
* Segment for user V1
*                          Segment 2
VMUSERID V1
IEASYM   V1
* Segment for Logical partition P4
*                          Segment 3
LPARNAME P4
IEASYM   P4
```

Figure 21. MVS Image Running in a Partition

In Figure 22, the last segment shown, Segment 4, applies to an MVS image IPLed on any machine, in logical partition P4, and under VM/ESA user MVSP4. This means that this MVS image uses parmlib member IEASYMQ4 when it is IPLed.

```
* Default Segment.
*                          Segment 1
NUCLEUS  1
NUCLST   XX
IEASYM   XX
SYSPLEX  WTSCPLX1 Y
IODF     ** SYS6     LO6RMVS1 01 Y
SYSCAT   TOTCAT1   CATALOG.TOTICFM.VTOTCAT
PARMLIB  SYS0.IPLPARM
PARMLIB  SYS1.OS390R2.PARMLIB
PARMLIB  SYS1.PARMLIB
* Segment for user V1
*                          Segment 2
VMUSERID V1
IEASYM   V1
* Segment for Logical partition P4
*                          Segment 3
LPARNAME P4
IEASYM   P4
* VMUSER MVSP4
*                          Segment 4
VMUSERID MVSP4
IEASYM   Q4
```

Figure 22. MVS Image Running in a Partition under VM/ESA

Segment 5, shown in Figure 23 on page 33, applies to an MVS being IPLed on machine ITSO942A, in any logical partition, and under any VM/ESA user. The MVS image IPLed on machine ITSO942A uses member NUCLST00 when being IPLed.

```
* Default Segment.
*                        Segment 1
NUCLEUS  1
NUCLST   XX
IEASYM   XX
SYSPLEX  WTSCPLX1 Y
IODF     ** SYS6     LO6RMVS1 01 Y
SYSCAT   TOTCAT1   CATALOG.TOTICFM.VTOTCAT
PARMLIB  SYS0.IPLPARM
PARMLIB  SYS1.OS390R2.PARMLIB
PARMLIB  SYS1.PARMLIB
* Segment for a specific machine
*                        Segment 5
HWNAME   ITSO942A
NUCLST   00
```

Figure 23. LOADxx Segment for a Specific Machine

Segment 6, shown in Figure 24, applies to an MVS image being IPLed on machine ITSO942A, in any logical partition, and under VM/ESA user P4. The MVS image IPLed on machine ITSO942A in any logical partition and under VM/ESA user P4 uses NUCLST member NUCLST01.

```
* Default Segment.
*                        Segment 1
NUCLEUS  1
NUCLST   XX
IEASYM   XX
SYSPLEX  WTSCPLX1 Y
IODF     ** SYS6     LO6RMVS1 01 Y
SYSCAT   TOTCAT1   CATALOG.TOTICFM.VTOTCAT
PARMLIB  SYS0.IPLPARM
PARMLIB  SYS1.OS390R2.PARMLIB
PARMLIB  SYS1.PARMLIB
* Segment for a specific machine
*                        Segment 5
HWNAME   ITSO942A
NUCLST   00
* Segment for a specific machine and specific VM id
*                        Segment 6
VMUSERID P4
NUCLST   01
```

Figure 24. Specific Machine and Specific VM User

Segment 7, shown in Figure 25 on page 34, applies to an MVS being IPLed on machine ITSO942A, in logical partition P4, and under VM/ESA user P4. This means that the MVS image being IPLed in machine ITSO942A, logical partition P4 and under VM/ESA user P4 uses NUCLST member NUCLST02.

```
* Default Segment.
*                          Segment 1
NUCLEUS  1
NUCLST   XX
IEASYM   XX
SYSPLEX  WTSCPLX1 Y
IODF     ** SYS6     LO6RMVS1 01 Y
SYSCAT   TOTCAT1   CATALOG.TOTICFM.VTOTCAT
PARMLIB  SYS0.IPLPARM
PARMLIB  SYS1.OS390R2.PARMLIB
PARMLIB  SYS1.PARMLIB
* Segment for a specific machine
*                          Segment 5
HWNAME   ITSO942A
NUCLST   00
* Segment for a specific machine, specific logical partition and
* specific VM/ESA user.
*                          Segment 7
LPARNAME P4
VMUSERID P4
NUCLST   02
```

Figure 25. Specific Machine, Logical Partition, and Specific User

If IODF, SYSCAT, SYSPARM, SYSPLEX, IEASYM, NUCLEUS, and NUCLST statements appear in more than one segment that is applicable to the system being IPLed, only the last one encountered is used. In Figure 26, if MVS is being IPled in logical partition P4 on machine ITSO942A, note that all three segments are applicable, but only parmlib member IEASYS11 is used.

```
*
* Use PARMLIB member IEASYS01 by default
*
SYSPARM  01
*
* If IPLing on machine ITSO942A,
*  then use PARMLIB member IEASYS10.
*
HWNAME   ITSO942A
SYSPARM  10
*
* If IPLing in logical partition P4 on machine ITSO942A,
*  then use PARMLIB member IEASYS11
*
LPARNAME P4
SYSPARM  11
. . .
*
```

Figure 26. Statements That Are Overwritten

If PARMLIB statements appear in more than one segment that is applicable to the system being IPled, all of them are used; that is, if PARMLIB statements accumulate across applicable segments, they are not overwritten. In the example shown in Figure 27 on page 35, if MVS is being IPLed in logical partition P4 on machine ITSO942A, note that SYS0.IPLPARM,

SYS1.OS390.PARMLIB, SYS1.P4.PARMLIB and SYS1.PARMLIB, in that order, comprise the logical parmlib.

```
*
* Use PARMLIB SYS0.IPLPARM as default
*
PARMLIB  SYS0.IPLPARM
*
* If IPLing on machine ITSO942A,
*  then use PARMLIB SYS1.OS390.PARMLIB.
*
HWNAME   ITSO942A
PARMLIB  SYS1.OS390.PARMLIB
*
* If IPLing in logical partition P4 on machine ITSO942A,
*  then use PARMLIB SYS1.P4.PARMLIB
*
LPARNAME P4
PARMLIB  SYS1.P4.PARMLIB
```

Figure 27. Statements That Are Cumulative

In the sample provided in Figure 28, all the MVS images IPLed using this LOADxx member have the three parmlib data sets concatenated. The MVS image IPLed on “ITSO942A” with an LPARNAME of “P4” uses a different IODF.

```
NUCLEUS  1
NUCLST   XX
IEASYM   XX
SYSPLEX  WTSCPLX1 Y
IODF     ** SYS6     LO6RMVS1 01 Y
SYSCAT   TOTCAT1   CATALOG.TOTICFM.VTOTCAT
PARMLIB  SYS0.IPLPARM
PARMLIB  SYS1.OS390R2.PARMLIB
PARMLIB  SYS1.PARMLIB
*-----------DEFINITION FOR SC48-------------*
HWNAME   ITSO942A
LPARNAME P4
IODF     ** SYS5     MVSW1    11
```

Figure 28. A Sample LOADxx Member

With the statements in LOADxx, it is possible to have one LOADxx member for all the systems in a sysplex, as shown in Figure 29 on page 36.

```
*
* Default statements for all the systems
*
NUCLEUS  1
NUCLST   XX
IEASYM   XX
SYSPLEX  WTSCPLX1 Y
IODF     ** SYS6     LO6RMVS1 01 Y
SYSCAT   TOTCAT1   CATALOG.TOTICFM.VTOTCAT
PARMLIB  SYS0.IPLPARM
PARMLIB  SYS1.OS390R2.PARMLIB
PARMLIB  SYS1.PARMLIB
*
* Statements that are applicable for the MVS being IPLed on
*  logical partition P4 and in machine ITSO942A.
*
HWNAME   ITSO942A
LPARNAME P4
IODF     ** SYS5     MVSW1    11
*
* Statements that are applicable for the MVS being IPLed on
*  logical partition T5 and in machine ITSO942A.
*
LPARNAME T5
IEASYM   R2
. . .
*
```

Figure 29. One LOADxx for the Sysplex

3.3 PARMLIB Commands

New commands have been added to enable installations to display the current logical parmlib, display general IPL information, and also to be able to change the current logical parmlib settings.

The commands are the following:

- DISPLAY PARMLIB

  This command displays the logical parmlib setup for the IPLed system. The output of this command includes the parmlib data set name(s) and volser(s) that was defined by LOADxx PARMLIB statement(s), and if used, MASTER JCL IEFPARM DD statements. When the errors option on the command is used, the display shows any parmlibs that were defined in LOADxx but were not found. This command is only valid before a SETLOAD command is issued. A sample output is shown in Figure 30 on page 37.

```
D PARMLIB
 IEE251I 09.32.31 PARMLIB DISPLAY 452
  PARMLIB DATA SETS SPECIFIED
  AT IPL
  ENTRY  FLAGS  VOLUME  DATA SET
    1      D    TOTSY1  SYS1.PARMLIB
  MASTER PROCESSING USING THE FOLLOWING PARMLIBS
  ENTRY  FLAGS  VOLUME  DATA SET
    1      S    TOTSY1  SYS1.PARMLIB
```

Figure 30. DISPLAY PARMLIB Command Issued before Changing Parmlibs

- DISPLAY IPLINFO

  This command displays the general IPL information used by the system. The output includes the date and time of the IPL, release level, LOADxx information, and what IEASYSxx and IEASYMxx parmlib members were used. Shown in Figure 31 is sample output of the command.

```
D IPLINFO
IEE254I  13.06.10 IPLINFO DISPLAY 025
 SYSTEM IPLED AT 08.25.41 ON 08/05/1996
 RELEASE SP6.0.2
 USED LOADR2 IN SYS0.IPLPARM ON 0CD0
 IEASYM LIST = XX
 IEASYS LIST = (R2,XX) (OP)
```

Figure 31. Display IPLINFO Command

- SETLOAD xx,PARMLIB

  This command allows the installation to dynamically change a parmlib concatenation without having to IPL. A sample output of the SETLOAD command is shown in Figure 32.

```
SETLOAD R2,PARMLIB,DSN=SYS0.IPLPARM
IEF196I IEF237I 0CD0 ALLOCATED TO SYS00006
IEE252I MEMBER   LOADR2 FOUND IN SYS0.IPLPARM
IEF196I IEF237I 0CD0 ALLOCATED TO SYS00007
IEF196I IEF237I 0FC1 ALLOCATED TO SYS00008
IEF196I IEF237I 0FC1 ALLOCATED TO SYS00009
IEF196I IEF285I   SYS1.PARMLIB                                KEPT
IEF196I IEF285I   VOL SER NOS= TOTSY1.
IEF196I IEF285I   SYS0.IPLPARM                                KEPT
IEF196I IEF285I   VOL SER NOS= IODFPK.
IEF107I PARMLIB CONCATENATION WAS UPDATED FROM LOADR2
```

Figure 32. Sample SETLOAD Command Output

The sample output shown in Figure 33 on page 38 was issued after the SETLOAD command was issued.

```
D PARMLIB
IEE251I 09.36.52 PARMLIB DISPLAY 470
 PARMLIB DATA SETS SPECIFIED
 AT 09.34.11 ON 08/03/1996
 LOADR2 DATA SET=SYS0.IPLPARM
        VOLUME=CATALOG
 ENTRY  FLAGS  VOLUME  DATA SET
   1      S    CATALOG SYS0.IPLPARM
   2      S    CATALOG SYS1.OS390R2.PARMLIB
   3      S    CATALOG SYS1.PARMLIB
```

Figure 33. DISPLAY PARMLIB Command after SETLOAD Has Been Issued

Figure 34 shows the output of a DISPLAY PARMLIB command after an IPL had taken place using LOADR2. PARMLIB statements were specified in LOADR2. Note the difference in the FLAGS column between Figure 33 and Figure 34.

The FLAGS describe how the parmlibs were specified:

- S denotes the LOADxx PARMLIB statement.
- D denotes the default (SYS1.PARMLIB).

```
D PARMLIB
IEE251I 13.10.57 PARMLIB DISPLAY 027
 PARMLIB DATA SETS SPECIFIED
 AT IPL
 ENTRY  FLAGS  VOLUME  DATA SET
   1      S    IODFPK  SYS0.IPLPARM
   2      S    TOTSY1  SYS1.OS390R2.PARMLIB
   3      D    TOTSY1  SYS1.PARMLIB
```

Figure 34. Display Parmlib Command after IPL

3.4 User Interface to Parmlib

To provide logical parmlib support, a service (IEFPRMLB) is provided to allow programs to access logical parmlib in a straight forward manner. The IEFPRMLB macro provides a straight forward way for programs to access the logical parmlib. Four basic functions are available.

- IEFPRMLB REQUEST=ALLOCATE,DDNAME=ddname....

  This allows a program to allocate the logical parmlib data set concatenation without the knowledge of the specific data sets in the concatenation. This function optionally provides the ability to read a specified member of the logical parmlib and have the member's contents returned in an input buffer.

- IEFPRMLB REQUEST=FREE,DDNAME=ddname...

  This allows a program to unallocate the logical parmlib using a DDname.

- IEFPRMLB REQUEST=READMEMBER,DDNAME=ddname, MEMNAME=MEMBER,READBUF=buffer,...

  This allows a program to read a specified member of an already allocated logical parmlib data set concatenation and have the member's contents returned in an input buffer.

- IEFPRMLB REQUEST=LIST,BUFFER=buffer....

  This allows a program to retrieve information about the logical parmlib data set concatenation.

A control block, the IPA (Initialization Parameter Area), is constructed during IPL and contains the following information:

- The TOD value at Master Scheduler Initialization.
- The load parameter used by IPL.
- The system names (hardware processor name, logical partition name, and VM/ESA user ID).
- The name and device number of the data set in which the LOADxx member used by IPL was found.
- All LOADxx information.
- The IEASYSxx parameter values with all symbolics resolved and with any additional operator supplied values merged in. Default values are supplied for those IEASYSxx parameters that have default values and were not specified.

The IPA is mapped using the IHAIPA macro. The method of accessing the control block is shown in Figure 35.

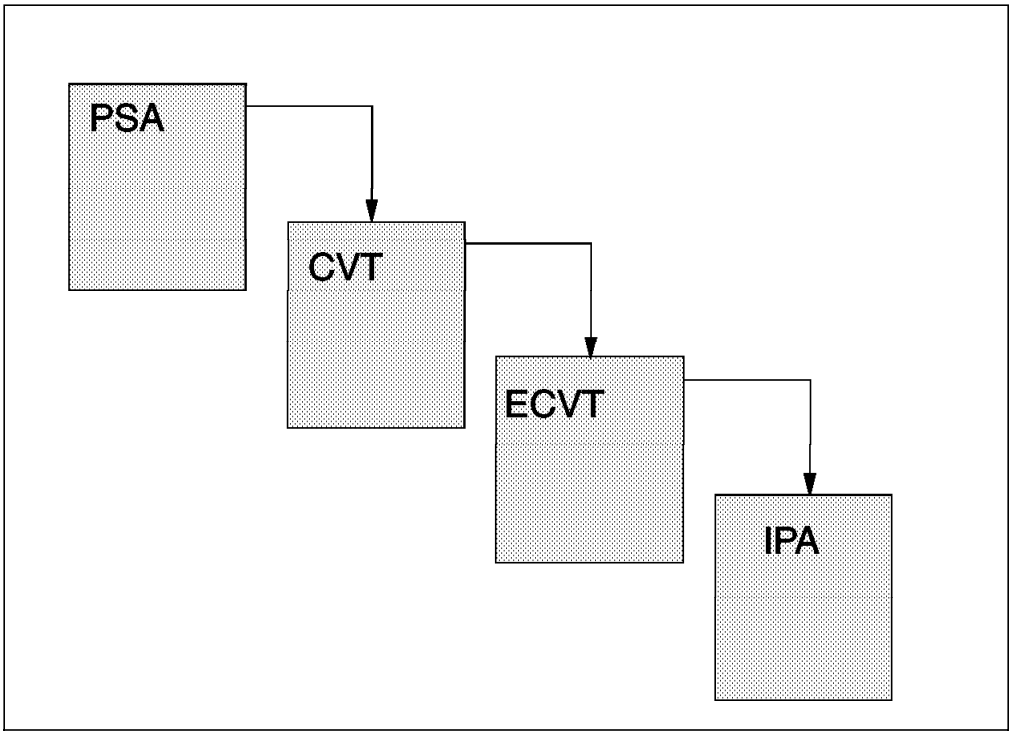


Figure 35. Initialization Parameter Area

3.5 Master Scheduler JCL Changes

The IEFPARM DD name statement used in master JCL processing will no longer be used if one specifies the PARMLIB statement in LOADxx. The logical parmlib setup is recommended to be used by all processes, including master scheduling processing.

The following rules apply for a logical parmlib and the master JCL:

- If a logical parmlib is established, and the master JCL specifies an IEFPARM DD statement:

  - Ignore the IEFPARM DD statement. Master processing uses the logical parmlib setup.

  - Issue message IEE253I.

- If there is no logical parmlib established, then the master JCL IEFPARM DD statement is used for master processing.
- Samplib member IEESMJCL is no longer included in the IEFPARM DD card in the JCL.

The master scheduler parmlib processing is shown in Figure 36, and the flowchart is shown in Figure 37 on page 41.

MASTER JCL IEFPARM	LOADxx's Parmlibs specified and found	LOADxx's Parmlibs specified but not found	LOADxx's Parmlibs not specified
DD Statement specified	IEFPARM is ignored with a warning message. Use LOADxx's Parmlibs.	IEFPARM is ignored with a warning message. Use default SYS1.PARMLIB.	Use IEFPARM's Parmlibs.
DD statement not specified	Use LOADxx's Parmlibs.	Use default SYS1.PARMLIB.	Use default SYS1.PARMLIB.

Figure 36. Master Scheduler Decision Table

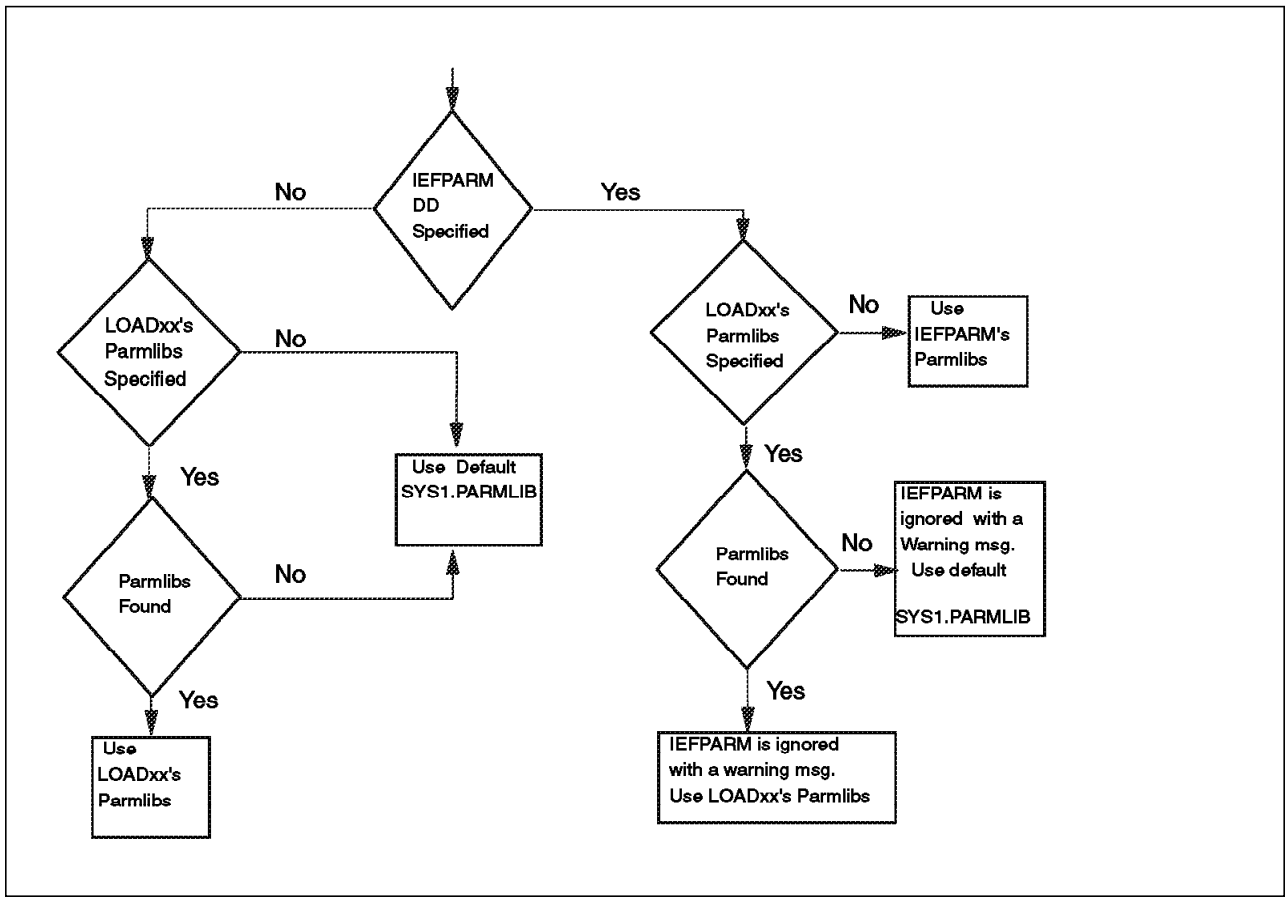


Figure 37. Master Scheduler Decision Flow Chart

3.6 Changes to CSVLLAxx

The CSVLLAxx member has a parameter added:

- PARMSUFFIX(xx)

The PARMSUFFIX statement allows an installation to specify an additional CSVLLAxx member to be processed. The system processes this member completely when encountering this statement.

This statement is very similar to the PARMLIB(dsn) SUFFIX(xx) statement. The difference is that instead of having to specify a data set name, PARMSUFFIX searches the logical parmlib for the CSVLLAxx member.

CSVLLAxx Recommendations

- Examine your installation CSVLLAxx members. If they currently contain PARMLIB (dsn) SUFFIX (xx) parameters, determine if the (dsn) specified is a data set in the parmlib concatenation. If so, consider replacing the PARMLIB (dsn) SUFFIX (xx) parameters with the PARMSUFFIX (xx) parameter, so you will not have to change the PARMLIB (dsn) SUFFIX (xx) parameters if you move the member to another data set within the parmlib concatenation.

3.7 Implementation of Logical Parmlib Support

To implement the logical parmlib support, do the following:

- First install OS/390 Release 2 on the current systems.
- Create a matrix of parmlib members.

  The reason for creating a matrix of parmlib members is so that you can easily see which members are used by which systems in the sysplex. From the matrix, you can also note whether an installation can share the LOADxx member.
- Decide which members may reside in different libraries.

  Members should reside in libraries that correspond to specific functional areas of the installation.
- Create the different libraries.
- Create a different LOADxx member in SYSx.IPLPARM.
- Concatenate the parmlib libraries in the correct sequence.

  The concatenation of the libraries is important because the normal search order through concatenated libraries also holds for logical parmlibs, if an installation has decided on the following libraries as shown in Figure 38.

```
SYS1.CICS.PARMLIB    to contain parmlib members for CICS
SYS1.IMS.PARMLIB     to contain parmlib members for IMS
SYS1.IPCS.PARMLIB    to contain parmlib members for IPCS
SYS1.SYSPROG.PARMLIB to contain all the IEASYSxx related members
SYS1.PARMLIB         installation default library always last.
```

Figure 38. Sample Concatenation

  If the CICS library is ahead of the SYSPROG library in LOADxx, the installation is using IEASYS04, and a different IEASYS04 is created in the CICS library, that member in the CICS library is used at IPL.
- Copy the members from SYS1.PARMLIB into the appropriate library.
- Ensure that the correct security profiles are established.
- Change the JCL for the appropriate concatenation sequence.

  As shown in 3.7.1, “Logical Parmlib Support” on page 43, certain products support logical parmlib. Remove the specified DD card from the products that do support logical parmlib. Ensure that the products that do not have logical parmlib support have the correct concatenation in the JCL.
- Issue SETLOAD to implement the logical parmlib.
- Start the STCs that have been changed.
- Remember to change the IPL parameters to show the correct LOADxx member.

  Changing the LOAD parameter ensures that the next IPL automatically brings in the logical parmlib support.

3.7.1 Logical Parmlib Support

The products that support logical parmlib are shown in Table 2.

Table 2. Products That Support Logical Parmlib

Product	**APAR**
OAM	OW18641
RMM	OW19803
HSM	OW19925
TCAM	OW20298
SDSF	PN83037
RMF	OS/390 Release 2
OE	OS/390 Release 2

The following products can use the logical parmlib if an installation so chooses:

- DLF, VLF, LLA, and IPCS

  JCL changes must occur for DLF, VLF LLA and IPCS products to use logical parmlib.

The following listed products use the logical parmlib by default:

- SMF, CTRACE, MMS, and APPC/ASCH

Note: When using logical parmlib support, ensure that any product not supporting the logical parmlib support has the correct concatenation of input parmlib members when trying to read the parmlib members.

Chapter 4. LNKLST and LPALST Enhancements

Prior to OS/390 Release 2, SYS1.LINKLIB, SYS1.MIGLIB, and SYS1.CSSLIB were concatenated ahead of the LNKLSTxx libraries specified in the LNKLSTxx member. Also prior to OS/390 Release 2, SYS1.LPALIB was concatenated ahead of the libraries specified in the LPALSTxx member in parmlib.

The following enhancements have been made in OS/390 Release 2 for the LNKLST and LPALST:

- An installation may place user-defined libraries ahead of LINKLIB, MIGLIB, and CSSLIB in the LNKLST concatenation.
- An installation may place a user-defined library ahead of LPALIB in the LPALST concatenation.
- Another enhancement to LNKLST is that the LNKLST libraries may now be defined in PROGxx.
- You may have more than 123 extents in the LNKLST and LPALST concatenation and up to 255 extents.
- Commands to display the LNKLST and display diagnostic information about exits have been added.

4.1 SYSLIB Statement

If an installation wants to specify alternates for the following listed data sets, then the SYSLIB statement in PROGxx is used to do this.

- SYS1.LINKLIB
- SYS1.MIGLIB
- SYS1.CSSLIB
- SYS1.LPALIB

The SYSLIB statement in PROGXX allows an installation to place data sets other than SYS1.LINKLIB, SYS1.MIGLIB, and SYS1.CSSLIB first in the LNKLST concatenation, and a data set other than SYS1.LPALIB first in the LPALST concatenation. However, the installation must ensure that SYS1.LINKLIB, SYS1.MIGLIB, and SYS1.CSSLIB are defined to the LNKLST concatenation and that SYS1.LPALIB is defined to the LPALST concatenation that you define for the system.

You may use SYSLIB statements in PROGxx when you want to set up a test environment to apply maintenance or other code fixes to the system code. Use the LINKLIB, MIGLIB, and CSSLIB options of the SYSLIB statements to place the alternate data sets defined at the beginning of the LNKLST concatenation. Use the LPALIB option of the SYSLIB statement to place the alternate data set defined at the beginning of the LPALST concatenation. Using these alternate system data sets, the installation can test the fixes without having to modify the system defaults. Once testing of the code fixes is complete, a re-IPL can occur, with the regular system libraries appearing first in the LNKLST and LPALST concatenations. The installation can specify SYSLIB statements in PROGxx and use either PROGxx with LNKLST statements or LNKLSTxx to define and activate the LNKLST concatenation.

Figure 39 shows what occurs when the SYSLIB statement is specified in PROGxx.

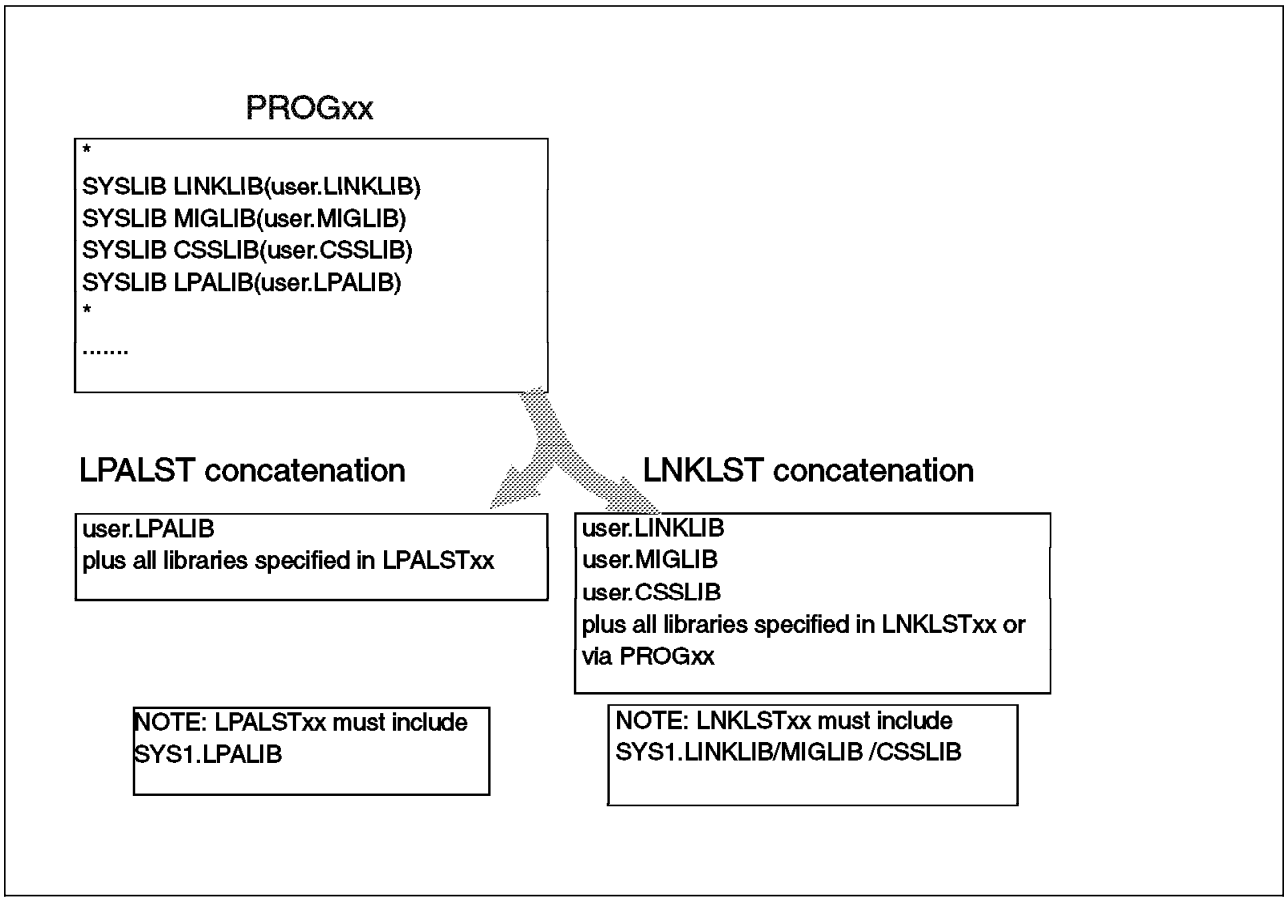


Figure 39. SYSLIB Statement

4.2 Syntax of the SYSLIB Statement

The SYSLIB statement has the following syntax:

- SYSLIB LINKLIB(data_set_name)
- SYSLIB MIGLIB(data_set_name)
- SYSLIB CSSLIB(data_set_name)
- SYSLIB LPALIB(data_set_name)

4.3 Statements/Parameters for the SYSLIB Statement

- SYSLIB

  Statement type indicating that an alternate data set is to be defined for SYS1.LINKLIB, SYS1.MIGLIB, and SYS1.CSSLIB in the LNKLST concatenation, and for SYS1.LPALIB in the LPALST concatenation.

- LINKLIB(name)

  Specifies the name of the LINKLIB data set. If a different library other than SYS1.LINKLIB is specified, you must ensure that SYS1.LINKLIB occurs within

the LNKLST concatenation. The system places the LINKLIB data set first in the LNKLST concatenation.

Default value: If you do not specify LINKLIB, the system uses SYS1.LINKLIB as the LINKLIB data set.

- MIGLIB(name)

  Specifies the name of the MIGLIB data set. If you specify a library other than SYS1.MIGLIB, you must ensure that SYS1.MIGLIB occurs within the LNKLST concatenation. The system places the MIGLIB data set after the LINKLIB data set in the LNKLST concatenation.

  Default value: If you do not specify MIGLIB, the system uses SYS1.MIGLIB as the MIGLIB data set.

- CSSLIB(name)

  Specifies the name of the CSSLIB data set. If you specify a library other than SYS1.CSSLIB, you must ensure that SYS1.CSSLIB occurs within the LNKLST d concatenation. The system places the CSSLIB data set after the MIGLIB data set in the LNKLST concatenation.

  Default value: If you do not specify CSSLIB, the system uses SYS1.CSSLIB as the CSSLIB data set.

- LPALIB(name)

  Specifies the name of the LPALIB data set. If you specify a library other than SYS1.LPALIB, you must ensure that SYS1.LPALIB occurs within the LPALST concatenation. The system places the LPALIB data set first in the LPALST concatenation.

  Default value: If you do not specify LPALIB, the system uses SYS1.LPALIB first in the LPALST concatenation.

4.4 Example of the SYSLIB Statement

The following example shows a PROGxx parmlib member to be IPLed in a test environment that is applying code fixes for the system. The libraries specified on these SYSLIB statements contain no data set members. Whether you use PROGxx or LNKLSTxx to activate the LNKLST concatenation, the system places SYS2.LINKLIB, SYS2.MIGLIB, and SYS2.CSSLIB at the start of the LNKLST concatenation. The system places SYS2.LPALIB at the beginning of the LPALST concatenation. (You must ensure that SYS1.LINKLIB, SYS1.MIGLIB, and SYS1.CSSLIB appear somewhere in the LNKLST concatenation and SYS1.LPALIB appears somewhere in the LPALST concatenation.)

```
SYSLIB  LINKLIB(SYS2.LINKLIB)
SYSLIB  MIGLIB(SYS2.MIGLIB)
SYSLIB  CSSLIB(SYS2.CSSLIB)
SYSLIB  LPALIB(SYS2.LPALIB)
```

Figure 40. Sample SYSLIB Statements in PROGxx

4.5 LNKLST Set Description

A LNKLST set consists of an ordered list of data sets for processing as the LNKLST concatenation. Unless overridden by SYSLIB statements, every LNKLST set contains the following libraries as the first data sets in the concatenation:

- SYS1.LINKLIB
- SYS1.MIGLIB
- SYS1.CSSLIB

The system automatically adds these data sets to the beginning of the LNKLST set that you define. If these data sets are not available to the system at IPL, a wait state occurs.

In Figure 41, two sets are shown:

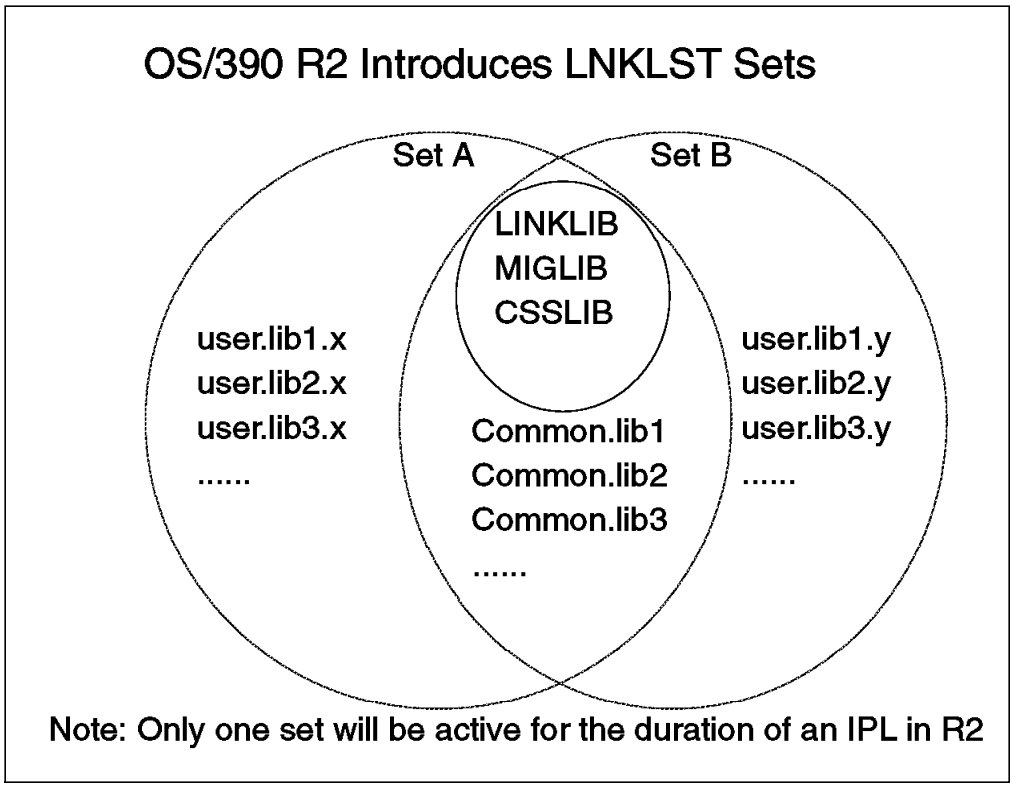


Figure 41. Description of Sets

The default set consists of the following libraries:

- LINKLIB
- MIGLIB
- CSSLIB

This is the case if no LNKLSTxx member was specified and no LNKLST statements were specified in PROGxx.

SET A consists of the following libraries:

- LINKLIB
- MIGLIB
- CSSLIB

- Common.lib1
- Common.lib2
- Common.lib3
- user.lib1.x
- user.lib2.x
- user.lib3.x

SET B consists of the following libraries:

- LINKLIB
- MIGLIB
- CSSLIB
- Common.lib1
- Common.lib2
- Common.lib3
- user.lib1.y
- user.lib2.y
- user.lib3.y

4.6 Using the LNKLST Statement

To be able to use the correct SET at IPL time, a LNKLST statement has been added to PROGxx. The LNKLST statement is used:

- To define the LNKLST set
- To add a data set to the LNKLST set
- To indicate that a LNKLST set is to be activated at IPL

You can add a data set to any LNKLST set that you define and specify the position of the data set in the list. You cannot add the data set before any of the system default data sets in the concatenation; that is, you can only concatenate the data set after SYS1.CSSLIB in the LNKLST set. To read how to replace the system default data sets with alternate data sets at the beginning of the LNKLST concatenation, see 4.1, “SYSLIB Statement” on page 45.

4.7 Using PROGxx Instead of LNKLSTxx

You can use LNKLST statements in PROGxx instead of LNKLSTxx to define the LNKLST concatenation. At IPL, ensure that you have a LNKLST ACTIVATE statement for the LNKLST set that you have defined, and specify PROG=xx in IEASYSxx instead of LNK=xx. To convert a LNKLSTxx member to PROGxx format, refer to 4.12, “Converting from LNKLSTxx to Using LNKLST in PROGxx” on page 54.

The LNKLST statement in PROGxx has three formats.

- LNKLST DEFINE
- LNKLST ADD
- LNKLST ACTIVATE

For the complete syntax, refer to *OS/390 V1R2.0 MVS Initialization and Tuning Guide*.

The parameters for the LNKLST statement are:

LNKLST Statement type indicating that an action is to be performed for a LNKLST set.

DEFINE Specifies the definition of a LNKLST set (a set of ordered data sets for the LNKLST concatenation).

Default value: None

ADD Indicates that you want to add a data set to the specified LNKLST set.

Default value: None

ACTIVATE Indicates that you want to activate the specified LNKLST set as the LNKLST concatenation. If a data set in the LNKLST set has been migrated before the LNKLST set is activated, the request waits until the data set is available.

Default value: None

NAME(name) The name of the LNKLST set that you want to specify. The naming conventions are as follows:

- You can specify from 1 to 16 characters for name.
- You can use alphanumerics, underscores, periods, and #, $, or @.
- Do not use imbedded blanks.
- Do not use the names CURRENT or IPL. The system uses CURRENT to mean the current LNKLST set and IPL to mean LNKLST information specified in SYS1.PARMLIB member LNKLSTxx.
- Do not begin the name with SYS because SYS is reserved for IBM use.

  Default value: None

In Figure 42, an example of a NAME parameter is shown.

```
NAME(MY.LNKLST.SET)
```

Figure 42. Specification of NAME Parameter on LNKLST Statement

DSNAME(dsname) The 44-character name of a data set or library that you want to add to the specified LNKLST set. DSN, LIB, and LIBRARY are accepted synonyms for this parameter.

The data set can be a PDS or a PDSE. (IBM recommends that you use PDSEs because of the limitations on the number of extents for a LNKLST concatenation.)

VOLUME=(name) The name of the volume on which the data set resides. The data set must be cataloged. If the volume does not match the name on the catalog, the ADD request fails. The name can be from 1 to 6 characters.

When the data set is cataloged in a user catalog, instead of the master catalog, you can use this parameter. If a data set is cataloged in a user catalog, but not in the system master catalog, you must specify the VOLSER of the volume on which the data set resides.

Default value: If you omit this parameter, the system uses the volume indicated on the catalog.

ATBOTTOM ATBOTTOM|ATTOP|AFTER(dsname) indicates where in the LNKLST set you want to place the data set.

ATBOTTOM indicates that you want to place the data set specified on the DSNAME parameter at the bottom of the list of data sets in the LNKLST set.

ATTOP ATTOP indicates that you want the data set specified on the DSNAME parameter to be added to the beginning of the LNKLST set. The system always places the LINKLIB, MIGLIB, and CSSLIB data sets (in that order) at the beginning of every LNKLST set in the LNKLST concatenation. If you use ATTOP, the system always places the data set after the CSSLIB data set.

AFTER(dsname) AFTER(dsname) indicates that the system places the data set specified on the DSNAME parameter after the data set specified by dsname. You cannot use this parameter to place a data set after the LINKLIB, MIGLIB, or CSSLIB data set in the LNKLST set. Instead, use ATTOP if you want to place the data set immediately after the CSSLIB data set.

Default value: If you omit ATBOTTOM, ATTOP, or AFTER, the system adds the data set to the bottom of the LNKLST set.

4.8 Examples of the LNKLST Statement

The example in Figure 43 shows how to define LNKLST set SET.A and how to indicate that SET.A is to be activated at IPL. The resulting LNKLST concatenation consists of the LINKLIB, MIGLIB, and CSSLIB data sets, in that order.

```
. . .
LNKLST DEFINE NAME(SET.A)
LNKLST ACTIVATE NAME(SET.A)
. . .
```

Figure 43. Sample PROGxx Member with LNKLST Statements

The sample PROGxx listed in Figure 44 on page 52 shows how to add data sets to the LNKLST set SET.B:

```
. . .
/* Define the LNKLST set SET.B.
LNKLST DEFINE NAME(SET.B)
/*
/* Place SDSF.SISFLOAD at the beginning of SET.B.
/* The system automatically adds data sets for LINKLIB, MIGLIB, and
/* CSSLIB to the beginning of SET.B before SDSF.SISFLOAD.
/*
LNKLST ADD NAME(SET.B) DSNAME(SDSF.SISFLOAD) ATTOP
/*
/* Place SYS1.DGTLIB at the end of SET.B.
/*
LNKLST ADD NAME(SET.B) DSNAME(SYS1.DGTLIB) ATBOTTOM
/*
/* Place SYS1.CMDLIB after SDSF.SISFLOAD in SET.B.
LNKLST ADD NAME(SET.B) DSNAME(SYS1.CMDLIB) AFTER(SDSF.SISFLOAD)
/*
/* Activate SET.B
LNKLST ACTIVATE NAME(SET.B)
```

Figure 44. Sample PROGxx Member with LNKLST Statements

The sample PROGxx member in Figure 45 shows how the concatenation of data sets for LNKLST1 set is defined.

```
. . .
SYSLIB LINKLIB(SYS2.LINKLIB)   /* define alternate LINKLIB */
SYSLIB MIGLIB(SYS2.MIGLIB)     /* define alternate MIGLIB  */
SYSLIB CSSLIB(SYS2.CSSLIB)     /* define alternate CSSLIB  */
SYSLIB LPALIB(SYS2.LPALIB)     /* define alternate LPALIB  */
LNKLST DEFINE NAME(LNKLST1)
LNKLST ADD NAME(LNKLST1) DSNAME(SYS1.LINKLIB) ATTOP
LNKLST ADD NAME(LNKLST1) DSNAME(SYS1.MIGLIB)
LNKLST ADD NAME(LNKLST1) DSNAME(SYS1.CSSLIB)
LNKLST ADD NAME(LNKLST1) DSNAME(SYS1.AUXLIB) VOLUME(U32PAK)
LNKLST ACTIVATE NAME(LNKLST1)
. . .
```

Figure 45. Sample PROGxx Member with LNKLST and SYSLIB Parameters

As a result of the PROGxx specifications in Figure 45, the following data sets, in the order specified, are concatenated at IPL:

- SYS2.LINKLIB,SYS2.MIGLIB,SYS2.CSSLIB,SYS1.LINKLIB, SYS1.MIGLIB,SYS1.CSSLIB,SYS1.AUXLIB

In the example of the LNKLST1 concatenation, note the following:

- The SYSLIB statements specify that SYS2.LINKLIB, SYS2.MIGLIB, and SYS2.CSSLIB replace the system defaults at the beginning of the LNKLST concatenation.
- SYS2.LPALIB is to appear first in the LPALST concatenation.
- SYS1.LPALIB must be defined in the LPALSTxx member.
- SYS1.LINKLIB, SYS1.MIGLIB, and SYS1.CSSLIB must be defined in the LNKLST concatenation.

- SYS1.LINKLIB, SYS1.MIGLIB, and SYS1.CSSLIB are specified, in that order, after SYS2.LINKLIB, SYS2.MIGLIB, and SYS2.CSSLIB.
- SYS1.AUXLIB is specified at the end of the LNKLST1. VOLUME indicates that SYS1.AUXLIB is cataloged on VOLSER U32PAK.
- The LNKLST ACTIVATE statement activates LNKLST1 at IPL.

4.9 Concatenating Data Sets to the LNKLST Concatenation

The number of data sets you can concatenate to form the LNKLST concatenation is limited by the total number of DASD extents the data sets will occupy. The total number of extents must not exceed 255. You must have DFSMS/MVS 1.3 or later installed to support this limit of 255 extents for the LNKLST concatenation. A partitioned data set extended (PDSE) counts as one extent.

The system concatenates as many of the data sets as possible until the limit of 255 extents is reached. The system ignores the remaining data sets. When the limit has been exceeded, the system writes error message IEA328E to the operator's console. This message is issued whether the concatenation is defined by LNKLSTxx or by PROGxx.

After the system has been IPLed, you cannot add or delete data sets from the LNKLST concatenation.

4.10 Placement of SYSLIB and LNKLST Statements in PROGxx

You can place LNKLST statements for a LNKLST set in different PROGxx members. For example, you can specify PROG=(01,02,03) and place the LNKLST DEFINE statement in PROG01, LNKLST ADD statements in PROG02, and the LNKLST ACTIVATE statement in PROG03.

SYSLIB statements must always appear before any LNKLST statements in PROGxx. If you specify multiple PROG=xx members, define any SYSLIB statements ahead of LNKLST statements. For example, if you specify PROG=(01,02) during IPL, consider the following:

- If PROG01 has a LNKLST statement, ensure that no SYSLIB statement appears after the LNKLST statement, or in PROG02.
- If only PROG02 has a LNKLST statement, ensure that no SYSLIB statement appears after the LNKLST statement in PROG02.

4.11 Usage of LNKLST Statements in PROGxx

To be able to use the LNKLST statements in PROGxx, the PROGxx parmlib member must be used. This is done by using the PROG= parameter in IEASYSxx, or it may be specified by the operator.

The following rules apply to specifying PROG= and LNK=.

- You can specify PROG=xx instead of LNK=xx for the LNKLST concatenation. Whether you use PROGxx or LNKLSTxx to define the LNKLST concatenation, the system always places the LINKLIB, MIGLIB, and CSSLIB data sets (either the system defaults or the data sets specified on SYSLIB statements) first in the concatenation.

- If you use PROGxx and do not use LNKLST statements, the system uses LNKLSTxx, if specified on LNK=xx to define the LNKLST concatenation.
- If you define a LNKLST set to be activated through PROGxx and specify both PROG=xx and LNK=xx, the system uses the definitions in PROGxx and issues the following message:

  CSV487I LNK IPL PARAMETER HAS BEEN IGNORED. LNKLST SET lnklstname IS BEING USED

4.12 Converting from LNKLSTxx to Using LNKLST in PROGxx

If you want to define the LNKLST concatenation through PROGxx, convert the definitions in LNKLSTxx to equivalent definitions in PROGxx.

You can use the CSVLNKPR REXX EXEC to convert the syntax of LNKLSTxx to a PROGxx format. The CSVLNKPR EXEC resides in SYS1.SAMPLIB and requires TSO/E V2 and ISPF V2.3.

To install CSVLNKPR, place the EXEC in a data set that is accessible to ISPF and PDF edit macros. If you specify EXECUTIL SEARCHDD(YES) command, or if you have modified the TSO/E installation parameters to search SYSEXEC automatically, place CSVLNKPR in a data set allocated to SYSEXEC. Otherwise, place CSVLNKPR in a data set allocated to SYSPROC.

To invoke CSVLNKPR:

- Make sure that the LNKSTxx member to be converted is valid and syntactically correct.
- Edit a PROGxx parmlib member.

  Make sure it is a new PROGxx member. If it is not, then the CSVLNKPR EXEC puts comment tags around the existing statements.
- Copy the LNKLSTxx member into the PROGxx member.
- Enter CSVLNKPR on the edit command line. The system places the modified member in the edit buffer.

  If you enter CSVLNKPR without a name, the name of the LNKLST set in PROGxx defaults to LNKLSTXX. If you specify CSVLNKPR xxxx, the system uses xxxxx as the name of the LNKLST set in PROGxx.
- Copy the current PROGxx member into the new PROGxx member.
- Save the new PROGxx member.

After you convert to PROGxx, remove the LNK=xx system parameter from IEASYSxx and from IEASYS00, and then activate PROGxx (specify the PROG=xx system parameter at the next IPL). You can place PROG=xx in IEASYSxx and IEASYS00.

4.13 Display Commands for LNKLST

A command to display information about the LNKLST concatenation has been added. This command displays the LNKLST whether the LNKLST was created with LNKLSTxx or with the LNKLST statements in PROGxx. The syntax of the command is as follows:

```
D PROG,LNKLST
```

Figure 46. Command to Sisplay LNKLST Concatenation

The output of the command is shown in Figure 47. For a complete description of the output fields, refer to message CSV470I in OS/390 V1R2.0 MVS System Messages, Vol 2 (ASB-ERB).

```
D PROG,LNKLST
CSV470I 12.46.46 LNKLST DISPLAY 516
LNKLST SET IPL   LNKAUTH=APFTAB
ENTRY  APF  VOLUME  DSNAME
   1    A   OS3R2T  SYS1.LINKLIB
   2    A   OS3R2T  SYS1.MIGLIB
   3    A   OS3R2T  SYS1.CSSLIB
   4    A   TOTSY1  SYS2.LINKLIB
   5    A   OS3R2T  IHV.V1R3M0.SIHVMOD1
   6    A   OS3R2T  SDSF.SISFLOAD
   7        OS3R2T  ISP.V4R2M0.SISPLOAD
   8        OS3R2T  ISP.V4R2M0.SISPSASC
   9    A   OS3R2T  SYS1.CMDLIB
  10    A   OS3R2T  SYS1.CEE.V1R6M0.SCEERUN
  11        OS3R2T  CBC.V3R2M0.SCBC3CMP
  . . .
```

Figure 47. D PROG,LNKLST Output before Conversion

The output of the same command is shown in Figure 48. Note that the LNKLST set has now changed. The output of the command shown in Figure 47 was done before converting the LNKLST to using the LNKLST statements in PROGxx. The LNKLST set of IPL denotes the set that exists in LNKLSTxx.

```
D PROG,LNKLST
CSV470I 11.56.24 LNKLST DISPLAY 918
LNKLST SET LNKLSTR2   LNKAUTH=APFTAB
ENTRY  APF  VOLUME  DSNAME
   1        TOTSY1  SYS1.SYSPROG.LINKLIB
   2        TOTSY1  SYS1.SYSPROG.MIGLIB
   3        TOTSY1  SYS1.SYSPROG.CSSLIB
   4    A   OS3R2T  SYS1.LINKLIB
   5    A   OS3R2T  SYS1.MIGLIB
   6    A   OS3R2T  SYS1.CSSLIB
   7    A   TOTSY1  SYS2.LINKLIB
   8    A   OS3R2T  IHV.V1R3M0.SIHVMOD1
   9    A   OS3R2T  SDSF.SISFLOAD
  10        OS3R2T  ISP.V4R2M0.SISPLOAD
. . .
```

Figure 48. D PROG,LNKLST Output after Conversion

The LNKLST set is different in Figure 48 from the LNKLSt set shown previously. In Figure 48, the concatenation of the installation libraries for LINKLIB, MIGLIB, and CSSLIB are also shown.

4.14 Display Commands for Exits

This command is used to display diagnostic information about an exit. The information displayed shows the entry point, load point, and length associated with each exit routine. The information may be incomplete (shown as zeros) when the exit routine was located in LPA. In the case of the loadpt and length being zeros, only the entry point address has useful information. When the exit is located using LNKLST, then the information for loadpt and length is complete. No information is shown for an inactive routine, as such routines have not been loaded into storage.

Shown in Figure 49 is the command and the output of the command. Refer to *OS/390 V1R2.0 MVS System Commands* for the complete syntax.

```
D PROG,EXIT,EN=SYS.IEFUJI,DIAG
CSV464I 13.42.29 PROG,EXIT DISPLAY 578
EXIT SYS.IEFUJI
MODULE    STATE EPADDR    LOADPT    LENGTH    JOBNAME
IEFUJI      A   83EA3598  00000000  00000000  *
```

Figure 49. D PROG,EXIT,EN=...,DIAG Command and Output

Chapter 5. Symbolic Pre-Processor

Prior to this offering, when adding or changing symbols to IEASYMxx, the installation had to IPL to test the changes. This was extremely unproductive as often the symbols were not correct and only discovered at an IPL.

The solution to this problem is the availability of a parmlib symbolic pre-processor tool.

The parmlib symbolic pre-processor is an ISPF-based interactive dialog. This tool can show you what your system symbolics and parmlib members would be, given a hypothetical hardware and software configuration. The tool notifies you if data sets or members cannot be found. The tool also detects certain syntax errors and displays the results in the parmlib member(s) that you select. Other functions of the tool allow you to manipulate the symbol table and perform other useful parmlib functions.

5.1 Installation of Symbolic Pre-Processor Tool

The installation program SPPINST unpacks the specified data set. It follows special commands embedded in the package data set to allocate, copy, and display data sets.

It is merely a mechanism to provide an easy installation process for the system programmer who is installing the tool.

To install the dialog, follow the directions that follow:

If this exec resides in a SYS1.SAMPLIB that is uncatalogued, as can happen if you are installing a new version of MVS, then your first step is to copy both the SPPINST and SPPPACK members into a private, catalogued partitioned data set.

Assuming you are invoking the exec from SYS1.SAMPLIB, then type in the following command at the TSO ready prompt. Otherwise, substitute your private, catalogued partitioned data set name for SYS1.SAMPLIB in the following command:

- EX ′SYS1.SAMPLIB(SPPINST)′ ′′′SYS1.SAMPLIB(SPPPACK)′′′

Or, if the exec resides on another ID:

- EX ′myid.SAMPLIB(SPPINST)′ ′′′myid.SAMPLIB(SPPPACK)′′′

The tool consists of five parts:

- REXX Programs

  Reside in xxxx.PARMLIB.EXEC

- ISPF Panels

  Reside in xxxx.PARMLIB.PANELS

- ISPF Messages

  Reside in xxxx.PARMLIB.MESSAGES

- Notes and Instructions

Reside in xxxx.PARMLIB.NOTES

- LOAD Modules

  Installed in SYS1.MIGLIB

The first four data sets will have your user ID as the high level qualifier, unless you specify a different prefix via PROFILE PREFIX(nnn).

If you had entered PROFILE PREFIX(XYZ) at the TSO ready message, the data sets would be as follows:

```
XYZ.PARMLIB.EXEC
XYZ.PARMLIB.PANELS
XYZ.PARMLIB.MESSAGES
XYZ.PARMLIB.NOTES
```

Figure 50. Sample of Which Data Sets Would Be Created for PREFIX(XYZ)

Although the files for the tool have been created, they must be appended to the appropriate dynamic ISPF data set concatenations before the tool can be invoked. SYS1.MIGLIB must be available for LOADing various modules when needed by the tool.

An attempt has been made to minimize the size of these files. The sizes of the data sets are defined such that they should fit into the first extent to avoid an 'E37'X ABEND.

Command syntax for installing the package:

- EX 'SYS1.SAMPLIB(SPPINST)' '''packagedsn'' </<Q><R>>'

where packagedsn equals the fully qualified package data set name.

If the package dsn is not "SYS1.SAMPLIB" but a user data set, the command used to install the package could be the following:

- EX 'SYS1.SAMPLIB(SPPINST)' '''user.SAMPLIB(SPPPACK)'''

SPPINST has two options that may be specified during runtime. Runtime options will follow a "/" character.

- Q -- Quiet Mode

  This mode will not issue any informational messages. Any error messages during the installation process will be displayed, however. See Figure 51 for a sample command for Quiet Mode.

```
EX 'SYS1.SAMPLIB(SPPINST)' '''SYS1.SAMPLIB(SPPPACK)'' /Q'
```

Figure 51. Sample Command for Quiet Mode

- R -- Replace

  This mode would be used if the tool already exists, and you would like to replace it with an updated version, or you would like to refresh the tool data sets.

Important

This option deletes and reallocates each data set. It is therefore recommended that no user data be kept in the package data sets because execution of this tool with the replace option will destroy any user data in the data sets.

See Figure 52 for a sample command for Replace Mode.

```
EX 'SYS1.SAMPLIB(SPPINST)' '''SYS1.SAMPLIB(SPPPACK)'' /R'
```

Figure 52. Sample Command for Replace Mode

NOTE: When using the replace option, the data sets that comprise the tool must not be appended to any system files. TSO will refuse permission to delete a data set that has been appended to dynamically allocated ddnames.

5.1.1 Restrictions for Installation

- SPPINST must execute without the assistance of any other REXX Execs.
- SPPINST must not use any ISPF services.
- SPPINST must execute from native TSO.
- SPPINST code is restricted to columns 1-72
- None of the data sets that are being built by the SPPINST routine can be currently allocated.

5.1.2 Software Requirements

The following products are required:

- TSO/E Version 2
- REXX
- ISPF/PDF Version 4 Release 1

5.1.3 Compatability Issues

The compatability issues are:

- The tool is available in SP5.2.0 and OS/390 R1 with OW16294.
- The tool is portable across ESA and OS/390 systems
- The OS/390 Release 2 version of the tool supports concatenated parmlibs.

5.1.4 Installation Considerations

Once installation is complete, the tool can now be invoked from within ISPF using the following command:

- COMMAND ===> TSO ex 'myid.PARMLIB.EXEC(SYSPARM)'

Note: If your installation has installed the tool under a different name, you may specify a different name for the tool to use. For example, if the tool data sets have a name of "myid.myfiles.*," you would type:

- TSO ex "myid.myfiles.EXEC(SYSPARM)" "myid.myfiles"

5.1.5 Help Panels

The tool has at least one help panel for each ISPF Panel. Additional information about appending the tool files and operational characteristics will be copied into myid.PARMLIB.NOTES at installation time.

Chapter 6. OS/390 Release 2 Global Resource Serialization

In a multisystem sysplex, MVS provides a platform of services that applications and subsystems can exploit. Multisystem applications reside on more than one system and give a single-system image to the users. This requires serialization services that extend across systems.

Global resource serialization (GRS) provides the services to ensure the integrity of resources in a multisystem environment. Combining the systems that access shared resources into a global resource serialization complex enables serialization across multiple systems.

GRS has been available in MVS since MVS/SP 1.3. However, activating GRS in cross-system serialization (global) mode in a multisystem complex has been optional. MVS/ESA Version 4 introduced the concept of “guaranteed serialization” in a sysplex, which requires that global resource serialization be active in cross-system serialization mode. Any multisystem sysplex is thus required to activate global serialization for all members of the sysplex.

All previous releases of MVS/ESA and OS/390 that supported GRS consisted of one or more systems connected to each other in a *ring* configuration.

OS/390 Release 2 introduces the ability to design a GRS complex in a *star* configuration. It is still possible to run OS/390 Release 2 using the ring configuration. The main change in the “star mode” method of processing requests for global resources is that they are managed through a coupling facility lock structure, and that each system maintains only its own global requests, while in the “ring mode” method of processing, each system in the GRS complex maintains all the requests for all global resources in the complex.

The new star configuration has positive effects on sysplex growth, response time for global requests, and processor and storage resources consumption.

In this chapter, the following sections describe GRS in OS/390 Release 2:

- GRS services overview
- GRS star rationale
- GRS global requests processing in star configuration
- GRS star design and planning
- GRS star implementation and migration
- GRS star operation
- GRS star lock structure rebuild
- GRS star tuning considerations
- RNL processing and considerations

6.1 GRS Services Overview

MVS provides four types of resource serialization services: ENQ/DEQ, RESERVE/DEQ, SETLOCK, and global resource serialization latch manager services.

The global resource serialization latch manager is a service that authorized programs can use to serialize resources within an address space or, using cross memory capability, within a single MVS system.

The MVS SETLOCK service controls access to the system's serially reusable resources. The serially reusable resources that can be serialized with the SETLOCK service within an MVS image are predetermined and are assigned a separate lock. The users of SETLOCK service must be running in supervisor state and PSW key 0.

Both the SETLOCK and global resource serialization latch manager services are for local resources. For that reason they are not considered further in this book.

6.1.1 ENQ/DEQ Service

The ENQ service attempts to assign control of a resource to the current requester. The resource to be serialized is identified by name and scope. There are three levels of serialization:

STEP The resource is unique to a single address space.
SYSTEM The resource is unique to a single system.
SYSTEMS The resource is unique to the GRS complex.

The requester also identifies the type of serialization (shared or exclusive).

Note: Even if there is only one system in the global resource serialization complex, the scope SYSTEM and SYSTEMS ENQ requests for the same resource name identify two *different* resources.

The DEQ service releases serialization of one or more held resources.

6.1.2 RESERVE/DEQ Service

The RESERVE service "hardware reserves" is a device for use by a particular system; it must be issued by applications requiring serialization on a named resource (a data set) resident on a shared DASD. A RESERVE by default has a scope of SYSTEMS. When the reserving program no longer needs the reserved device, it should issue a DEQ macro to release the resource.

The hardware reserve prevents any I/O access from any other system in the complex to the device that has an outstanding RESERVE.

Serializing access to data sets on shared DASD resident resources through a hardware reserve generally protects the resource. However, it also creates several critical problems:

- When a requester on one system issues a RESERVE macro to obtain control of a data set, no other systems can access any device while it is reserved. All I/O activity from other systems against a reserved device must wait until the system that issued the reserve releases the device.

  Because the RESERVE ties up an entire device, it greatly increases the chance of an interlock (also called a deadlock) occurring between two tasks on different systems.

- A single system can monopolize a shared DASD device when it issues multiple RESERVE requests for the same device. The system does not release the device until DEQ requests are processed for all outstanding RESERVE requests.

- A system reset while a reserve is outstanding terminates the reserve. The loss of the reserve can leave the resource in an unpredictable state.

GRS allows the RESERVE requests to be converted to SYSTEMS scope ENQs, which makes the systems and the resources more available.

Table 3 summarizes the ENQ/DEQ and RESERVE/DEQ processing.

Table 3. ENQ/DEQ and RESERVE/DEQ Serialization in MVS

Scope of Serialization Request:	**Service to Use:**
Within an address space	The ENQ and DEQ macros with a scope of STEP
Within a single MVS system	The ENQ and DEQ macros with a scope of SYSTEM
Across multiple MVS systems	• The RESERVE and DEQ macros with a scope of SYSTEMS for shared DASD resident resources • The ENQ and DEQ macros with a scope of SYSTEMS

6.1.3 GQSCAN Service

The GRS Queue SCAN is the name of the service provided by GRS to allow tasks to query the status of GRS-managed resources and their requesters.

6.1.4 Resource Name List Service

GRS allows an installation to modify the scope on the ENQ/DEQ and RESERVE processing without changing the applications using the RNLs. The RNLs are specified in the parmlib member GRSRNLxx, have a GRS complex scope, and operate at the following three levels:

- SYSTEM inclusion RNL
  Resources in this list are “promoted” to SYSTEMS ENQs.
- SYSTEMS exclusion RNL
  Resources in this list are “demoted” to SYSTEM ENQs.
- RESERVE conversion RNL
  Hardware reserves for resources in this list are suppressed and issued as SYSTEMs ENQs.

There is no change in the RNL processing with GRS star support. For more information on RNL see 6.10, “Resource Name List Processing” on page 104, and 6.11, “RNL Considerations” on page 109.

6.2 Background of GRS Star Configuration

The GRS star configuration introduces a significant change in the method GRS uses to coordinate ENQ/DEQ global resource serialization requests in a sysplex environment. To understand the rationale behind the design, an overview of the GRS ring follows.

6.2.1 GRS Ring Topology

The GRS ring topology has been available since MVS/SP Version 1.3. In the design, the configuration is viewed as a ring of systems.

Figure 53 on page 65 shows an example of a ring complex configuration.

A global resource serialization complex consists of one or more systems connected to each other in a *ring* configuration using:

- XCF communication paths (CTCs) and signaling paths through a coupling facility, for the MVS systems that belong to a sysplex. If all MVS systems belong to a sysplex, then GRS uses XCF services to communicate along the GRS ring; this configuration is called a sysplex matching complex.
- Global resource serialization managed channel-to-channel (CTCs) adapters; for the MVS systems that do not belong to a sysplex, this configuration is called a mixed complex.

Logically, GRS communication follows a sequential ring philosophy, with GRS message traffic flowing to each system in turn. This message traffic consists of a Ring System Authority (RSA) message, which contains details of all the global resource serialization requests within the GRS complex. There is a single RSA message token that is passed sequentially around the GRS ring, pausing in each system for local processing and being updated with any serialization changes that have occurred since the last pass of the message.

Each request for global resources (ENQ or DEQ) is thus circulated around each system in the ring using the RSA token before actually granting the request. In this approach, all systems are peers, with each system maintaining a local copy of the entire queue of global resource requests.

The ring topology does not scale very well for the following reasons:

- Every system′s storage utilization increases with each additional system.
- ENQ delay time increases with each additional system.
- Ring disruption and rebuild time increase with each additional system.
- In a heterogeneous sysplex, large systems overpower small systems with requests.

The star topology addresses each of these ring complex scaling problems.

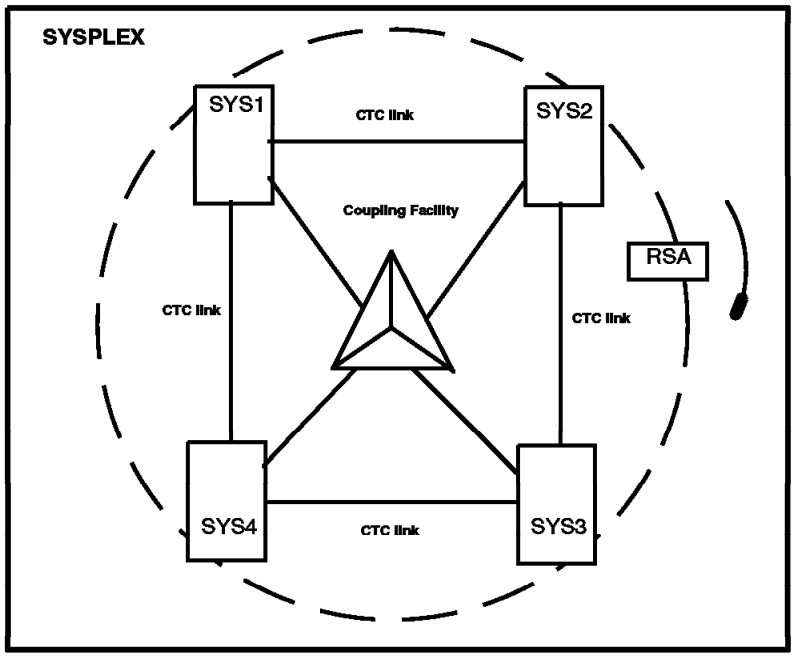


Figure 53. Full Redundancy GRS Ring Topology

6.2.2 GRS Star Topology

With the introduction of the GRS star support, GRS will use the contention detection and management capability of the XES lock structure to determine and assign ownership of a particular global resource. Each system maintains only a local copy of its own global resources, and the GRS coupling facility lock structure has the overall image of all system global resources in use.

A GRS star complex configuration uses:

- A coupling facility lock structure.
- XCF communication paths for global GQSCAN service. See 6.3.4, “Global GQSCAN Processing” on page 72 for additional information.

Figure 58 on page 74 shows a GRS star complex, where the global ENQ/DEQ services use the coupling facility lock structure to manage and arbitrate the serialization requests, and the global GQSCAN service uses the XCF services to communicate across the sysplex.

The highlights of the GRS star design are:

- Real storage consumption
- Processing capacity
- CPU consumption
- Availability and recovery
- Ring versus star topology

All of these are discussed in the following sections.

6.2.3 Real Storage Consumption

In a GRS ring configuration, each system in the ring maintains a queue of all global resource serialization requests for the entire sysplex. Because of the frequency of ENQ/DEQ processing, each system uses real storage that is proportional to the number of outstanding resource requests and the number of systems in the sysplex. Basically, in a GRS ring the amount of real storage consumed by GRS in each system goes up linearly with the number of systems in the sysplex.

In the case of the GRS star support, no system in the sysplex maintains a complete queue of all the global resource requests for the entire sysplex. Because of this approach, the real storage consumption by GRS for each system is governed only by the number of requests from that system.

6.2.4 Processing Capacity

In a GRS ring, all global resource serialization requests are circulated around the ring in a single message buffer called the ring system authority (RSA). Basically, the RSA can be viewed as a floating data area that is shared by all systems in the sysplex, but owned by only one system at a time.

To handle a global resource request, the originating system must place the request in the RSA, and pass it around the ring so all other systems are aware of the request.

Clearly, since the RSA is a shared data area of limited size, the number of requests that each system can place in the RSA diminishes as the number of systems sharing the RSA increases. Also, as the number of systems in the ring increases, the length of time for the RSA to be passed around the ring increases. The net effect of this is that the GRS processing capacity goes down as the number of systems in the ring is increased (that is, fewer requests per second are handled by the ring).

In the case of the GRS star support, this problem is eliminated since all of the requests are handled by mapping ENQ/DEQ requests to XES lock structure requests. With this approach, there is no need to wait for the serialization mechanism, the RSA, to be held by this system before processing the global resource request. Being able to process requests as they are received will improve processing capacity. For instance, there can never be a situation where the system will have to defer excess requests (when the RSA is full).

6.2.5 Responsiveness

Very closely related to the capacity problem is the response time that it takes for GRS to handle a request. As can be seen from the capacity discussion, an increase in the number of systems in the GRS ring results in an increase in the average length of time a given system must wait before it can receive control of the RSA to initiate a new request. Additionally, the length of time that it then takes to circulate the request around the ring before it can be processed is also increased.

As a result, the GRS response time on a per request basis increases linearly as the number of systems in the ring is increased. In the case of the GRS star support, the same processing flow that addressed the capacity constraint also addresses the problem of responsiveness.

6.2.6 CPU Consumption

In a GRS ring, each system must maintain a view of the entire global resource queue. This means each system must not only process the requests originating on it, but also must process the global requests originating on all the other systems in the ring. The effect of this is that each global ENQ/DEQ request generated in a ring consumes processor time on all the systems in the sysplex, and as the sysplex grows, the amount of processor time GRS consumes across the sysplex grows.

In the case of the GRS star support, the overhead of processing an ENQ/DEQ request is limited to only the system on which the request originates. Thus, the total processor time consumed across the sysplex will be less than that consumed by a GRS ring.

6.2.7 Availability and Recovery

In a GRS ring, if one of the systems fails, all the other systems are affected during the ring reconstruction time. None of the remaining systems in the sysplex can process any further ENQ/DEQ requests until the ring is reconstructed.

To rebuild the ring, each of the remaining systems must resynchronize and rebuild their view of the global resource queue. The amount of time this may take depends on the number of systems in the sysplex and the number of requests in the global resource queue.

In the case of the GRS star support, availability is improved by the fact that there is less interdependency between the systems in the GRS complex. Whenever a system fails, no action need be taken by the other systems in the sysplex, because there is no resource request data kept on any other system for the requesters on the failing system.

All that needs to occur is the analysis of the contention for the resource, which could determine a new owner (or owners) of the resource, should the resource have been owned by the requester on the failing system.

This approach allows for a much quicker recovery time from the systems remaining in the sysplex, since there is no longer a need to clean up the resource queues or determine a new ring topology for the remaining systems.

6.2.8 Ring versus Star Topology

When choosing a topology configuration, the following considerations apply:

The GRS ring configuration is required when you:

- Have no coupling facility available
- Have a mixed ring configuration, with systems that do not belong to the sysplex participating in the GRS complex.

The GRS star configuration is suggested for all parallel sysplex configurations. The GRS star configuration is recommended when you have a:

- New installation and coupling facility is available
- Large complex of four or more systems
- Heterogeneous set of machines

Note: The GRS star configuration allows the sysplex growth, and also is of value to installations currently running a sysplex because of the improved responsiveness, the reduced consumption of processor and storage, and the better availability and recovery time.

6.3 GRS Star Processing

As stated before, in a star complex, global resource serialization uses the lock services of the cross system extended services (XES) component of MVS to serialize access to global resources, and the design requires that all systems in a star complex must be members of the same sysplex, and be connected to a coupling facility containing the global resource serialization lock structure.

6.4, “GRS Star Configuration Design and Planning” on page 74 shows an example of a star complex configuration. Figure 58 on page 74 shows an overview of a four-system star complex.

6.3.1 Global Processing Overview

In a GRS star complex, when an ENQ, RESERVE, or DEQ request is issued for a global resource, the request is converted to an XES lock request. The XES lock structure coordinates the requests it receives to ensure proper resource serialization across all systems in the complex, and notifies the originating systems about the status of each request. Based on the results of these lock requests, GRS responds to the requester with the result of the global request.

As can be seen in Figure 54 on page 69, each system in the GRS star complex has a “server task” (dispatchable unit) executing in the GRS address space. All ENQ, DEQ, RESERVE, and GQSCAN requests for global resources that are issued from any of the address spaces on a system are always first passed to the GRS address space on the requester′s system. In the GRS address space, the server task performs the processing necessary to handle a given request and interfaces with the XES lock structure, if necessary, to process the request. In case of a GQSCAN request for global resource information, the server task packages the request for transmission and sends it to each of the other server tasks in the GRS complex using the XCF services. This is necessary because the status of global requests is maintained at system level.

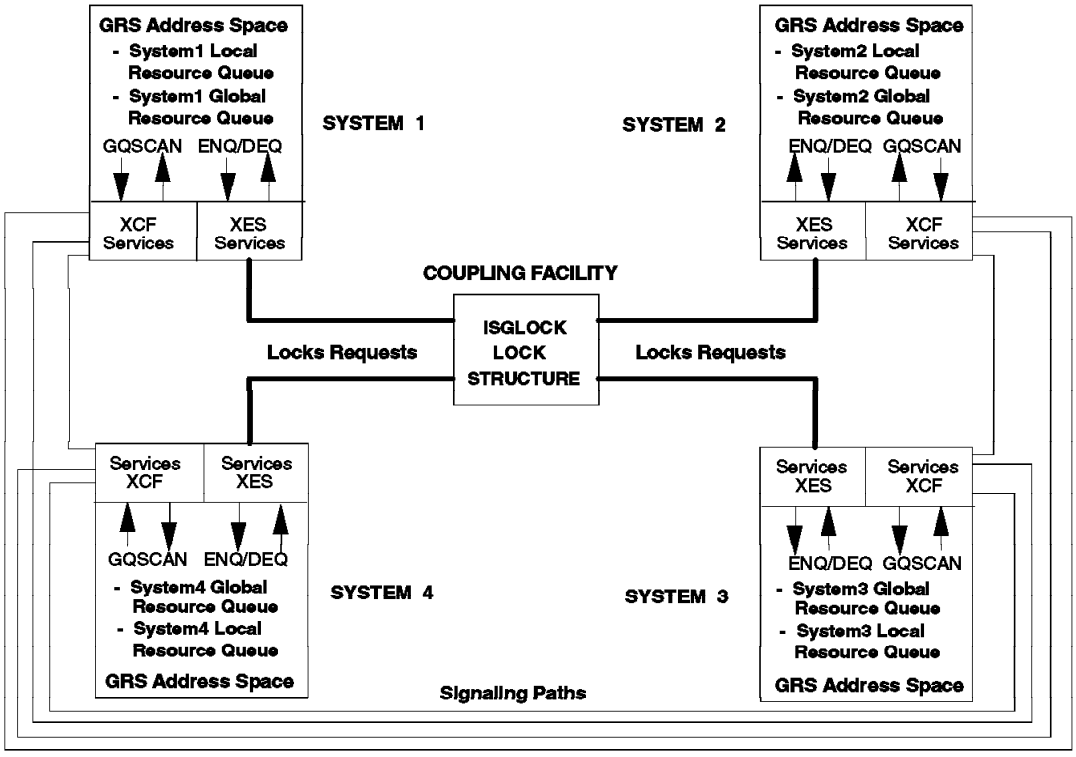


Figure 54. Overview of the Global Resource Serialization Star Complex

6.3.2 Global ENQ Processing

In a GRS star complex, no system maintains a complete view of the outstanding global resource requests, in contrast with the GRS ring philosophy of maintaining a complete view of the global resource requests on all systems in the complex. Instead, each system maintains a queue of each of the local requests, called the “System Global Resource Queue.” Arbitrating requests for global resources from different systems in the complex is managed by putting a subset of the information from the system global resource queue into the user data associated with the lock request made to the XES Lock Structure for a particular resource. For additional information see *MVS Programming: Sysplex Services Guide*.

In a GRS star complex, requests for ownership of global resources are handled through a lock structure on a coupling facility that is fully interconnected with all the systems in the sysplex. GRS interfaces with the XES lock structure to reflect a composite system level view of interest in each of the global resources for which there is at least one requester. This interest is recorded in the user data associated with the lock request.

Each time there is a change in the composite state of a resource, GRS updates the user data reflecting the new state of interest in the XES lock structure. In general, this composite state is altered each time a change is made to the set of owners of the resource or the set of waiters for the resource. ENQ processing

for the RET=NONE and RET=ECB keywords is illustrated at a high level in Figure 55 on page 70.

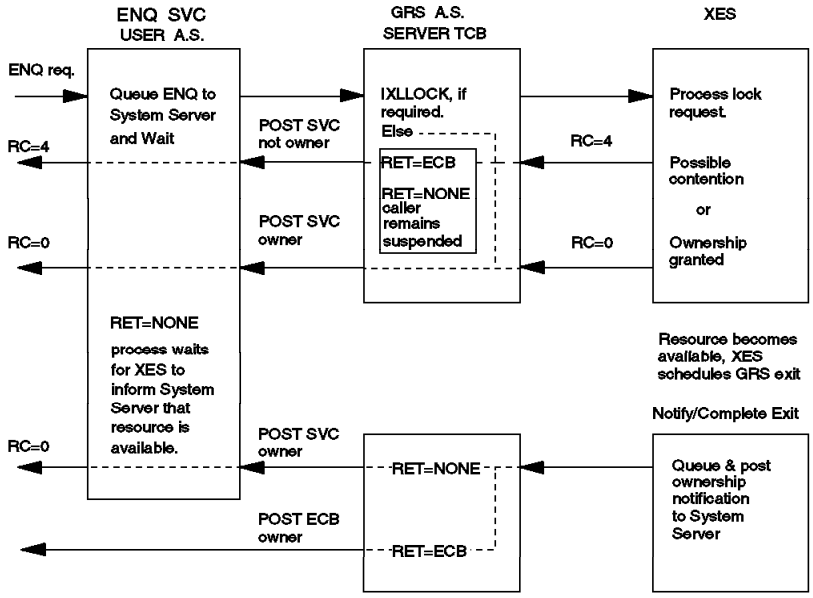


Figure 55. Overview of Global ENQ Processing

Each time an ENQ request is received, the server task analyzes the state of the resource request queue for the resource. If the new request alters the composite state of the queue, an IXLLOCK request is made to reflect the changed state of the resource for the requesting system. If the resource is immediately available, the IXLLOCK request indicates that the system can grant ownership of the resource.

If the XES lock structure cannot immediately grant ownership of the resource, the IXLLOCK request completes with the return code X′04′, and the server task holds the request pending and the requester in a wait state until the completion exit is driven for the request.

Only one IXLLOCK can ever be in progress from a particular system and for a particular global resource at one time. Requests for a resource that are received while a request is in progress are held pending until all preceding requests for the resource have been through IXLLOCK processing. The requests are processed in FIFO order.

If contention exists for the resource, the requester is not granted ownership until some set of requesters dequeue from the resource. When the requester is to be assigned ownership of the resource, the contention exit, driven as the result of a DEQ request, drives the notify exit of the server task that is managing the requester. The notify exit informs the server task of the ownership change. Finally, the server task informs the requester to continue by posting for RET=NONE the SVC or posting the requester′s ECB for RET=ECB. The requester now owns the resource.

6.3.2.1 ENQ RET=TEST

ENQ RET=TEST is a method through which a program can test if a particular resource request would have been granted, had it actually been made. In the design of GRS star complex, the RET=TEST is not fully supported. The implementation only checks to see if there is contention locally for the global resource; no IXLLOCK request is made in the processing of such ENQ request. This is because of a limitation of the XES locking services which have no external way to determine if a particular lock is currently in contention, without actually forcing an escalation of the lock.

Conversely, in a GRS ring complex, this interface provides information that is correct for the last time that the system held the RSA and updated its local version of the global resource.

For local resources (SCOPE STEP | SYSTEM), the RET=TEST interface provides the correct information for the resource.

If an application wants to determine if a global resource is available, the GQSCAN service can provide equivalent function.

6.3.3 Global DEQ Processing

In a GRS star complex, releasing a resource has the effect of altering the system′s interest in the resource. GRS alters the user data associated with the resource, and issues an IXLLOCK request to reflect the new interest. Figure 56 illustrates the global DEQ process.

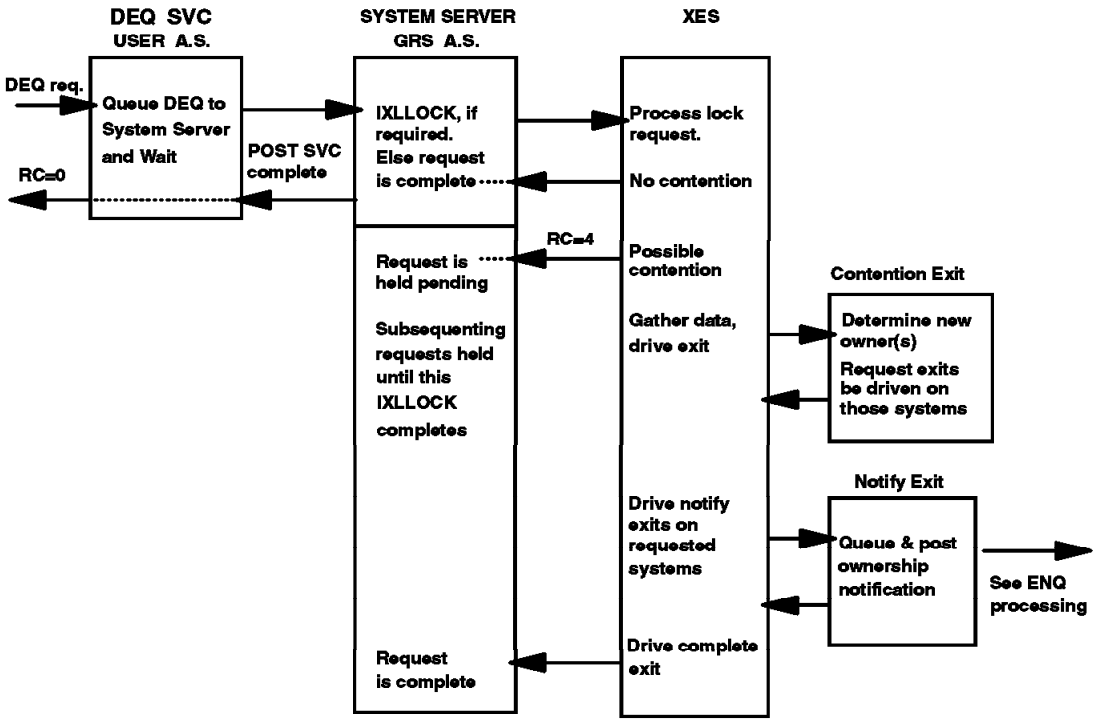


Figure 56. Overview of the Global DEQ Process

After resuming the user, the server task performs local processing and issues an IXLLOCK request if the DEQ results in an alteration of the system′s composite state of the resource. If there is no global contention on the resource, XES returns from the IXLLOCK service with a return code of X′0′, indicating that no asynchronous ownership processing will occur.

When IXLLOCK completes with return code X′04′, XES indicates that there may be contention for the resource and that asynchronous processing occurs for this resource. The server task places this resource into a pending state, keeping all subsequent requests queued in FIFO order until the asynchronous processing completes.

XES gathers the user data from each of the systems interested in the resource and presents them to the contention exit that GRS provides. The contention exit determines, based on each of the system′s composite states, which systems have owners of the resource. The contention exit causes the notify exit on each of the systems to be driven. The notify exit informs the server task of the change in ownership of the resource. The server task posts the new owner (or owners) of the resource.

In a GRS star complex, a DEQ request is handled as a fast DEQ under the following conditions:

- The DEQ is for a single resource.
- The resource request (ENQ for the resource) has already been reflected in the GRS lock structure.
- The request for the resource is not the target of a MASID/MTCB request.

This is equivalent to the implementation of fast DEQ in a GRS ring complex.

If the resource request has not yet been reflected in the GRS lock structure, the DEQ requester waits until the IXLLOCK for that request has been completed. Following this, the request is resumed in parallel with the completion of the DEQ request.

6.3.4 Global GQSCAN Processing

In a GRS star complex, the basic flow of a GQSCAN request for global resource is illustrated by Figure 57 on page 73.

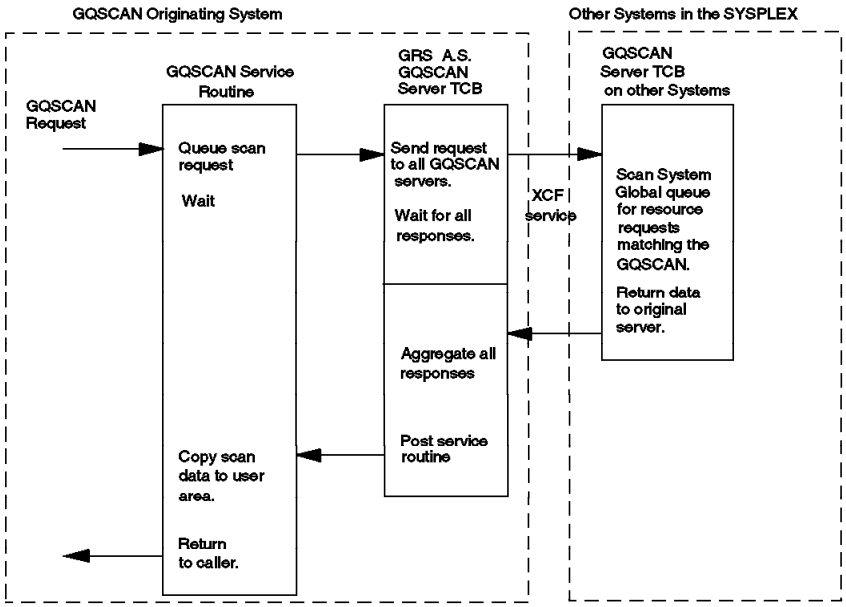


Figure 57. Overview of the GQSCAN Request for Global Resource Data

In the case of a GQSCAN request for global resource information, the request is passed from the issuing address space to the GRS server task on that system, and the GQSCAN request is waited. The server task then packages the request for transmission and sends it to each of the server tasks in the GRS star complex.

Each server task in the complex scans their system global resource queue that matches the GQSCAN selection criteria, responding to the originating system with the requested global resource information (if any) and a return code. The originating server task waits for responses from all of the server tasks in the complex and builds a composite response to be returned to the caller.

Waiting the issuer of GQSCAN when global resource information is requested is a significant change from the way GQSCAN works in a GRS ring complex. In a GRS ring complex, GQSCAN processing is performed synchronously with respect to the caller's task.

A new GQSCAN macro option is provided to allow the issuer to indicate whether or not cross-system processing for global resource information should be performed.

The new "no cross-system" option (XSYS=NO) is provided primarily for the GQSCAN that cannot be put into a wait, and does not require that data about requesters on other systems in the complex. XSYS=YES is the default option.

The GQSCAN abend 09A is removed and changed to new return and reason codes. The output from GQSCAN is enhanced to indicate resource ownership as the result of a MASID/MTCB request. The current output from GQSCAN inaccurately reports these requesters as waiting.

6.4 GRS Star Configuration Design and Planning

The design of the global resource serialization allows for an unlimited number of systems. The star method for serializing global resources can support complexes up to 32 systems efficiently and effectively. See 6.2, “Background of GRS Star Configuration” on page 63 for GRS star design rationale.

Because it is possible for a single installation to have two or more global resource serialization complexes, each operating independently, it should be considered that the independent complexes cannot depend on GRS to share resources. Therefore, there should be no common links made available to GRS star on any two complexes.

To avoid a data integrity exposure, ensure that no system outside the complex can access the same shared DASD as any system in the complex. If that is unavoidable, as often happens, you must serialize the data on the shared DASD with the RESERVE macro. The sample GRS exit ISGGREX0 given in Appendix C, “ISGGREX0 Sample Exit” on page 199 may help in managing DASD sharing among and outside sysplexes. For additional information on GRS RNL exits see 6.10.2, “User Exit for Scanning the RNL” on page 108.

As stated before, in a star complex, GRS uses the lock services of the cross-system extended services (XES) component of MVS to serialize access to global resources, and the GRS star method for processing global requests operates in a sysplex like any other MVS component that uses the coupling facility. Therefore a coupling facility with connectivity from all systems in the complex is required.

The overall design of GRS star makes the connectivity configuration relatively simple. Figure 58 shows a GRS star connectivity configuration.

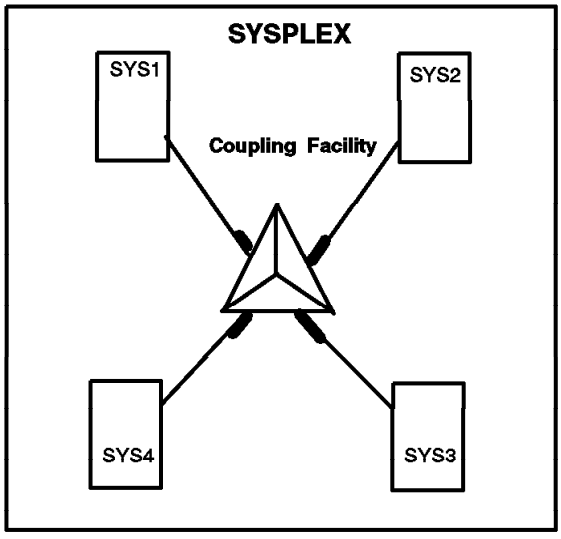


Figure 58. GRS Star Connectivity Configuration

MVS requires DASD to be shared by all systems in the sysplex for the sysplex couple data set. An alternate data set is recommended to be used to facilitate the migration from a ring to a star complex, and for availability reasons. For additional information see 6.5, “GRS Star Implementation and Migration” on page 75.

MVS stores information related to sysplex, systems, and XCF groups (such as global resource serialization), on the sysplex couple data set. GRS star stores RNL information into the sysplex couple data set.

To manage certain aspects of GRS star, the following policies are required:

- Coupling facility resource management (CFRM) policy, to define the GRS lock structure to MVS
- Sysplex failure management (SFM) policy, to define how MVS is to manage system and signaling connectivity failures

In designing a GRS star configuration, verify that the following requirements are met:

- Hardware requirements:
  - A fully interconnected coupling facility (accessible by all systems)
  - Coupling facility links
- Unlike a GRS ring, all systems in a GRS star complex must be in the same sysplex.
- All systems in a star complex must be connected to a coupling facility containing the GRS lock structure whose name should be ISGLOCK. For availability reasons in case of coupling facility failure, a second coupling facility should be available to rebuild the structure used by GRS.
- No channel-to-channel (CTC) connection of systems, other than those managed by XCF, will be supported by global resource serialization. GRS star does not use CTC links.
- A GRS star complex and a ring complex cannot be interconnected, and therefore they cannot coexist in a single GRS complex.
- GRS star does not support systems that are not in the sysplex. A mixed complex is not possible.
- All systems must be at OS/390 Release 2 level or above.

6.5 GRS Star Implementation and Migration

Four major elements are required to run a GRS star complex. They are:

- Sysplex couple data set definition
- GRS lock structure definition
- Parmlib changes
- Migrating from GRS ring to GRS star mode of operation

The following are the steps necessary to implement the GRS star complex. It is assumed that a sysplex has been already implemented, that a coupling facility is operational, and that a GRS ring complex that matches the sysplex is working. This also means that the RNLs have been implemented according to the installation needs.

1. Before switching GRS to a star configuration, all MVS systems must be at OS/390 Release 2 or higher.

   All the following steps, up to GRS switch to star, can be done with only one system in the sysplex at OS/390 Release 2 level.

2. If applicable, convert from mixed GRS complex (complex not matching the sysplex) to a “pure” sysplex GRS complex.

3. Define the sysplex couple data sets for the star complex.

   A couple data set must be formatted with the new GRS parameter before initializing a star complex or migrating to a star sysplex from a ring complex. The GRS parameter is in the DEFINEDS statement for the sysplex couple data set. The OS/390 Release 2 version of the IXCL1DSU utility must be used; if not, the following diagnostic messages are issued:

```
                      DATA TYPE(SYSPLEX)
                           ITEM NAME(GRS) NUMBER(1)
IXC291I ITEM NAME NOT DEFINED, GRS
```

   For more information on the IXCL1DSU utility, see *OS/390 MVS Setting Up a Sysplex*.

   The DEFINEDS statement for the TYPE(SYSPLEX) couple data set is as follows:

```
DEFINEDS SYSPLEX(WTSCPLX1)
     DSN(SYS1.XCF.CDS03) VOLSER(TOTDS1)
     MAXSYSTEM(16)
     CATALOG
   DATA TYPE(SYSPLEX)
   ITEM NAME(GRS) NUMBER(1)
   ITEM NAME(GROUP)  NUMBER(100)
   ITEM NAME(MEMBER) NUMBER(200)
```

   The ITEM NAME(GRS) NUMBER(1) record allocates storage in the couple data set for use by GRS and is used to maintain the RNL. The support for star complex *does not* introduce changes in RNL functions or in the dynamic RNL changes. For GRS star use it is recommended you allocate and format a second couple data set to be used as an alternate, for availability reasons.

4. Make the newly formatted couple data sets available to sysplex with the SETXCF command if the sysplex is already active.

   Note: A sysplex couple data set formatted with a GRS record may be used in a sysplex with MVS/ESA Version 5 and upward systems.

   The following is an example of the SETXCF operator commands:

```
SETXCF COUPLE,PSWITCH   <-- Required if an alternate is active.
                        Switches the current alternate sysplex couple
                        data set to become the primary sysplex couple
                        data set. This command removes the primary old
                        sysplex couple data set from the service.

SETXCF COUPLE,ACOUPLE=dsname1
                        Specifies the data set to be used
                        as alternate sysplex couple data set
                        is the newly formatted with GRS item.

SETXCF COUPLE,PSWITCH
                        Switches the current alternate sysplex couple
                        data set to become the primary sysplex couple
                        data set. This command removes the primary old
                        sysplex couple data set from the service.

SETXCF COUPLE,ACOUPLE=dsname2
                        Add dsname2, newly formatted data set, as
                        alternate couple data set.
```

5. Update the COUPLExx parmlib member with the new sysplex couple data set names for subsequent IPLs.

```
    COUPLE SYSPLEX(&sysplex)
          PCOUPLE(SYS1.XCF.CDS01)
           ACOUPLE(SYS1.XCF.CDS02)
```

6. Update the CFRM couple data set.

   GRS uses the XES lock structure to reflect a composite system level of interest for each global resource. The interest is recorded in the user data associated with the lock request.

   The name of the lock structure must be ISGLOCK, and the size depends on the following factors:

   - The size and type of systems in the sysplex
   - The type of workload being performed

   GRS requires a lock structure large enough for at least 32K entries. The number of lock entries used depends on the number of outstanding global requests. In a typical environment, the major contributors for active global requests are related to the number of data sets and databases allocated (qname SYSDSN), VSAM data sets opened (qname SYSVSAM), and ISPF used data sets (qname SPFEDIT). It is assumed that the major name SYSDSN is in the RNL inclusion list.

   A small sysplex, made up of smaller processors and running a transaction processing workload, needs a smaller structure than a larger sysplex that is composed of large processors and running a batch and TSO workloads combination.

   It is recommended you use the following formula to determine the size for ISGLOCK.

```
structure Size (in K) = INT(((Peak # Global Resources * 6)/1024) + 1) * 10 )

Where:

Peak # of Global Resources = The number of unique globally managed resources
                             (SYSTEMS ENQs and converted RESERVEs) outstanding
                              measured at a time of peak load
                      INT = The integer part of the calculated expression
```

If the calculated structure size is less than 10M (10000K), you should start using a size of 10M (10000K) that, according to the formula, corresponds to 170K global outstanding requests. GRS uses ten bytes for each lock entry. Therefore, given a number of outstanding global requests, the formula gives a structure size for a number of locks six times the number of global requests. This is to keep the volume of false lock contention low (see 6.9.2.4, “ISGLOCK Structure” on page 100).

The utility program ISGSCGRS, available in SYS1.LINKLIB of OS/390 Release 2, can be used to verify the number of outstanding global requests. The program issues a WTO every 10 seconds indicating the number of outstanding requests and runs for two minutes. To execute the program, use the following JCL, which can be found in SYS1.SAMPLIB member ISGCGRS of OS/390 Release 2. Samples of issued WTOs are also included.

```
//ANYNAME  JOB (999,POK),CLASS=A,MSGCLASS=X,TIME=1440
//STEP001 EXEC PGM=ISGSCGRS

11:49.06 ANYNAME  00000090  IEF403I ANYNAME - STARTED - TIME=12.11.49
11:49.14 ANYNAME  00000090  +Number of global resources outstanding:   00002253
11:59.20 ANYMANE  00000090  +Number of global resources outstanding:   00002253
12:09.26 ANYNAME  00000090  +Number of global resources outstanding:   00002253
12:19.32 ANYNAME  00000090  +Number of global resources outstanding:   00002253
12:29.37 ANYNAME  00000090  +Number of global resources outstanding:   00002253
12:39.43 ANYNAME  00000090  +Number of global resources outstanding:   00002253
12:49.49 ANYNAME  00000090  +Number of global resources outstanding:   00002253
12:59.55 ANYNAME  00000090  +Number of global resources outstanding:   00002253
13:09.60 ANYNAME  00000090  +Number of global resources outstanding:   00002252
   //               //                    //
12:13:49 ANYNAME  00000090  IEF404I ANYNAME - ENDED - TIME=12.13.49     0002257
```

Table 4 gives a reference between the number of global active requests and the size of the lock structure.

Table 4. Global Requests and GRS Lock Structure Size

Number of global outstanding requests	Structure size in bytes
32K	1885K
50K	2940K
100K	5970K
150K	8800K
170K	10000K
500K	29307K
1000K	58604K

Because GRS does not support the change of the ISGLOCK structure size using the SETXCF START,ALTER command, it is recommended that the CFRM policy either specifies the SIZE for the structure or SIZE equal to

INITSIZE. The structure size can be dynamically changed using the SETXCF START,REBUILD command.

The REBUILDPERCENT parameter determines, along with the system weights in the SFM policy, if a structure should be rebuilt in case of a connectivity failure. If a system in a GRS star configuration loses connectivity and the structure is not rebuilt, the system is lost (wait X'0A3'). See 6.7, "GRS Lock Structure Rebuild" on page 89 for additional information.

For this reason, the best GRS availability can be achieved specifying REBUILDPERCENT(1) for the ISGLOCK structure. This, of course, requires an alternate coupling facility with the necessary spare capacity. The default value for REBUILDPERCENT is 100, which does not allow the structure rebuild.

The following is an example of a CFRM definition for the GRS lock structure to be used with the utility IXCMIAPU:

```
STRUCTURE NAME(ISGLOCK)
   INITSIZE(10000)
   SIZE(10000)
   REBUILDPERCENT(1)
   PREFLIST(CF01,CF02)
```

For more information on the utility IXCMIAPU, see "OS/390 MVS Setting Up a sysplex."

The size of the lock structure can be tuned by checking the number of false contention occurrences and the number of lock entries used versus the number available in the resource management facility (RMF) report. For references see 6.9, "GRS Measurements and Tuning" on page 91. To increase the size of the structure, update and activate the CFRM policy and then rebuild the structure. See 6.6, "GRS Star Operation" on page 82 for additional information on structure rebuild.

7. Activate the new CFRM policy.

   To activate the new policy, a SETXCF command can be issued as follows:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=anyname
```

   The policy may be activated in a GRS ring configuration with some systems at MVS/ESA Version 5 or OS/390 Release 1 level with no problem, because global resource serialization does not connect to it.

   When the first system joins the star, global resource serialization uses the information about the structure size specified in the CFRM policy and then creates the largest lock structure that fits within that size. Message ISG3371I is issued to document the number of locks that may be used to satisfy global resource serialization requests.

8. Update SFM policy.

   XES uses REBUILDPERCENT (CFRM) and system weights from SFM to determine if a rebuild is allowed for a structure connectivity failure. Therefore, the SFM policy should eventually be updated to modify the system weights according to the installation decision in case of connectivity failure for the ISGLOCK structure. If REBUILDPERCENT(1) has been specified for

GRS lock structure, it is not necessary to change the system weights, because a structure rebuild is always attempted in case of a connectivity failure.

In case of a connectivity failure, OS/390 Release 2 does not have dependencies on the CONNFAIL(YES|NO) parameter of SFM policy to activate XES analysis for structure rebuild. The analysis is done using the REBUILDPERCENT (CFRM) and the system weights specified in SFM policy. APAR OW19718 removes the dependencies on MVS/ESA Version 5 and OS/390 Release 1. For additional information see 7.9, “Migration and Coexistence Considerations” on page 140.

9. Activate the new SFM policy.

   If the system weights have been changed, the policy can be activated in a GRS ring configuration.

   To activate the new SFM policy, a SETXCF command can be issued:

```
SETXCF START,POLICY,TYPE=SFM,POLNAME=anyname
```

10. Update parmlib member IEASYSxx for a star complex.

    Initializing a star complex requires specifying “STAR” on the “GRS=” system parameter in the IEASYSxx parmlib member, or in response to message IEA101A during IPL.

    The START, JOIN, and TRYJOIN options remain the same, except that these options apply to a ring complex only. The revised syntax of the GRS= parameter for GRS star is described below. For a description of what occurs when systems with mismatched GRS= specifications are initialized, see 6.6.1, “Possible Errors” on page 88.

```
{GRS=NONE}   --> GRS is not to perform any global serialization process
    {STAR}   --> system being IPLed is to join a GRS star complex
```

    The PLEXCFG parameter should be MULTISYSTEM; the relation between GRS= and PLEXCFG parameters are the following:

    - Use GRS=NONE and PLEXCFG=XCFLOCAL or MONOPLEX when there is a single-system sysplex and no global resource serialization complex.
    - Use GRS=TRYJOIN (or GRS=START or JOIN) and PLEXCFG=MULTISYSTEM when there is a sysplex of two or more systems and the GRS ring complex uses XCF signalling services.
    - Use GRS=STAR and PLEXCFG=MULTISYSTEM when there is a GRS star complex.

11. Update parmlib member GRSCNFxx for a star complex.

    - Basically, the GRSCNFxx of parmlib is not required when initializing a star complex that uses the default CTRACE parmlib member, CTIGRS00, which is supplied with the system.

      If you want to initialize a star complex using a CTRACE parmlib member other than the default, you must use the GRSDEF statement in the GRSCNFxx parmlib member. The following is a GRSCNFxx example that refers to the GRS configuration shown in Figure 58 on page 74.

```
GRSDEF MATCHSYS(*)       /* GRSDEF FOR SYSTEMS SYS1, SYS2, SYS4      ss

        CTRACE(CTIGRS01) /* PARMLIB MEMBER CTIGRS01 CONTAINS TRACE
                         /* OPTIONS

GRSDEF MATCHSYS(SYS3)    /* GRSDEF FOR SYSTEM SYS3

        CTRACE(CTIGRS03) /* PARMLIB MEMBER CTIGRS03 CONTAINS TRACE
                         /* OPTION
```

- Remove ring-related parameters from GRSCNFxx.

  Apart from the two GRSDEF parameters listed in the example, all the other parameters on the GRSDEF statement (ACCELSYS, RESMIL, TOLINT, CTC, REJOIN, and RESTART) apply only to systems initializing in a ring complex. Although they can be specified on the GRSDEF statement and are parsed and syntax checked, they are not used when initializing systems into a GRS star complex. Global resource serialization ignores the parameters that are not applicable on the GRSDEF statement when initializing systems into a star complex, as well as when initializing systems into a ring complex.

  If the ring parameters are left in the GRSCNFxx parmlib member, there is the potential, due to making an error in the specification of the GRS= parameter in IEASYSxx, to create two GRS complexes. For additional information, see 6.6.1, “Possible Errors” on page 88.

  For this reason, even if it is possible for an installation to create a single GRSCNFxx parmlib member that can be used for the initialization of either a star or a ring complex, it is suggested you have different GRSCNFxx members, one for star and one for ring, and use the GRSCNF= keyword in IEASYSxx to select the proper one. This helps in the transition from a ring to a star complex if the installation elects to use the SETGRS MODE=STAR capability to make the transition.

  For additional information about the GRS parmlib members see *OS/390 MVS Initialization and Tuning Reference*, SC28-1751 and SC28-1752.

12. Migrate from a ring to a star complex.

    The SETGRS command is used to migrate an active ring to a star complex. There is no SETGRS option to migrate from a star to a ring complex; returning to a ring complex requires an IPL of the entire sysplex.

    The following command is used to migrate from a ring complex to a star complex:

    ```
    SETGRS MODE=STAR
    ```

    The SETGRS command can be issued from any system in the complex and has sysplex scope.

    While processing a SETGRS MODE=STAR command, processing is suspended for the global resource serialization ENQ, DEQ, RESERVE, and GQSCAN for global resource data fails with RC=X′0C′, RSN=X′10′. The length of time global resource serialization requesters are suspended may be a few minutes while the ISGLOCK lock structure and global resource serialization sysplex couple data set records are initialized, and changes to the internal GRS control block structures are initialized as well.

Note: The migration should be invoked at a time when the amount of global resource request activity is likely to be minimal.

SETGRS MODE=STAR request is valid under the following conditions:

- GRS is currently running a ring complex that exactly matches the sysplex.
- All systems in the ring complex support OS/390 Release 2 or later.
- All systems in the ring complex are connected to a coupling facility.
- All systems can access the ISGLOCK lock structure on the coupling facility.
- The global resource serialization records are defined on the sysplex couple data sets.
- There are no active dynamic RNL changes in progress.

Once global resource serialization completes the transition to the star complex, the following message is issued:

```
ISG300I GRS STAR COMPLEX INITIALIZATION COMPLETE
```

6.5.1 GRS Star Complex Alternative Method

Another method to bring up a GRS star complex is to “cold” IPL the entire sysplex directly into a GRS star. If, for any reason, it is required to fall back to a GRS ring configuration, a cold IPL of the sysplex is required.

In such an event, the GRS= keyword in the IEASYSxx parmlib member should be changed to specify START, JOIN, or TRYJOIN, and the GRSCNFxx parmlib member should be updated with the required values for ACCELSYS, CTC, RESMIL, and TOLINT if they have been removed.

Note: The GRSRNLxx that are used with the GRS ring can initially be used with a GRS star configuration. For additional information, see 6.9, “GRS Measurements and Tuning” on page 91.

6.6 GRS Star Operation

The normal operation of a GRS star complex is almost unchanged when compared with a GRS ring complex.

To minimize operator intervention during IPL, the following recommendations still apply.

- Specify the keywords GRS, GRSRNL, and GRSCNF in the IEASYSxx parmlib member. For information on how to do this, see 6.5, “GRS Star Implementation and Migration” on page 75. The system comes up issuing information messages and without operator intervention for the GRS operation.
- Specify GRS=STAR in IEASYSxx on each system in the sysplex. The first system that IPLs in the star sysplex initializes the GRS lock structure in a coupling facility, and all the others connect to the lock structure.

The GRS and GRSCNF parameters remain in effect for the duration of the IPL; the only way to change their value is to IPL the system again. As with the ring configuration, it is possible to change the RNLs without having to reIPL the entire complex by using the SET GRSRNL command.

During normal processing, operators can use system commands to switch from ring to star configuration in order to monitor and control a GRS star complex. The system commands related to global resource serialization are:

- SETGRS command, to migrate an active ring to a star complex

  The following is an example of the command:

```
SETGRS MODE=STAR
```

- DISPLAY GRS command, which displays the status of each system in the complex

  The D GRS command displays the status of the star from the point of view of the system where the command is entered. The following is an example of the result of the command and shows the differences with a ring configuration.

```
             18.40.07 ISG343I 18:40:06 GRS

              SYSTEM STATE       SYSTEM STATE
               SYS2  CONNECTING  SYS1   CONNECTED
               SYS3  REBUILDING  SYS4   CONNECTED

              GRS STAR MODE INFORMATION
               LOCK STRUCTURE(ISGLOCK) CONTAINS 1048576 LOCKS

SYSTEM     --> The first system shown is the system on which the
               command was entered

CONNECTING --> The system processing the GRS=STAR parameter

CONNECTED  --> The system is part of the star sysplex.
REBUILDING --> The system is not part of a star complex, but is
               rebuilding the GRS lock structure.  All tasks that
               are trying to obtain global resources are suspended
```

  DISPLAY GRS uses a new message ID to prevent automation failures.

- SET GRSRNL command, which changes the RNLs dynamically. It is possible to dynamically change the RNLs that global resource serialization uses in a star configuration, because the sysplex always matches the complex.

  To change the RNLs currently being used by global resource serialization, set up the GRSRNLxx parmlib members with the new RNLs. Next, issue the SET GRSRNL command on a system that has access to those members. The new RNLs are then communicated to all systems in the complex.

  Note: Even though only one system needs the updated parmlib members to start the change, be sure to copy the updated GRSRNLxx parmlib members to each system′s parmlib. Any system that needs to can then IPL again into the sysplex.

  Global resource serialization ensures that the integrity of all resources is maintained throughout the RNL change. In particular, before an RNL change can complete, special processing may be performed if any jobs are using the resources that are different in the old and new RNLs. Jobs issuing new requests for these resources are suspended until the RNL change is complete.

When jobs are holding affected resources and delaying the change, the following messages are issued on whichever console originated the RNL change:

```
ISG219E RNL CHANGE WAITING FOR RESOURCES TO BE FREED.
        TO LIST DELAYING JOBS, USE ROUTE SYSNAME,DISPLAY GRS,DELAY.

        TO LIST SUSPENDED JOBS, USE ROUTE SYSNAME,DISPLAY GRS,SUSPEND.
ISG220D REPLY C TO CANCEL RNL CHANGE COMMAND, OR S FOR SUMMARY OF RNL
        CHANGE PROGRESS.
```

If the operator chooses not to respond to message ISG220D with a C, the change takes place when all delaying jobs release the affected resources.

However, there might be instances where the operator must either cancel the RNL change command or cancel jobs that hold the affected resources:

- A job that is not cancellable is holding affected resources for a long time.
- A job holding an affected resource cannot DEQ that resource because it is suspended by global resource serialization pending a new ENQ for another affected resource, or else the job is waiting for some other work in the system that has issued an ENQ for an affected resource and has become suspended.
- A job that is suspended by the RNL change is considered more important than the RNL change.

Note: This condition is likely to happen when trying a generic change for qnames like SYSDSN, SYSVSAM, and SPFEDIT.

- VARY XCF,sysname,OFFLINE command, which removes a system from the sysplex. In a GRS star configuration, there is no longer the concept of “ring disruption” (that is, the period of time that GRS is unable to process global resource requests due to the clean up activity required when a system leaves the GRS ring complex). The GRS star processing requires significantly less processing to respond to a system leaving the complex, making the processing similar to a massive set of DEQs being issued at the same time.
- SETXCF START,REBUILD command, which changes the size of the GRS lock structure after the CFRM policy update that changes the structure SIZE value.

  GRS does not support the SETXCF START,ALTER command for its structure. The following is an example of the message issued if an ALTER is attempted:

```
SETXCF START,ALTER,STRNAME=ISGLOCK,SIZE=20000

IXC531I SETXCF START ALTER REQUEST FOR STRUCTURE ISGLOCK 280
REJECTED. REASON: AT LEAST ONE CONNECTION INDICATED THAT ALTER IS NOT
ALLOWED
```

  The following is an example of the sequence of commands that are used to activate the changed CFRM policy and to REBUILD the ISGLOCK structure.

```
SETXCF START,POLICY,POLNAME=CFRM17,TYPE=CFRM

IXC511I START ADMINISTRATIVE POLICY CFRM17 FOR CFRM ACCEPTED
IXC512I POLICY CHANGE IN PROGRESS FOR CFRM 296
TO MAKE CFRM17 POLICY ACTIVE.
1 POLICY CHANGE(S) PENDING.

SETXCF START,REBUILD,STRNAME=ISGLOCK

IXC367I THE SETXCF START REBUILD REQUEST FOR STRUCTURE 298
ISGLOCK WAS ACCEPTED.
*ISG323A GLOBAL RESOURCE SERIALIZATION STOPPED ON SC52. 339
 LOCK STRUCTURE (ISGLOCK) REBUILD IS DUE TO OPERATOR REQUEST.
*ISG323A GLOBAL RESOURCE SERIALIZATION STOPPED ON SC42. 503
 LOCK STRUCTURE (ISGLOCK) REBUILD IS DUE TO OPERATOR REQUEST.
*ISG323A GLOBAL RESOURCE SERIALIZATION STOPPED ON SC48. 489
 LOCK STRUCTURE (ISGLOCK) REBUILD IS DUE TO OPERATOR REQUEST.
*ISG323A GLOBAL RESOURCE SERIALIZATION STOPPED ON SC43. 608
 LOCK STRUCTURE (ISGLOCK) REBUILD IS DUE TO OPERATOR REQUEST.
*ISG323A GLOBAL RESOURCE SERIALIZATION STOPPED ON SC49. 108
 LOCK STRUCTURE (ISGLOCK) REBUILD IS DUE TO OPERATOR REQUEST.
*ISG323A GLOBAL RESOURCE SERIALIZATION STOPPED ON SC54. 142
 LOCK STRUCTURE (ISGLOCK) REBUILD IS DUE TO OPERATOR REQUEST.
*ISG323A GLOBAL RESOURCE SERIALIZATION STOPPED ON SC53. 124
 LOCK STRUCTURE (ISGLOCK) REBUILD IS DUE TO OPERATOR REQUEST.
*ISG323A GLOBAL RESOURCE SERIALIZATION STOPPED ON SC47. 299
 LOCK STRUCTURE (ISGLOCK) REBUILD IS DUE TO OPERATOR REQUEST.
*ISG323A GLOBAL RESOURCE SERIALIZATION STOPPED ON SC50. 415
 LOCK STRUCTURE (ISGLOCK) REBUILD IS DUE TO OPERATOR REQUEST.
*ISG323A GLOBAL RESOURCE SERIALIZATION STOPPED ON SC55. 166
 LOCK STRUCTURE (ISGLOCK) REBUILD IS DUE TO OPERATOR REQUEST.
 IXC526I STRUCTURE ISGLOCK IS REBUILDING FROM 125
 COUPLING FACILITY CF01 TO COUPLING FACILITY CF01.
 REBUILD START REASON: OPERATOR INITIATED
 INFO108: 0000004D 0000004D.
 IXL014I IXLCONN REQUEST FOR STRUCTURE ISGLOCK WAS SUCCESSFUL.
 JOBNAME: GRS ASID: 0007 CONNECTOR NAME: SC53
 CFNAME: CF01
 IXL015I STRUCTURE ALLOCATION INFORMATION FOR 127
 STRUCTURE ISGLOCK, CONNECTOR NAME SC53
  CFNAME     ALLOCATION STATUS/FAILURE REASON
  --------   --------------------------------
  CF01       STRUCTURE ALLOCATED
  CF02       PREFERRED CF ALREADY SELECTED

                                                ---> next picture
```

```
ISG330I NEW GRS LOCK STRUCTURE, ISGLOCK, CONTAINS 2097152 LOCKS. 128
ORIGINAL GRS LOCK STRUCTURE CONTAINED 1048576 LOCKS.
IXL014I IXLCONN REQUEST FOR STRUCTURE ISGLOCK WAS SUCCESSFUL. 609
JOBNAME: GRS ASID: 0007 CONNECTOR NAME: SC43
CFNAME: CF01
ISG330I NEW GRS LOCK STRUCTURE, ISGLOCK, CONTAINS 2097152 LOCKS. 610
ORIGINAL GRS LOCK STRUCTURE CONTAINED 1048576 LOCKS.
IXL014I IXLCONN REQUEST FOR STRUCTURE ISGLOCK WAS SUCCESSFUL. 490
JOBNAME: GRS ASID: 0007 CONNECTOR NAME: SC48
CFNAME: CF01
ISG330I NEW GRS LOCK STRUCTURE, ISGLOCK, CONTAINS 2097152 LOCKS. 491
ORIGINAL GRS LOCK STRUCTURE CONTAINED 1048576 LOCKS.
IXL014I IXLCONN REQUEST FOR STRUCTURE ISGLOCK WAS SUCCESSFUL. 300
JOBNAME: GRS ASID: 0007 CONNECTOR NAME: SC47
CFNAME: CF01
ISG330I NEW GRS LOCK STRUCTURE, ISGLOCK, CONTAINS 2097152 LOCKS. 301
ORIGINAL GRS LOCK STRUCTURE CONTAINED 1048576 LOCKS.
IXL014I IXLCONN REQUEST FOR STRUCTURE ISGLOCK WAS SUCCESSFUL. 143
JOBNAME: GRS ASID: 0007 CONNECTOR NAME: SC54
CFNAME: CF01
ISG330I NEW GRS LOCK STRUCTURE, ISGLOCK, CONTAINS 2097152 LOCKS. 144
ORIGINAL GRS LOCK STRUCTURE CONTAINED 1048576 LOCKS.
IXL014I IXLCONN REQUEST FOR STRUCTURE ISGLOCK WAS SUCCESSFUL. 504
JOBNAME: GRS ASID: 0007 CONNECTOR NAME: SC42
CFNAME: CF01
ISG330I NEW GRS LOCK STRUCTURE, ISGLOCK, CONTAINS 2097152 LOCKS. 505
ORIGINAL GRS LOCK STRUCTURE CONTAINED 1048576 LOCKS.
IXL014I IXLCONN REQUEST FOR STRUCTURE ISGLOCK WAS SUCCESSFUL. 167
JOBNAME: GRS ASID: 0007 CONNECTOR NAME: SC55
CFNAME: CF01
ISG330I NEW GRS LOCK STRUCTURE, ISGLOCK, CONTAINS 2097152 LOCKS. 168
ORIGINAL GRS LOCK STRUCTURE CONTAINED 1048576 LOCKS.
IXL014I IXLCONN REQUEST FOR STRUCTURE ISGLOCK WAS SUCCESSFUL. 340
JOBNAME: GRS ASID: 0007 CONNECTOR NAME: SC52
CFNAME: CF01
ISG330I NEW GRS LOCK STRUCTURE, ISGLOCK, CONTAINS 2097152 LOCKS. 341
ORIGINAL GRS LOCK STRUCTURE CONTAINED 1048576 LOCKS.
IXL014I IXLCONN REQUEST FOR STRUCTURE ISGLOCK WAS SUCCESSFUL. 416
JOBNAME: GRS ASID: 0007 CONNECTOR NAME: SC50
CFNAME: CF01
IXL014I IXLCONN REQUEST FOR STRUCTURE ISGLOCK WAS SUCCESSFUL. 109
JOBNAME: GRS ASID: 0007 CONNECTOR NAME: SC49
CFNAME: CF01
ISG330I NEW GRS LOCK STRUCTURE, ISGLOCK, CONTAINS 2097152 LOCKS. 110
ORIGINAL GRS LOCK STRUCTURE CONTAINED 1048576 LOCKS.
ISG330I NEW GRS LOCK STRUCTURE, ISGLOCK, CONTAINS 2097152 LOCKS. 41
ORIGINAL GRS LOCK STRUCTURE CONTAINED 1048576 LOCKS.
ISG325I GRS LOCK STRUCTURE (ISGLOCK) REBUILD HAS COMPLETED ON SC42.
ISG325I GRS LOCK STRUCTURE (ISGLOCK) REBUILD HAS COMPLETED ON SC53.
ISG325I GRS LOCK STRUCTURE (ISGLOCK) REBUILD HAS COMPLETED ON SC48.
ISG325I GRS LOCK STRUCTURE (ISGLOCK) REBUILD HAS COMPLETED ON SC47.
ISG325I GRS LOCK STRUCTURE (ISGLOCK) REBUILD HAS COMPLETED ON SC43.
ISG325I GRS LOCK STRUCTURE (ISGLOCK) REBUILD HAS COMPLETED ON SC52.
ISG325I GRS LOCK STRUCTURE (ISGLOCK) REBUILD HAS COMPLETED ON SC54.
ISG325I GRS LOCK STRUCTURE (ISGLOCK) REBUILD HAS COMPLETED ON SC55.
ISG325I GRS LOCK STRUCTURE (ISGLOCK) REBUILD HAS COMPLETED ON SC50.
IXC521I REBUILD FOR STRUCTURE ISGLOCK HAS BEEN COMPLETED
ISG325I GRS LOCK STRUCTURE (ISGLOCK) REBUILD HAS COMPLETED ON SC49.
IXC513I COMPLETED POLICY CHANGE FOR CFRM. 113
CFRM17 POLICY IS ACTIVE.
```

The following is the display of the structure after REBUILD.

```
DISPLAY XCF,CF
IXC361I  07.30.34  DISPLAY XCF 115
  CFNAME     COUPLING FACILITY
  CF01       009672.IBM.02.000000040104
             PARTITION: 1   CPCID: 00
  CF02       009672.IBM.02.000000040104
             PARTITION: 1   CPCID: 01
D XCF,STR,STRNAME=ISGLOCK
IXC360I  07.30.48  DISPLAY XCF 304
STRNAME: ISGLOCK
 STATUS: ALLOCATED
  POLICY SIZE    : 20000 K
  POLICY INITSIZE: N/A
  REBUILD PERCENT: 1
  PREFERENCE LIST: CF01     CF02
  EXCLUSION LIST IS EMPTY

 ACTIVE STRUCTURE
----------------
 ALLOCATION TIME: 08/12/96 07:30:25
 CFNAME         : CF01
 COUPLING FACILITY: 009672.IBM.02.000000040104
                    PARTITION: 1   CPCID: 00
 ACTUAL SIZE    : 20224 K
 STORAGE INCREMENT SIZE: 256 K
 VERSION        : AD4E6F76 CB2A9004
 DISPOSITION    : DELETE
 ACCESS TIME    : 0
 MAX CONNECTIONS: 32
 # CONNECTIONS  : 10

CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
---------------- -- -------- -------- -------- ---- ----------------
SC42             01 00010008 SC42     GRS      0007 ACTIVE
SC43             03 00030003 SC43     GRS      0007 ACTIVE
SC47             0A 000A0004 SC47     GRS      0007 ACTIVE
SC48             07 00070003 SC48     GRS      0007 ACTIVE
SC49             09 00090003 SC49     GRS      0007 ACTIVE
SC50             06 00060004 SC50     GRS      0007 ACTIVE
SC52             02 00020005 SC52     GRS      0007 ACTIVE
SC53             05 0005000D SC53     GRS      0007 ACTIVE
SC54             08 00080003 SC54     GRS      0007 ACTIVE
SC55             04 00040003 SC55     GRS      0007 ACTIVE
```

See *OS/390 MVS System Commands* for more information about the preceding commands.

The actions required by global resource serialization to change its processing configuration and options, and to add or remove an MVS image to the sysplex, are summarized in the following table:

Table 5. Actions Required to Change GRS Processing Options and Add or Remove MVS Images

Processing Options or MVS Image	Action
Switch from Ring to Star	Dynamic - SETGRS MODE=STAR
Switch from Ring to Star	Static - Sysplex-wide IPL
Switch from Star to Ring	Sysplex-wide IPL
GRSRNL=EXCLUDE | xx	Sysplex-wide IPL
Change RNLs	Dynamic - SET GRSRNL=xx (*)
Change exit ISGGREX0	Sysplex-wide IPL
Remove an MVS from Complex	Dynamic - VARY XCF,sys,OFFLINE
Add an MVS to Complex	Dynamic - IPL MVS
Change ISGLOCK structure	Dynamic - SETXCF START,REBUILD
Note: (*) Generic change for qnames SYSDSN, SYSVSAM may require a sysplex-wide IPL.	

6.6.1 Possible Errors

Because global resource serialization now supports two types of complexes, star or ring, it is possible that due to an error in the specification of the GRS= parameter in IEASYSxx, the installation could accidentally try to IPL a system into the wrong type of complex. The worst result would be the creation of two GRS complexes.

The possible error cases and the action taken by global resource serialization in each case are as follows:

- Error detected
  - If an OS/390 Release 2 system is IPLed with GRS=STAR specified and a ring complex already exists in the sysplex, global resource serialization issues an error message and places the IPLing system in a X′0A3′ wait state with X′84′ reason code.
  - If an OS/390 Release 2 system is IPLed with one of the ring-related parameters (GRS=START, JOIN, or TRYJOIN) and a star complex already exists in the sysplex, global resource serialization issues an error message and places the IPLing system in a X′0A3′ wait state with a X′84′ reason code.
  - If a down-level MVS system with XCF support (that is, at least a MVS/ESA Version 4 system) is IPLed with one ring-related parameters (GRS=START, JOIN, or TRYJOIN), and a star complex already exists, global resource serialization issues an error message and places the IPLing system in a X′0A3′ wait state with a X′84′ reason code.
- Error undetected
  - If a pre-XCF MVS system (at least prior to MVS/ESA Version 4) is IPLed with a ring-related parameter (START, JOIN, or TRYJOIN) and a star complex already exists, global resource serialization is not able to detect this type of error, and two separate global resource serialization complexes are created. This does not serialize resources across the two sets of systems correctly.
  - If the installation is running a ring complex where all systems are interconnected using CTC support rather than XCF communication, and GRS=STAR is specified when IPLing a system that should be part of the ring, GRS is not able to detect this type of error. As in the preceding condition, two separate GRS complexes are created.

Two suggestions to prevent the previously mentioned potential errors are:

- Remove the CTC definitions from the GRSCNFxx member used for systems that should be participating to a star complex.
- If the processing has to remain for a ring complex, do not add the global resource serialization record to the sysplex couple data set.

6.7 GRS Lock Structure Rebuild

In a global resource serialization star configuration, the coupling facility lock structure is the hub for global requests processing. Global resource serialization attempts rebuild processing in response to a loss of connectivity to the coupling facility, a coupling facility structure failure, or an operator request.

In case of connectivity error, MVS determines, depending on the CFRM policy (REBUILDPERCENT) and the SFM policy for the system weights, whether a rebuild is to be initiated:

- If yes, global resource serialization continues with rebuild.
- If not, the rebuild is stopped and global resource serialization attempts to return and use the original lock structure. Any systems without connectivity to the original lock structure cannot continue, and are put in non-restartable wait state code X′0A3′ and reason code X′A8′.

For this reason, it is recommended you implement CFRM and SFM policies to allow the structure rebuild based upon the amount of capacity loss an installation can accept in case of a connectivity failure.

Note: If you want to have the highest GRS availability, use REBULDPERCENT(1) in the CFRM policy for the ISGLOCK structure. This guarantees that if any system loses connectivity to the structure, the structure rebuild is always attempted in a different coupling facility. For more information about CFRM and SFM policies and XES rebuild of a structure, see 6.5, “GRS Star Implementation and Migration” on page 75 and 7.9, “Migration and Coexistence Considerations” on page 140.

During the rebuild processing, global resource serialization on each system is notified, and the global resource serialization requesters are suspended. If the rebuild completes successfully, message ISG325I is issued and normal global resource serialization process is resumed.

If the global resource serialization lock structure fails at IPL during system initialization, system initialization ends with X′0A3′ wait state and reason code X′A4′.

If the connectivity to global resource serialization lock structure fails at IPL during system initialization, system initialization ends with X′0A3′ wait state and reason code X′A8′.

It is also possible to rebuild the lock structure with the operator command SETXCF START,REBUILD. This can be used to move a structure to a different coupling facility or to enlarge the structure. For additional information, see 6.6, “GRS Star Operation” on page 82.

Table 6 on page 90 summarizes the possible results during the rebuild of the global resource serialization lock structure.

Table 6. Rebuild the GRS Lock Structure		
Old Structure	**New Structure**	**Result**
Connectivity failure	OK	Rebuild successful.
Structure failure	OK	Rebuild successful.
Connectivity Failure	System or CF configuration does not meet policy requirements to initiate or complete a rebuild (see note).	The system is partitioned from the sysplex and put in wait state X′0A3′.
Connectivity Failure	Operator stops rebuild	Systems that retain connectivity to the old structure return to it. Other systems are partitioned from the sysplex and put in a wait state X′0A3′.
Connectivity failure	Connectivity failure	CFFP decides. See 7.9, “Migration and Coexistence Considerations” on page 140.
Connectivity failure	Structure failure	Rebuild is restarted.
Structure failure	Operator stops rebuild	Rebuild is restarted.
Structure failure	Connectivity failure	Rebuild is restarted.
Structure Failure	Structure failure	Rebuild is restarted.
Note: GRS depends on the decisions made by XES based upon CFRM and SFM policies.		

6.8 GRS Star Component Trace

Global resource serialization component tracing (CTRACE) can be used to trace events pertaining to serializing requests for global resources, and has been extended to include events pertaining to the star processing environment.

Of the event trace options that can be specified in a CTnGRSxx parmlib member or in response to a prompt from the TRACE command, you can use the following ones in a star complex:

- CONTROL group to trace unusual events and events related to the establishment, termination, and modification of the control structure needed for processing. This includes:
  - Dynamic RNL changes
  - Error events
  - XCF services used when setting up for processing
- REQUEST group to trace the path of requests.
- MONITOR group to trace XCF/XES exit events.
- SIGNAL group to trace XCF events pertaining to the message exit and signaling services.
- FLOW group, which traces the flow of control from one entry point to another.

Additions to the global resource serialization component trace for the star complex are:

- FLOW option
- The introduction of sub-options, which allow you to refine the set of events that are to be traced for each major entry point

- The definition of a set of minimum options (MINOPS) that can be traced at all times:
  - For GRS=NONE, the minimum trace option is CONTROL.
  - For GRS=STAR, the minimum trace options are CONTROL1, CONTROL2, and SIGNAL0.
- The ability to filter trace entries by major option when viewing the trace using the IPCS CTRACE subcommand
- Enhanced IPCS CTRACE formatting

For additional information, see *OS/390 MVS Diagnosis Reference*, LY28-1084 (available to IBM licensed customers only).

6.9 GRS Measurements and Tuning

To analyze the behavior of a GRS star complex, a comparison has been made with a GRS ring complex. The sysplex configuration has ten MVS systems in the complex running on 9021 H5 (two partitions), 9672-R72 (two partitions), and 9672-E0x (six partitions). The GRS complex matches the sysplex. The sysplex is lightly loaded, but comparable data can still be used. The objectives are to compare GRS ring and GRS star configuration for:

- Processor and storage resources used by GRS in the sysplex
- Response time for the most used resources serialized with RESERVE macros
- Behavior of RESERVEs excluded (hardware reserve + system) and converted (no-hardware reserve + systems)
- Enqueue delay time

6.9.1 GRS Ring Measurements and Considerations

This section describes the measurements made in a GRS ring configuration.

6.9.1.1 CPU and Processor Storage Utilization

The overall picture of the processor and storage resources utilized by GRS in the ring configuration is shown in Figure 59 on page 92. The reporting MVS system is SC47, which runs in a 9021 H5 partition. The values shown are related to an elapsed time of 127 hours since a sysplex-wide IPL.

```
SYSPLEX  IPLed  since  127 hours  or 457.200 seconds

-------------------------------------------------------------------------------
SYSTEMs  NAMES

SC42   SC43   SC47   SC48   SC49   SC50   SC52   SC53   SC54   SC55
-------------------------------------------------------------------------------

                      GRS   CPU     UTILIZATION in seconds

7170   2040   1330   2240  11600  10000  12500    565   3550  13400  --> 64395 sec.

                        64395 cpu-sec. / 457200 elapsed sec. = 14.08% of 1 cp
                                                                  1
-------------------------------------------------------------------------------

                      GRS   STORAGE UTILIZATION in frames

 241    602    263    230    442    602    293    388    597    446  --> 41MB
                                                                        2
-------------------------------------------------------------------------------


OUTSTANDING  GLOBAL  REQUESTS:

                    SYSDSN  2242       (included)
                    SYSVSAM  452
                    SPFEDIT  160
                             ------
                     TOTAL  2854

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

From SC47 ----->   79577  (18.3%) global enq/deq  in 114.000 sec. -->  0.7 sec.
                  355072  (81.7%) local  enq/deq  in 114.000 sec. -->  2.5 sec.

Note: the above numbers do not include the STEP requests and the
      local request for SYSZTIOT, SYSZJVTP, and temporary datasets.
```

Figure 59. GRS Ring Configuration Resources Utilization

6.9.1.2 Enqueue Delay

The following is a short description of the reasons for the enqueue delay in a GRS ring configuration. The enqueue delay is affected by the “time slicing” for the RSA token, which is passed sequentially around the ring. The tuning parameters available for GRS ring are in the GRSCNFxx parmlib member; they are:

- ACCELSYS(num), which is the number of systems that must see the global request before granting. The use of ACCELSYS requires a fully connected sysplex. The maximum acceleration is achieved with a value of 2, the default value is 99.
- RESMIL(num|off), which specifies the minimum RSA residency time in milliseconds in the system. RESMIL(off) indicates that GRS is not to attempt to tune the RSA residency time on the system, while with RESMIL(0) GRS tunes the residency time with a minimum of zero. GRS tuning means that if there is no global enqueue activity in the system, the RESMIL value is increased by one millisecond up to five times. The default RESMIL value is ten milliseconds.

In the sysplex configuration, GRS is using XCF for ring communication, and XCF is using ESCON CTC connections for performance reasons. The GRSCNFxx parmlib member specifies ACCELSYS=2 and RESMIL=0.

The enqueue delay is considered the time between the issue of a global request to GRS and the granting of the request, in a condition where there is no contention for the requested resource.

The enqueue delay in the ten MVS systems sysplex averages 30 milliseconds. It has been observed with the ENQ/DEQ monitor (see Appendix B, "ENQ/DEQ/RESERVE Analysis Aid Reports" on page 191).

An enqueue delay evaluation has also been made using the ring cycle time determined with the last RSA number, the number of seconds last since sysplex IPL, and the number of systems in the ring. The RSA number is incremented by one each time it is transmitted from a system to the next in the ring; therefore, it is incremented by ten for each complete ring circulation when ten is the number of systems in the sysplex. The last RSA number was obtained with a modification to the monitor. Figure 60 shows the logic used to evaluate the enqueue delay.

```
1000 / (last RSA # / ( # secs. since IPL * #sys)) = RING TIME msec.

1000 / ( 108.356.400 / ( 457.200 * 10 )) = 1000 / 23.7 = 42.19 msec.

This means that the average time for the RSA to go from a system
to the next one is 42.19 / 10 = 4.219 msec.

The ENQ delay is the RSA latency (cycle time/2) plus the time
for RSA to go through two systems being ACCELSYS=2.

    ENQ DELAY =  (42.19 / 2 + (42.19 / 10) * 2) = 29.533 msec.

which is closed to 30 msec. measured with the monitor
```

Figure 60. GRS Enqueue Delay Evaluation Using Last RSA Number

Figure 61 indicates the enqueue delay values that can be experienced in sysplex configurations of three and four systems, which are more common in the field. The value related to the test environment is also reported.

```
          sysplex matches GRS complex
          GRS uses XCF for communication
          XCF  uses CTCs for communication


          ACCELSYS=2  RESMIL=1
          3 systems sysplex --> 10 - 13 msec.
          4 systems sysplex --> 15 - 18 msec.


          ACCELSYS=2  RESMIL=0
          10 systems sysplex --> 30 - 35 msec.
```

Figure 61. GRS Ring Complex - Enqueue Delay Values

In the test environment, the enqueue delay value is not constant; it fluctuates It can be lower than 30 milliseconds, but it has also been measured above 200

milliseconds for 200 times in 140000 seconds (39 hours) of elapsed time. Figure 62 on page 94 shows the enqueue delays over 200 milliseconds that have been measured between 8:30 and 9:30.

```
              ENQ/DEQ Monitor - Minor Name List

                           Major Name  : ENQDELAY
                           RNL . . . . :
                           Scope . . . : SYSTEM

  Minor Name-------------------------------------------  --------
  mic-dly   date   time                                  Counter:
  00418382 96.218 08:21:58                                      1
  00200362 96.218 08:28:14                                      1
  00383997 96.218 08:28:51                                      1
  00376175 96.218 08:30:56                                      1
  39608595 96.218 08:30:56                                      1
  00589608 96.218 08:31:14                                      1
  00548659 96.218 08:32:36                                      1
  00268045 96.218 08:33:17                                      1
  00254504 96.218 08:33:48                                      1
  00362709 96.218 08:34:49                                      1
  00391093 96.218 08:36:32                                      1
  00312764 96.218 08:39:06                                      1
  00431770 96.218 08:45:47                                      1
  00257867 96.218 08:50:52                                      1
  00408538 96.218 08:52:45                                      1
  00348911 96.218 09:05:23                                      1
  00271839 96.218 09:10:00                                      1
  00353656 96.218 09:15:27                                      1
  00499690 96.218 09:15:38                                      1
  00240034 96.218 09:19:22                                      1
  00275200 96.218 09:19:54                                      1
  00309200 96.218 09:20:00                                      1
  00334960 96.218 09:21:46                                      1
  00289219 96.218 09:22:06                                      1
  00456638 96.218 09:25:42                                      1
```

Figure 62. GRS Ring Complex - Enqueue Delay Trace

6.9.1.3 RESERVE Requests Behavior

All global requests are delayed on average by 30 milliseconds. This can slow down processing for system services like catalog, DASD space management, RACF, VSAM open, and allocation (if SYSDSN is in the RNL inclusion list). The following analysis is related to the RESERVE time, which is another critical performance issue in a sysplex complex. The RESERVE time is considered the time between the RESERVE issued by a system component, like catalog or DADSM, and the DEQ that releases the resource.

A RESERVE results in two global serialization requests: a hardware reserve that serializes access to the DASD volume, and a SYSTEMS enqueue (the default) that is propagated along the ring to all systems in the sysplex. Only one serialization request is required; double serialization causes overhead and potential interlock situations. RESERVEs can be converted and processed with scope SYSTEMS without the hardware reserve, or excluded and processed with scope SYSTEM with the hardware reserve. This is done with the parmlib member GRSRNLxx, which contains the RNL tables. For additional information about the RNL processing, see 6.10, “Resource Name List Processing” on page 104, and 6.10.1, “RESERVE Conversion” on page 107.

The report used has been obtained with the ENQ/DEQ monitor that indicates, along with the reserve time, if a RESERVE has been converted, excluded, or left as it was.

Figure 63 shows the RESERVEs issued by SC47 (9021 H5 partition) in 140000 seconds (39 hours) with the average response times of those RESERVEs:

```
              ENQ/DEQ Monitor - Major Name List

                                          Elapsed seconds:  114000

----------   ---------   --------   -------  -average-   -Reserved-
Major Name      RNL       Scope     Counter     msec        seconds
SYSIGGV2     CONVERTED   *SYSTEMS     25561       51  A       1340
IGDCDSXS     CONVERTED   *SYSTEMS     11525      108          1275
SYSZRACF     CONVERTED   *SYSTEMS      4657       52  B        359
SPFEDIT      CONVERTED   *SYSTEMS      3015       54           165
ARCGPA       CONVERTED   *SYSTEMS       544       77            42
SYSIEWLP     CONVERTED   *SYSTEMS        11     1647            18
SYSZJES2     EXCLUDED    *RESERVE     98252      143         14338
SYSZVVDS     EXCLUDED    *RESERVE     49248        6  C        296
SYSVTOC      EXCLUDED    *RESERVE      1266       34  D         43
SYSIGGV2     EXCLUDED    *RESERVE       203        3             0
SYSIGGV2     FORCED       RESERVE         8       76             0
```

Figure 63. GRS Ring Complex - RESERVE Response Time

The average reserve times for the converted RESERVEs are elongated by 30 milliseconds, while the excluded RESERVEs are not penalized by the enqueue delay.

The converted catalog reserve SYSIGGV2 **A** has an average time of 51 milliseconds, and the converted RACF reserve SYSZRACF **B** has an average time of 52 milliseconds; both reserve times include 30 milliseconds of enqueue delay, which is around 60% of the total time. These RESERVEs influence the response time for catalog and RACF requests, and are substantially penalized by the conversion processing. This is an important point to consider when comparing the data with a GRS star complex.

The excluded catalog reserve SYSZVVDS **C** has an average time of 6 milliseconds, while the DASDM reserve SYSVTOC **D** has an average time of 34 milliseconds. These reserves are excluded; GRS gives them scope SYSTEM and, therefore, the reserves are not propagated along the ring. Instead, response time depends on the application logic of the system component, on the DASD device activity, and on the DASD device contention caused by the hardware reserve.

6.9.2 GRS Star Measurements and Considerations

With the introduction of the GRS star support, GRS uses the contention detection and management capability of the XES lock structure to determine and assign ownership of a particular global resource; RSA passing is no longer used. Each system maintains only a local copy of its own global resources, and the GRS coupling facility lock structure has the overall image of all system global resources in use.

The GRS processing has been switched to star mode using the SETGRS MODE=STAR global command. The switch was done in a period of low activity and it only took only a couple of minutes. The active RNLs were not changed,

nor was the hardware configuration; therefore, the sysplex complex is the same as for the GRS ring configuration.

6.9.2.1 CPU and Processor Storage Utilization

The measurements data for the GRS star configuration have been taken from the MVS system SC47, which runs in a 9021 H5 partition. They are related to an elapsed time of 10 hours; for the GRS ring, the measurements data were related to an elapsed time of 127 hours. There is no contention for the coupling facility containing the ISGLOCK structure. See Figure 72 on page 102 for the coupling facility RMF output.

The overall picture of the processor and storage resources utilized by GRS in star configuration is shown in Figure 64.

```
 SYSPLEX  IPLed  since  10 hours  or 36.000 seconds

-------------------------------------------------------------------------------
 SYSTEMs  NAMES

 SC42   SC43   SC47   SC48   SC49   SC50   SC52   SC53   SC54   SC55
-------------------------------------------------------------------------------

                       GRS   CPU     UTILIZATION in second

  134     85    144     34    131    130    157    124    128    140  --> 1207 sec.

                          1207 cpu-sec. / 36000 elapsed sec. =  3.35% of 1 cp
                                                                 3
-------------------------------------------------------------------------------

                       GRS   STORAGE UTILIZATION in frames

  710    988    123    312    800    789    890    970    826    983  --> 74MB
                                                                        4
-------------------------------------------------------------------------------


 OUTSTANDING  GLOBAL  REQUESTS:  1600


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

 From SC47 ----->   25840  (23.4%) global enq/deq  in   36000 sec. -->  0.7 sec.
                    84560  (76.6%) local  enq/deq  in   36000 sec. -->  2.6 sec.

 Note: the above numbers do not include the STEP requests and the
       local request for SYSZTIOT, SYSZJVTP, and temporary datasets.
```

Figure 64. GRS Star Configuration Resources Utilization

The total GRS processor storage utilization (ten systems) is 74MB **4** (Figure 64) in a star configuration with 1600 global outstanding requests. In comparison, it was 42MB **2** (Figure 59 on page 92) in a ring configuration with 2854 global outstanding requests. This seems to be in contrast with the expected decrease of processor storage utilization, but it may be related to the different size of the GRS control blocks structure and to the low number of global outstanding requests. In a production environment, the number of global outstanding requests is much higher, and a GRS star configuration is expected to use less processor storage than a GRS ring configuration.

The total GRS processor utilization (ten systems) is 3.35% of one CP 3 (Figure 64) in the star configuration, while it was 14.08% of one CP 1 (Figure 59 on page 92) in the ring configuration. In both measurements, the ENQ/DEQ load is low and comparable; the difference is mainly due to the RSA passing overhead of the ring configuration.

6.9.2.2 Enqueue Delay

Because GRS star uses the contention detection and management capability of the XES lock structure to process global requests, the enqueue delay for global requests is now related to the processor speed, to the coupling facility contention, and to the response time for synchronous requests. The enqueue delay is the elapsed time between the issue of a global request to GRS and the granting of the request, assuming that there is no contention for the requested resource.

Figure 65 summarizes the enqueue delays that have been measured on three different processor types. It is noted that the ten-system sysplex is lightly loaded, therefore the enqueue delay values shown may not be representative of a production sysplex. The values can be compared with GRS ring configuration, because it was running the same type of workload.

```
ENQUEUE DELAY STAR
                     9021  H5     0.5 - 0.7  msec.

                     9672  RX3    1.0 - 1.5  msec.

                     9672  E02    2.0 - 3.0  msec.

ENQUEUE DELAY RING CONFIGURATION  30 - 35  msec.
```

Figure 65. GRS Star Configuration - Enqueue Delay

To check the fluctuation of the enqueue delay, the ENQ/DEQ monitor threshold value has been set to 5 milliseconds in a star configuration; while in a ring configuration, it was set to 200 milliseconds. Figure 66 on page 98 shows some of the 42 ENQDELAYs over 5 milliseconds measured between 8:00 and 18:00 on SC47.

```
              ENQ/DEQ Monitor - Minor Name List

                           Major Name  : ENQDELAY
                           RNL . . . . :
                           Scope . . . : SYSTEM

 Minor Name------------------------------------------  --------
 mic-dly   date   time                                 Counter:
 00217767 96.221 08:42:29                                    1
 00174831 96.221 14:34:40                                    1
 00143401 96.221 14:52:39                                    1
 00137579 96.221 09:00:00                                    1
 00097328 96.221 10:47:35                                    1
 00040811 96.221 15:58:05                                    1
 00036067 96.221 12:46:01                                    1
 00033464 96.221 14:19:38                                    1
 00027253 96.221 11:52:19                                    1
 00026936 96.221 12:49:34                                    1
 00025074 96.221 14:14:54                                    1
 00022881 96.221 11:44:24                                    1
 00020886 96.221 15:51:37                                    1
 00018362 96.221 08:24:26                                    1
 00017812 96.221 12:06:38                                    1
 00016995 96.221 13:56:23                                    1
   //       //     //
 00009239 96.221 17:19:21                                    1
 00009188 96.221 12:37:27                                    1
 00009010 96.221 17:12:58                                    1
 00008941 96.221 13:45:14                                    1
 00008882 96.221 12:23:10                                    1
   //       //     //
 00006105 96.221 13:46:58                                    1
 00005940 96.221 11:37:21                                    1
 00005782 96.221 12:31:44                                    1
 00005724 96.221 11:25:18                                    1
 00005681 96.221 12:28:23                                    1
 00005648 96.221 09:43:00                                    1
 00005615 96.221 17:37:24                                    1
 00005530 96.221 09:36:55                                    1
 00005301 96.221 08:34:10                                    1
 00005165 96.221 08:37:20                                    1
 00005157 96.221 17:41:27                                    1
```

Figure 66. GRS Star Configuration - Enqueue Delay Trace

It is noted that the highest enqueue delays happen when the ISGLOCK structure is rebuilt, and when a system is partitioned from the sysplex. For ISGLOCK structure rebuild, the enqueue delay matches the time taken by the system to rebuild the structure. (In the test configuration, it takes about ten seconds to rebuild.) The enqueue delay when a system is partitioned from the test sysplex varies between two and six seconds. When a system joins the sysplex, the measured delay has not exceeded five milliseconds. Figure 67 on page 99 shows the enqueue delays measured during ISGLOCK rebuild and systems partitioning from the sysplex.

```
              ENQ/DEQ Monitor - Minor Name List

                          Major Name  : ENQDELAY
                          RNL . . . . :
                          Scope . . . : SYSTEM

Minor Name-------------------------------------------   --------
sec-dly    date   time                                  Counter:

10.641616 96.223 12:39:55   <-- during a STRUCTURE REBUILD    1

05.420486 96.223 11:34:52   <-- during a system partitioning  1

02.083256 96.223 11:16:30   <-- during a system partitioning  1
```

Figure 67. GRS Star - Enqueue Delay

6.9.2.3 RESERVE Requests Behavior

Figure 68 shows the RESERVEs issued by SC47 in 33496 seconds (9.3 hours) with the average time of the RESERVEs while running in GRS star mode.

```
              ENQ/DEQ Monitor - Major Name List

                                          Elapsed seconds:   33496

----------   ---------   --------   -------  -average-   -Reserved-
Major Name      RNL       Scope     Counter     msec        seconds
SYSIGGV2     CONVERTED   *SYSTEMS     39353        6  A       236
IGDCDSXS     CONVERTED   *SYSTEMS      3338       30           103
SYSIEWLP     CONVERTED   *SYSTEMS       153      291            44
SYSZRACF     CONVERTED   *SYSTEMS      8300        4  B        36
SPFEDIT      CONVERTED   *SYSTEMS       702       47            33
ARCGPA       CONVERTED   *SYSTEMS       153       57             8
SYSZJES2     EXCLUDED    *RESERVE     31793       83          2674
SYSZVVDS     EXCLUDED    *RESERVE     57512        4  C       230
SYSVTOC      EXCLUDED    *RESERVE      1257       26  D        33
SYSIGGV2     EXCLUDED    *RESERVE       206        5             1
SYSIGGV2     FORCED       RESERVE         9       22             0
```

Figure 68. GRS Star - RESERVEs Response Time

The converted catalog reserve SYSIGGV2 **A** has an average time of 6 milliseconds (it was 51 milliseconds with GRS ring, Figure 63 on page 95). The converted RACF reserve SYSZRACF **B** has an average time of 4 milliseconds (it was 52 milliseconds with GRS ring, Figure 63 on page 95). Both reserves improved because the enqueue delay is now about one millisecond for SC47 (9021 H5), down from 30 milliseconds for the ring configuration. The time differences indicate that the DASD I/O response time improved as well.

The excluded catalog reserve SYSZVVDS **C** has an average time of 4 milliseconds (it was 6 milliseconds with GRS ring, Figure 63 on page 95), while the DASDM reserve SYSVTOC **D** has an average time of 26 milliseconds (it was 34 milliseconds with GRS ring, Figure 63 on page 95). These reserves are excluded by GRS, and as such are local requests, and are therefore not using the lock structure.

Because the enqueue delay ranges from 0.5 to 3 milliseconds (Figure 65 on page 97), the RNLs have been changed to convert SYSZVVDS and SYSVTOC.

The RESERVE conversion, if it is not slowing down a system service (in this case catalog and DADSM), is the preferred choice because DASD devices are not tied up by hardware reserves that may cause interlocks. Figure 69 on page 100 shows the average time of RESERVEs for SC47 (9021 H5) after the RNL change.

```
              ENQ/DEQ Monitor - Major Name List


                                              Elapsed seconds:   16914

 ----------   ---------   --------   -------  -average-    -Reserved-
 Major Name      RNL       Scope     Counter     msec         seconds
 SYSIGGV2     CONVERTED   *SYSTEMS     20369        3            81
 IGDCDSXS     CONVERTED   *SYSTEMS      1686       32            56
 SYSZVVDS     CONVERTED   *SYSTEMS     26125        3 C          78
 SYSZRACF     CONVERTED   *SYSTEMS      8584        4            37
 SPFEDIT      CONVERTED   *SYSTEMS       229       29             6
 ARCGPA       CONVERTED   *SYSTEMS        72       53             3
 SYSVTOC      CONVERTED   *SYSTEMS       128       26 D           3
 SYSIEWLP     CONVERTED   *SYSTEMS         7       92             0
 SYSZJES2     EXCLUDED    *RESERVE     16086       80          1328
 SYSIGGV2     EXCLUDED    *RESERVE        82        3             0
 SYSZVVDS     FORCED       RESERVE        93        2             0
```

Figure 69. GRS Star - RESERVEs Response Time

The converted RESERVEs for SYSZVVDS **C** and SYSVTOC **D** did not show a significant change in the response time when moved from the RNL exclusion to the RNL conversion list. See Table 8 on page 104 for average RESERVEs response time for 9021 H5, CMOS2, and CMOS1 types of processors.

Note: In an equivalent condition, it is suggested to move the qname for SYSVTOC from the RNL exclusion to the RNL conversion list. This does not slow down VTOC processing, and is beneficial for all applications, like DFDSS, that may request a VTOC reserve for a considerable amount of time. See 6.11.19, “VTOC” on page 124 and “VSAM Catalogs and ICF Catalogs” on page 121 for additional information.

6.9.2.4 ISGLOCK Structure

The ISGLOCK structure has been defined with a size of 10MB, and it was expected to have room for approximately 170K lock entries, but the number of entries have been 1 million. The formula used (see 6.5, “GRS Star Implementation and Migration” on page 75) evaluates storage for a number of lock entries that is six times the given number of global outstanding requests. This is to avoid false lock contention if the structure should run out of lock entries.

Figure 70 on page 101 shows the output of a DISPLAY GRS command, which gives the number of lock entries.

```
D GRS

 ISG343I 11.40.56 GRS STATUS 067
 SYSTEM    STATE               SYSTEM    STATE
 SC47      CONNECTED           SC48      CONNECTED
 SC42      CONNECTED           SC43      CONNECTED
 SC54      CONNECTED           SC50      CONNECTED
 SC53      CONNECTED           SC55      CONNECTED
 SC52      CONNECTED           SC49      CONNECTED
 GRS STAR MODE INFORMATION
    LOCK STRUCTURE (ISGLOCK) CONTAINS 1048576 LOCKS.
```

Figure 70. Display GRS Command

Figure 71 shows the output display for the allocated ISGLOCK structure.

```
D XCF,STRUCTURE,STRNAME=ISGLOCK

STRNAME: ISGLOCK
 STATUS: ALLOCATED
  POLICY SIZE    : 10000 K
  POLICY INITSIZE: 10000 K
  REBUILD PERCENT: 1
  PREFERENCE LIST: CF01     CF02
  EXCLUSION LIST IS EMPTY

 ACTIVE STRUCTURE
 ----------------
  ALLOCATION TIME: 08/09/96 13:54:09
  CFNAME         : CF02
  COUPLING FACILITY: 009672.IBM.02.000000040104
                     PARTITION: 1   CPCID: 01
  ACTUAL SIZE    : 10240 K
  STORAGE INCREMENT SIZE: 256 K
  VERSION        : AD4AFFA3 79ED8682
  DISPOSITION    : DELETE
  ACCESS TIME    : 0
  MAX CONNECTIONS: 32
  # CONNECTIONS  : 10

 CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
 ---------------- -- -------- -------- -------- ---- ----------------
 SC42             03 00030002 SC42     GRS      0007 ACTIVE
 SC43             04 00040002 SC43     GRS      0007 ACTIVE
 SC47             01 00010003 SC47     GRS      0007 ACTIVE
 SC48             02 00020002 SC48     GRS      0007 ACTIVE
 SC49             0A 000A0002 SC49     GRS      0007 ACTIVE
 SC50             06 00060003 SC50     GRS      0007 ACTIVE
 SC52             09 00090002 SC52     GRS      0007 ACTIVE
 SC53             07 00070002 SC53     GRS      0007 ACTIVE
 SC54             05 0005000A SC54     GRS      0007 ACTIVE
 SC55             08 00080002 SC55     GRS      0007 ACTIVE
```

Figure 71. ISGLOCK Structure Display

For a lock structure to be effective, it should have a low number of false contentions. This depends on the number of lock entries in the structure, and on the effectiveness of the hashing algorithm used by the owner of the GRS structure. Figure 72 on page 102 shows the output of an RMF coupling facility report for the sysplex.

```
                                                % OF              %le 50    AVG    LST/DIR  DATA      LOCK     DIR
      STRUCTURE                        ALLOC    CF          #     ALL       REQ/   ENTRIES  ELEMENTS  ENTRIES  RECLAIM
TYPE  NAME             STATUS CHG     SIZE     STORAGE     REQ   REQ       SEC    TOT/CUR  TOT/CUR   TOT/CUR
                                                                                    0        0         0
      ISGLOCK          ACTIVE         10M      2.0%    18671   12.5%     31.12      0        0       1049K      N/A
                                                                                    0        0       1472  I
-------------------------------------------------------------------------------------------------------------------------

 AVERAGE CF UTILIZATION  (% BUSY)           2.1      LOGICAL PROCESSORS:   DEFINED   6    EFFECTIVE   6.0
                                            K
-------------------------------------------------------------------------------------------------------------------------

 STRUCTURE NAME = ISGLOCK           TYPE = LOCK
          # REQ    -------------- REQUESTS --------------   -------------- DELAYED REQUESTS -------------
 SYSTEM   TOTAL              #     % OF  -SERV TIME(MIC)-    REASON    #   % OF  ---- AVG TIME(MIC) -----  EXTERNAL REQUEST
 NAME     AVG/SEC            REQ    ALL     AVG   STD_DEV            REQ   REQ   /DEL    STD_DEV   /ALL   CONTENTIONS

 SC42     11468    SYNC     11K  63.9%   176.5     75.4                                                    REQ TOTAL       12K
          19.11    ASYNC     0    0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED    65
                   CHNGD     0    0.0%  INCLUDED IN ASYNC                                                  -CONT           65
                                                                                                           -FALSE CONT     23

 SC43       323    SYNC    323   1.8%   180.9     87.8                                                    REQ TOTAL      322
            0.54   ASYNC     0    0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED    74
                   CHNGD     0    0.0%  INCLUDED IN ASYNC                                                  -CONT           74
                                                                                                           -FALSE CONT      9

 SC47      5183    SYNC   5183  28.9%   131.7      9.7                                                    REQ TOTAL     5174
           8.64    ASYNC     0    0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED    95
                   CHNGD     0    0.0%  INCLUDED IN ASYNC                                                  -CONT           95
                                                                                                           -FALSE CONT     46

 SC48       145    SYNC    145   0.8%   140.5      7.6                                                    REQ TOTAL      146
            0.24   ASYNC     0    0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED    20
                   CHNGD     0    0.0%  INCLUDED IN ASYNC                                                  -CONT           20
                                                                                                           -FALSE CONT      0

 SC49       174    SYNC    174   1.0%   313.8    451.8                                                    REQ TOTAL      146
            0.29   ASYNC     0    0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED    44
                   CHNGD     0    0.0%  INCLUDED IN ASYNC                                                  -CONT           44
                                                                                                           -FALSE CONT      1

 SC50       196    SYNC    196   1.1%   268.9    303.6                                                    REQ TOTAL      168
            0.33   ASYNC     0    0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED    40
                   CHNGD     0    0.0%  INCLUDED IN ASYNC                                                  -CONT           40
                                                                                                           -FALSE CONT      0

 SC52       727    SYNC    727   8.8%   231.7    141.4                                                    REQ TOTAL      168
            1.21   ASYNC     0    0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED    40
                   CHNGD     0    0.0%  INCLUDED IN ASYNC                                                  -CONT           40
                                                                                                           -FALSE CONT      0

 SC53       154    SYNC    154   0.9%   330.2    570.9                                                    REQ TOTAL      727
            0.26   ASYNC     0    0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED    28
                   CHNGD     0    0.0%  INCLUDED IN ASYNC                                                  -CONT           28
                                                                                                           -FALSE CONT     11

 SC54       148    SYNC    148   0.8%   278.9    259.0                                                    REQ TOTAL      146
            0.25   ASYNC     0    0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED    14
                   CHNGD     0    0.0%  INCLUDED IN ASYNC                                                  -CONT           14
                                                                                                           -FALSE CONT      0

 SC55       153    SYNC    153   0.9%   288.7    380.7                                                    REQ TOTAL      146
            0.25   ASYNC     0    0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED    25
                   CHNGD     0    0.0%  INCLUDED IN ASYNC                                                  -CONT           25
                                                                                                           -FALSE CONT      6

-------------------------------------------------------------------------------------------------------------------------
 TOTAL    18671    SYNC     19K  100%   168.8    111.9                                                    REQ TOTAL       19K
          31.12    ASYNC     0    0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED   431
                   CHNGD     0    0.0%                                                                    -CONT          431
                                                                                                           -FALSE CONT     98
             G                                                J                                           H
```

Figure 72. ISGLOCK Structure RMF

The coupling facility and the ISGLOCK structure do not have contention: the coupling facility utilization is 2.1% **K**, the number of requests is 31.12 per second **G**.

The coupling facility is an LPAR with six dedicated processors of a 9673-E03 (CMOS 1). Its response time for synchronous lock requests depends on the type of processor from which they are issued. The averages are the following:

```
        SC47 - SC48  (9021-942) (H5)          140 microseconds
        SC42 - SC43  (9672-R72) (CMOS 2)      180 microseconds
SC49 - SC52 - SC53  (9672-E01) (CMOS 1)
SC50 - SC54 - SC55  (9672-E03) (CMOS 1)       300 microseconds
```

Only 1472 lock entries are used out of 1 million **I**.

The number of requests delayed because of false contention is 98 for 19K lock requests **H**, which is 0.5% of the total. Because there is no problem for the number of available lock entries in the structure, the false contentions are due to the hashing algorithm. The value experienced is very good, but the sysplex is lightly loaded.

Note: The structure size can be dynamically increased if the structure runs out of lock entries because of the number of global outstanding requests.

6.9.3 GRS Ring - GRS Star Considerations

Table 7 and Table 8 on page 104 summarize the measurements data for GRS ring and GRS star configurations. The sysplex has ten systems, is lightly loaded, and has no contentions for coupling facility and ISGLOCK structure.

Table 7. GRS Ring and GRS Star Measurements Comparison

	GRS Ring	**GRS Star**
ENQ delay (msec)	30	9021 H5 0.5 - 0.7 CMOS 2 1.0 - 1.5 CMOS 1 2.0 - 3.0
CPU (% of one cp) (GRS AS. of 10 systems)	14.08%	3.35%
STORAGE (GRS AS. of 10 systems)	42MB	74MB
Global outstanding requests	2854	1600
Global ENQ/DEQ (SC47 per sec) *	0.7 (127 hours run)	0.7 (10 hours run)
Local ENQ/DEQ (SC47 per sec) *	2.5 (127 hours run)	2.6 (10 hours run)
ISGLOCK response time (microseconds)		9021 H5 140 CMOS 2 180 CMOS 1 300

Note: *

STEP and SYSZTIOT SYSTEM requests are not included. With these requests, the local ENQ/DEQ number would be two or three times greater than the reported value.

Sysplex global request at peak load 30 - 50 per second.

SC47 is one of the ten systems in the sysplex, and runs on H5 LPAR.

Table 8. GRS Ring and GRS Star RESERVEs Comparison		
Resource Qname	**GRS Ring (msec)**	**GRS Star (msec)**
SYSIGGV2 (converted)	51	6 (H5) 9 (CMOS2) 12 (CMOS1)
SYSZVVDS (excluded)	6	4 (H5)
SYSZVVDS (converted)		3 (H5) 6 (CMOS2) 9 (CMOS1)
SYSVTOC (excluded)	34	26 (H5)
SYSVTOC (converted)		26 (H5) 34 (CMOS2) 40 (CMOS1)
SYSZRACF (converted)	52	4 (H5) 14 (CMOS2) 15 (CMOS1)
Note: The time differences indicate DASD I/O response time improvement.		

The following is a summary of basic considerations that can help while choosing the GRS configuration:

- Use GRS ring when:
  - You are not experiencing problems with your current configuration
  - You have a small complex, equal to or less than four systems
  - You have a homogeneous set of processors
  - No coupling facility is available
  - You need a mixed ring configuration (GRS mixed complex)
- Use GRS star when you have:
  - A brand new installation
  - Problems with the enqueue delay or with RESERVEs not converted (hardware reserve still issued)
  - A large complex, equal to or greater than four systems
  - A heterogeneous set of processors

6.10 Resource Name List Processing

The RNL processing is not changed with GRS star support. This section has only been included for completeness.

Global resource serialization allows you to change the scope of an ENQ request without having to rewrite the existing programs, by using three resource name lists (RNL), as follows:

SYSTEM INCLUSION RNL Lists resources for ENQ requests with scope SYSTEM to be converted to scope SYSTEMS.

SYSTEMS EXCLUSION RNL Lists resources for ENQ requests with scope SYSTEMS to be converted to scope SYSTEM.

RESERVE CONVERSION RNL Lists resources for RESERVE requests for which the hardware reserve is to be suppressed and the requests are treated as ENQ requests with scope SYSTEMS.

Global resource serialization resource names are made up of two parts:

QNAME Specifies an eight-character name. Every request for a serially reusable resource must use the same QNAME, RNAME, and scope to represent a specific resource.

RNAME Specifies the 1 to 255 character name used with QNAME to represent a single resource.

Each RNL entry indicates whether the name is generic or specific. A specific resource name entry matches a search argument only when they are exactly the same. In contrast, a generic resource name entry is a portion of a resource name. A match occurs whenever the specified portion of the generic resource name entry matches the same portion of the input search argument.

Note: RNLs for every system in the global resource serialization complex must be identical; they must contain the same resource name entries, and these must appear in the same order. During initialization, global resource serialization checks to make sure that the RNLs on each system are identical. If they are different, global resource serialization does not allow the system to join the complex.

When global resource serialization receives an ENQ/RESERVE/DEQ request for a resource with a scope of SYSTEM or SYSTEMS, unless the request specifies RNL=NO or you have specified GRSRNL=EXCLUDE on IEASYSxx parmlib member, the installation-replaceable scan RNL exit (ISGGREX0) is invoked to search the appropriate RNL. A return code from ISGGREX0 indicates whether or not the input resource name exists in the RNL. See 6.10.2, “User Exit for Scanning the RNL” on page 108.

Input to the ISGGREX0 consists of the QNAME, the RNAME, and the RNAME length of the resource. The IBM default ISGGREX0 exit search routine finds a matching RNL entry when:

- A specific resource name entry in the RNL matches the specific resource name in the search argument.

  Note: The length of the specific RNAME is important. A specific entry does not match a resource name that is padded with blank characters to the right.
- A generic QNAME entry in the RNL matches the QNAME of the search argument.
- A generic QNAME entry in the RNL matches the corresponding portion of the resource name in the search argument.

Figure 73 on page 106 shows the RNL scan processing flow.

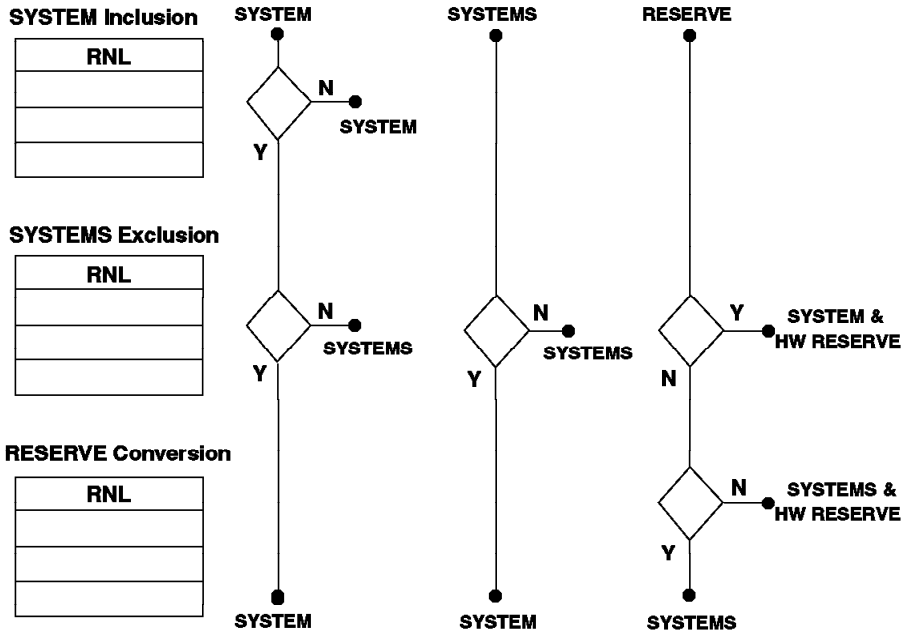


Figure 73. RNL Scan Processing

RNLs are scanned in the following order:

1. ENQ SCOPE=STEP requests are not filtered with the RNLs.
2. ENQ SCOPE=SYSTEM requests (local) are filtered by the SYSTEM INCLUSION RNL; those in the list are changed to ENQ SCOPE=SYSTEMS (global). Changes to ENQ SCOPE=SYSTEMS are filtered by the SYSTEMS EXCLUSION RNL.
3. ENQ SCOPE=SYSTEMS (global) requests are filtered by the SYSTEMS EXCLUSION RNL; those in the list are changed to ENQ SCOPE=SYSTEM (local).
4. RESERVEs are ENQ SCOPE=SYSTEMS plus a hardware reserve. Those ENQs are first filtered with the SYSTEMS EXCLUSION RNL. If they keep the SCOPE=SYSTEMS, they are filtered by the RESERVE CONVERSION RNL. Those in the RESERVE CONVERSION list have the hardware reserve suppressed.

DASD hardware reserve contention can be avoided by analyzing the use of data sets on shared DASD, selecting the resources that are causing DASD reserve contention, and including them in the RESERVE CONVERSION RNL. Also, note that when RESERVE CONVERSION RNLs are used for a shared DASD within a global resource serialization complex, the device should not be accessed by systems outside the complex.

To allow an installation to gradually migrate from a non-global serialization environment (GRS=NONE) to full global serialization use, use the GRSRNL=EXCLUDE option on the IEASYSxx parmlib member can be used to specify that all ENQ and RESERVE requests are excluded from global serialization and are processed locally. Global resource serialization is initialized into global serialization mode required by the sysplex support and can process the “guaranteed serialization” (ENQ SCOPE=SYSTEMS,RNL=NO)

requests globally. All other ENQ and RESERVE requests are processed locally, as in the non-global serialization environment.

The rest of this section assumes that GRSRNL=EXCLUDE has not been specified.

6.10.1 RESERVE Conversion

One major benefit of global resource serialization is that it can, through the RESERVE conversion RNL processing, alleviate the DASD contention problems created by hardware reserve serialization. These hardware reserve problems include:

- Interlocks
- Contention between jobs running on different systems that access the same device
- The possibility that one system might monopolize a shared device
- The data integrity exposure that occurs as a result of a system reset while a reserve is outstanding

The global resource serialization RESERVE conversion can be implemented in three different ways:

- Convert the RESERVE for a resource by placing the resource name in the RESERVE conversion RNL, and do not place its name in the SYSTEMS exclusion RNL. Global resource serialization suppresses the hardware reserve and treats the serialization request as a SYSTEMS scope ENQ request.
- Place the resource name in the SYSTEMS exclusion RNL. The hardware reserve is issued, and global resource serialization treats the serialization request as a SYSTEM scope ENQ.

  This choice is the recommended way to handle serialization requests where the hardware reserve must be issued. The processing delay due to the GRS global request is avoided because the implied ENQ is converted to SYSTEM scope.
- Do not place the resource name in either the RESERVE conversion RNL or the SYSTEMS exclusion RNL. The hardware reserve is issued and global resource serialization treats the serialization request as a SYSTEMS scope ENQ.

  Because both a hardware reserve and a SYSTEMS scope ENQ are issued, this choice effectively causes double serialization. Doing nothing, however, may lead to an interlock if jobs on different systems contend for multiple resources on the SAME SHARED DASD VOLUME, as shown in the following example:

  1. Job A on system SYS1 issues a RESERVE request for resource (A,B) on DASD device ABC.
  2. Job B on system SYS2 issues an *exclusive* RESERVE request for resource (C,D) on DASD device ABC.
  3. Job A on system SYS1 issues a RESERVE request for resource (C,D) on DASD device ABC.

  Job B on system SYS2 owns the SYSTEMS scope ENQ portion of resource (C,D), but it is unable to access DASD device ABC and release (DEQ)

resource (C,D), because JOB A has an outstanding hardware reserve for this device. When job A on system SYS1 issues a RESERVE request for resource (C,D), it is wait-listed (or enqueued) because job B on system SYS2 owns resource (C,D). If resources (A,B) and (C,D) had been in either the SYSTEMS exclusion RNL or the RESERVE conversion RNL, this interlock would not have occurred. Also, the interlock could not occur if all the RESERVE requests had been for *shared* control.

Note: Because of the higher overhead and the risk of deadlock, it is strongly recommended that resource names for which RESERVE is used appear in the RESERVE conversion or SYSTEMS exclusion RNL.

The following general recommendations for RESERVE conversion should be observed:

- Do not convert a reserve for a resource on a volume if any system in your complex shares the volume with a system that is not part of the global resource serialization complex (see 6.10.2, “User Exit for Scanning the RNL”).
- Do not convert the reserves for a resource when different systems in the complex use different names for the resource. This inconsistency can occur when the resource name includes system-dependent information, such as a control block address (for example, UCB).
- If an application issues a single reserve to serialize access to multiple resources on the same volume, you cannot convert the reserve unless you change the application.

Another general consideration is the use of the resource. You cannot convert some reserves because the application that uses the resource cannot tolerate the global ENQ delay time required to process a request for a global resource. Your installation might have applications where dependencies on quick access to a resource outweigh the additional availability resulting from converting the reserve. In this situation, put the resource names in the exclusion RNL.

6.10.2 User Exit for Scanning the RNL

Global resource serialization has a user exit, ISGGREX0, that receives control whenever an ENQ/DEQ/RESERVE request is issued for a resource. ISGGREX0 scans the input resource name list (RNL) for the resource name specified in the input parameter element list (PEL). A return code from the exit routine indicates whether or not the input resource name appears in the RNL.

You can use ISGGREX0 to determine whether the input resource name exists in the RNL. Replacing ISGGREX0 changes the technique that GRS normally uses to scan an RNL. Changing the search technique can have an adverse effect on system performance, especially when the RNLs contain many specific entries.

The routine has three external entry points:

- ISGGSIEX scans the SYSTEM inclusion RNL.
- ISGGSEEX scans the SYSTEM exclusion RNL.
- ISGGRCEX scans the RESERVE conversion RNL.

Depending on the RNL the request requires, the exit routine is invoked at the appropriate entry point for the SYSTEM inclusion RNL, the SYSTEM exclusion RNL, or the RESERVE conversion RNL.

You can modify the IBM-supplied exit routine to perform the processing necessary for your system. Use the source version of the exit routine provided as member ISGGREXS in SYS1.SAMPLIB.

For example, ISGGRCEX can be used to avoid converting reserves against a volume on a system outside the global resource serialization complex. An example is given in Appendix C, "ISGGREX0 Sample Exit" on page 199.

Note: The ISGGREX0 exit routine in each system belonging to the same global resource serialization complex must yield the same scan results; otherwise, resource integrity cannot be guaranteed. For the same reason, the RNLs themselves must be the same. The exit routine resides in the nucleus and to change or activate it, a sysplex-wide IPL is required. See *OS/390 MVS: Installation Exits* for details.

6.11 RNL Considerations

RNLs are in general highly installation dependent, which makes it impossible to provide default RNLs that would work in all environments. The IBM-supplied default for each RNL is shown in Figure 74. It is primarily meant to let you IPL the systems in a sysplex. An IPL with the default RNLs, however, may not reflect all the goals you have for the global resource serialization complex. For example, you may want to change the contents of the default RNL to emphasize RESERVE conversion and avoid potential interlocks.

```
SYSTEM Inclusion RNL:
       SYSDSN

SYSTEMS Exclusion RNL:
       SYSDSN PASSWORD
       SYSDSN SYS1.BRODCAST
       SYSDSN SYS1.DAE (applies only to MVS/XA and MVS/ESA systems)
       SYSDSN SYS1.DCMLIB
       SYSDSN SYS1.DUMP (generic -- all dump data sets)
       SYSDSN SYS1.LOGREC
       SYSDSN SYS1.MAN (generic -- all SMF data sets)
       SYSDSN SYS1.NUCLEUS
       SYSDSN SYS1.PAGE (generic -- all page data sets)
       SYSDSN SYS1.STGINDEX
       SYSDSN SYS1.SVCLIB
       SYSDSN SYS1.UADS

RESERVE Conversion RNL:
       The RESERVE conversion list is empty.
```

Figure 74. Contents of the Default Resource Name Lists

The generic QNAME entry for SYSDSN in the inclusion RNL list specifies that all data sets (except for VIO and subsystem data sets, such as SYSIN, SYSOUT, and SUBSYS data sets) are to be global resources. The entries in the exclusion RNL identify the system data sets with a QNAME of SYSDSN that specifically cannot be global resources.

Note, however, that if the shared master catalog supported since MVS/ESA Version 5 is used, the SMF, PAGE, and STGINDEX data set names must be unique for each system in the sysplex, and, therefore, the corresponding entries in the IBM-supplied RNL exclusion have to be changed.

The DISPLAY GRS,CONTENTION command provides information about resources that are causing contention, and RMF reports can also help. The most useful RMF reports are:

- Monitor I Enqueue Activity report
- Monitor II System Enqueue Contention (SENQ), and System Enqueue Reserve (SENQR) reports
- Monitor III (Workload Delay monitor) reports on resource-oriented ENQ delays and resource-oriented device delays

An ENQ/DEQ/RESERVE analysis aid captures and reports the use of resources serialized through the ENQ and DEQ SVCs. This aid is intended to assist in planning the RNLs during the implementation of a multiple systems global resource serialization environment. The tool (which is available in some countries as an IBM service) reports the following:

- Trace of ENQ/DEQ activities with resource names and user names
- Frequency of ENQ/DEQ by resource name
- The average, minimum, and maximum time of each RESERVE
- The total, average, minimum, and maximum RESERVE time and the maximum concurrent RESERVE count for those DASD devices that have RESERVE activity
- The global resource serialization global enqueue delay
- The behavior of RNLs when they are activated

The aid provides information either interactively through an ISPF dialog or through batch post-processing utilities. Examples of the reports available are shown in Appendix B, “ENQ/DEQ/RESERVE Analysis Aid Reports” on page 191.

In a sysplex, the GRS star complex can consist of OS/390 Release 2 or later systems. For these complexes, there are some general recommendations as well as some specific suggestions on known resources that are good candidates for the inclusion, exclusion, and conversion RNLs. As a general recommendation, a sysplex is to have as many resources as possible serialized as global resources.

Resource names for which RESERVE is used should appear in the exclusion or conversion RNL. The choice to exclude or convert can be based on the usage rate and the average time the RESERVE requires to complete, compared with global resource serialization global enqueue delay.

The following sections provide some general recommendations for common resource names.

6.11.1 CICS

If you are using CICS/XRF in a multi-MVS environment with DB2, you should use global resource serialization to ensure integrity of DB2. In the case of an alternate CICS system taking over from an active CICS system, the original DB2 region that the failed CICS was using must be completely terminated before a new DB2 can be restarted with the newly active CICS. Global resource serialization can reduce the risk of data integrity problems caused by concurrent execution of DB2 on both MVS systems. One recommendation is to set up global resource serialization to control key DB2 data sets so that only one DB2 region is allowed to update them at the same time. See *CICS/ESA XRF Guide* for additional information.

6.11.2 CVOLs

CVOLs (non-VSAM user catalogs) are protected by a RESERVE request that includes the UCB address. If you use CVOLs, you cannot convert the RESERVE request because the UCB address is system-dependent, and the resource name is not the same on every system.

To avoid the cost of propagating this resource in the complex, place a generic entry for QNAME(SYSCTLG) in the SYSTEM exclusion RNL.

6.11.3 DAE

Dump analysis and elimination (DAE) allows an installation to suppress SVC dumps and SYSMDUMP ABEND dumps that are not needed because they duplicate previously written dumps. A sysplex can share a single DAE data set.

The shared DAE data set, if it is not named SYS1.DAE, is already set up as a global resource by IBM-default RNLs; the QNAME SYSDSN is in the inclusion RNL. IBM recommends that you use a name different from SYS1.DAE for a DAE data set to be shared in a sysplex.

Note: In an OS/390 Release 2 sysplex environment, if you intend to use SYS1.DAE as the shared DAE data set for a sysplex, the default RNAME entry for SYS1.DAE data set, in the SYSTEMS exclusion RNL must be removed.

6.11.4 DB2

If you are using DB2 in a shared DASD environment, review *DB2 System Planning Administration Guide* for recommendations about tuning global resource serialization for DB2.

6.11.5 DFSMShsm

If you are using Data Facility Storage Manager (DFSMShsm) on all systems in the complex that run DFSMShsm, you do not need to take any action for user data set serialization as long as SYSDSN is named in the inclusion RNL. You can, however, convert some of the RESERVE requests that DFHSM issues, such as ARCBACV and ARCMIGV.

When using SMS-managed DASD volumes with DFSMShsm or DFHSM and USERDATASETSERIALIZATION active, DFSMShsm and DFHSM Version 2.6 try to delete temporary data sets during automatic primary space management. If global resource serialization is not performed for temporary data sets, an in-use temporary data set can be deleted.

More information about these RESERVE requests, as well as the facts to consider when deciding whether or not to convert them, appears in *Data Facility Storage Manager Installation and Customization Guide*.

6.11.6 DFSMS/MVS

You can place resource name IGDCDSXS in the RESERVE conversion RLN as a generic entry.

6.11.7 DFSMSrmm

If you are using Data Facility Storage Manager Removable Media Manager (DFSMSrmm) on all systems in the complex, you must be aware of the RESERVE requests on the DFSMSrmm control data set. It is recommended that you include the specific resource name in the SYSZRMM MASTER.RESERVE, in either the conversion RNL or the exclusion RNL.

Ensure the method you choose is the same for the SYSZVVDS resources. Both the SYSZRMM MASTER.RESERVE and SYSZVVDS resources must be treated in the same way to avoid potential lockouts between systems on these resources.

Note: Processing of other data on the volume that contains the RMM CDS may result in a deadlock condition if a RESERVE or a systems ENQ on the volume is required in order to process that data.

For example, a DFSMSdss full volume dump obtains a RESERVE or systems ENQ on the volume to prevent updates to the VTOC during dump processing. While the VTOC is held by DSS, a deadlock condition results if the following statements are true:

- DSS requires tape activity.
- The RMM CDS requires an additional extent in order to record that tape activity.

For additional information, see *DFSMSrmm Implementation/Customization*.

6.11.8 IMS

You can place resource name DSPURI01 in the RESERVE conversion RLN as a generic entry.

6.11.9 ISPF or ISPF/PDF

To serialize access to resources with concurrent batch or TSO/E use of the resources, ISPF relies on MVS allocation (QNAME of SYSDSN).

To ensure the integrity of shared data, batch or TSO/E users who are updating a data set must allocate it with DISP=OLD.

To serialize access to partitioned data sets among multiple ISPF users, ISPF also issues its own ENQ, DEQ, and RESERVE macros (QNAME of SPFEDIT), with scope SYSTEMS.

To allow users to update a data set that has a record format of “U,” ISPF serializes with the linkage editor to protect the entire partitioned data set (QNAME SYSIEWLP).

To convert SPFEDIT RESERVE and SYSIEWLP RESERVE requests for data sets you want to protect as global resources, place entries in the conversion RNL.

Note: If your complex includes both a system running ISPF and a system running SPF (a pre-ISPF product) you cannot use global resource serialization to serialize access to global resources among ISPF and SPF users. You must not include entries for either SPFEDIT or SPFDSN in the conversion RNL.

Resources that ISPF users on more than one system might share with batch users or TSO/E users must be global resources defined with both a QNAME of

SYSDSN and a QNAME of SPFEDIT. That is, you must define (either explicitly or by default), an entry in the inclusion RNL for SYSDSN, data_set_name.

In a JES complex, JES prevents a TSO user from logging on to more than one system at the same time using a single user ID. For additional information, see 6.11.16, “TSO/E” on page 117.

However, in a sysplex containing several JES complexes, TSO users may log on to several systems at the same time. In this situation, if you use ISPF on more than one system in the global resource serialization complex and do not use SPF on any system in the complex, the following considerations apply:

- The profile data set is protected by an exclusive ENQ SCOPE=SYSTEMS QNAME(SPFEDIT) RNAME(dsname-member_name). You cannot use the same profile data sets; either allocate different profile data sets on different systems, or place generic entries in the SYSTEM exclusion RNL for the QNAME(SPFEDIT) RNAME(dsname) for the profile data sets. The use of generic entries is suggested because the ENQ rname is made by the dsname-member_name.
- If one user logs on to more than one system after he submits a job on one system, he cannot submit another job on another system. The first submit allocates a work data set USERID.SPFCTLx.CNTL; the following submit tries to delete the data set and allocate a new one. The delete fails because the data set is serialized with a global ENQ. To avoid the problem, assign a preallocated system specific data set to the ISPCTLx DD-statement in the logon procedure.
- ISPF Exit 16 (Log, List, and Temporary Data Set Allocation Exit) lets you maintain your own data set naming conventions. ISPF calls the routine at this user exit before allocating the log, list, or temporary control data sets. As a result, you can provide a system with a specific prefix for the name of the data set to be allocated and avoid global resource serialization data set ENQ problems even if you allow multiple TSO logons for an user. However, if these data sets have been pre-allocated, ISPF does not use this prefix.

6.11.10 JES2

Do not convert the RESERVE request for the JES2 checkpoint data set. Even if in a GRS star complex, where JES2 performance is not impacted if the checkpoint data set is protected as a global enqueue (SYSZJES2 in the exclusion RNL), it is suggested you isolate the subsystem from the global resource processing to avoid having a GRS failure prevent JES2 from accessing its checkpoint. The location of the checkpoint data set, however, is another consideration that affects your decision:

- If the checkpoint data set is the only data set on a device or resides on a device that contains other data sets that are never serialized by RESERVE requests, do not convert the RESERVE request; include the name of the checkpoint data set in the exclusion RNL.
- If the checkpoint data set resides on a device that contains other data sets that are serialized by RESERVE requests, include the names of all data sets in the conversion RNL. This use of a device contradicts recommendations for placement of the JES2 checkpoint data set.

If your installation also uses an alternate checkpoint data set, the RESERVE request JES2 issues for the primary checkpoint data set also provides serialization for the alternate.

Note: With MVS/SP-JES2 Release 5.1.0, the primary checkpoint data can reside on a list structure in a coupling facility. In this case, JES2 does not use the RESERVE service. The serialization is through the coupling facility list lock index. It is recommended you allocate the alternate checkpoint data set on a DASD device. If the alternate checkpoint data set becomes the primary, the RESERVE macro is used again.

6.11.11 JES3

There are no JES3 specific global resource serialization considerations. JES3 performance may suffer if the JES3 checkpoint data set is on a device where *other* data sets are serialized with hardware reserves.

In general, the major advantage of global resource serialization in a JES3 environment is in the conversion of hardware reserves to avoid interlocks and reduce contention for data sets on shared DASD devices. Global resource serialization also provides the only way to serialize access to new non-specific, non-SMS managed DASD data sets across multiple systems.

6.11.12 RACF

You may already be sharing your RACF database. Sharing the database in a pre-RACF Version 2 Release 1 environment requires that:

- The database reside on shared DASD.
- The database name table (ICHRDSNT) is compatible for all sharing systems.
- The database range table (ICHRRNG) is identical on all sharing systems.
- The class descriptor table (ICHRRCDE) is compatible for all sharing systems.

RACF database sharing coexists with earlier versions of RACF, RACF on VM, and versions of MVS supported by the RACF release.

If you are experiencing RACF database contention problems, consider converting the hardware reserves, as long as all systems that access the RACF database are MVS systems that are part of the same global resource serialization complex. If a VM system is sharing access to the RACF database, you cannot convert the hardware reserves. To convert the RESERVE requests, place a generic entry for SYSZRACF in the RESERVE conversion RNL.

Actually, SYSZRACF can be in either the conversion or the exclusion RNL, depending on which results in better performance.

Note: When RACF Version 2 Release 1 (or later, including the OS/390 Security Server) is enabled for sysplex communication and is in data sharing mode, RACF changes its enqueue logic and uses a SYSTEMS enqueue with qname SYSZRACF for command propagation. Therefore, SYSZRACF cannot be in the SYSTEMS exclusion list.

In any case, do not put an entry that begins with SYSZRACF or SYSZRAC2 in the SYSTEM inclusion RNL.

RACF Version 1 Release 9 is using only the RESERVE service to serialize database accesses. You may continue to share your database as you have done

in the past when you migrate to RACF Version 2 Release 1. However, RACF Version 2 Release 1 provides you with the following new options that may be beneficial to your installation. When RACF Version 2 Release 1 is enabled for sysplex communication and is in non-data sharing mode, the database sharing requirements are as follows:

- MVS/ESA Version 4 Release 2 or higher is installed.
- All sharing systems must be in the same multisystem sysplex.
- The major name SYSZRACF cannot be in the exclusion RNL.
- The database resides on shared DASD.
- The database name table (ICHRDSNT) is compatible for all sharing systems.
- The database range table (ICHRRNG) is identical on all sharing systems.
- The class descriptor table (ICHRRCDE) is compatible for all sharing systems.

The RACF database name table (ICHRDSNT) enables a system for sysplex communication. When enabled, a RACF system can be in one of several modes for accessing the RACF database. The mode is controlled by a flag in the database name table (ICHRDSNT) and can later be modified through use of the RVARY command.

Systems enabled for sysplex communication and running in non-data sharing mode can share a database with earlier RACF systems and with RACF on VM systems. Systems that are not enabled for sysplex communication but that are sharing the database cannot exploit command propagation. In this mixed environment, the systems must not enter data sharing mode.

When RACF Version 2 Release 1 is enabled for sysplex communication and is in data sharing mode, the requirements are as follows:

- MVS/ESA Version 5 or higher is installed on all sharing systems.
- RACF Version 2 Release 1 is installed on all sharing systems.
- All sharing systems have access to the same coupling facility structure.
- All sharing systems are enabled for sysplex communication.
- Only members of the multisystem sysplex can share the RACF database.
- The major name SYSZRACF cannot be in the exclusion RNL.
- The database resides on shared DASD.
- The database name table (ICHRDSNT) is compatible for all sharing systems.
- The database range table (ICHRRNG) is identical on all sharing systems.
- The class descriptor table (ICHRRCDE) is compatible for all sharing systems.

Sysplex data sharing does not coexist with earlier versions of RACF or with RACF on VM. Database corruption can occur if this is attempted. RACF Version 2 Release 1 maintains frequently used information in a coupling facility cache structure for rapid access.

For additional information, see *RACF System Programmer's Guide*.

6.11.13 System Logger

The use of the system logger in a sysplex environment allows the single system view of some logging functions. IBM supplies two log streams for customer use: the logrec log stream, and the operation log stream (OPERLOG). The data sets allocated on behalf of the log streams are serialized with the major name SYSDSN and minor name “data_set_name.” If a generic entry for qname SYSDSN is in the SYSTEM inclusion RNL, the log streams are globally serialized. If not, the following RNL definition is required if the default high level qualifier is used:

```
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(SYSDSN) RNAME(IXGLOGR)
```

or, if allocated with a different high level qualifier (“hlq”):

```
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(SYSDSN) RNAME("hlq")
```

The same applies for customer-created log streams.

The logrec recording medium can be either a logrec data set or a logrec stream. If you are using a logrec data set, each system in the sysplex uses its own logrec data set. If the default data set name is used, and if qname SYSDSN is in the SYSTEM inclusion RNL, then the data set name should be in the SYSTEMS exclusion RNL as SYS1.LOGREC.

```
RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSDSN) RNAME(SYS1.LOGREC)
```

If you use different data set names on each system, it is not required to put those data set names in the SYSTEMS exclusion list.

6.11.14 Tape Volumes

If tape drives are being used across multiple MVS images and volume serial numbers are unique, add a generic entry for SYSZVOLS to the SYSTEM inclusion RNL to insure that a tape volume is used by only one job at a time.

```
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(SYSZVOLS)
```

When using automatic tape switching, adding this entry also prevents a system from holding a tape device while waiting for a mount of a volume that is being used by another system.

If tape drives are not being used by multiple MVS images, no RNL entry is required.

6.11.15 Temporary Data Sets

If you take no action with the default RNLs, global resource serialization treats non-VIO temporary data sets as global resources. One reason for letting temporary data sets be global resources is that your installation can then run scratch functions, such as IEHLIST SCRATCH VTOC,SYS and DFSMShsm automatic space management, safely at any time against shared volumes. For more information, see 6.11.19, “VTOC” on page 124 and 6.11.5, “DFSMShsm” on page 111.

However, if your installation wants global resource serialization to treat temporary data sets as local resources, the data set names must appear in the exclusion RNL. The format of a temporary data set name, however, is not

compatible with the format of an RNL entry. There is no way to create a generic entry that works all the time in every installation. The format of a temporary data set name is one of the following:

```
SYSyyddd.Thhmmss.RA000.jobname.R0000nnn
SYSyyddd.Thhmmss.RA000.jobname.ddname
```

If your installation wants to treat the temporary data sets as local resources, add the following generic entry in the SYSTEM exclusion RNL:

```
RNLDEF RNL(EXCL) TYPE(GENERIC)
QNAME(SYSDSN)
RNAME(SYS9)
```

Note: This entry excludes all temporary data sets created between 1/1/1990 and 12/31/1999 from global serialization. It also excludes all permanent data sets which begin with SYS9, such as SYS9.USERLIB, so use it with care.

DFSMShsm attempts to delete temporary data sets during automatic primary space management. If global serialization is not performed for temporary data sets, an in-use temporary data set can be deleted.

6.11.16 TSO/E

When one or more of the systems in your global resource serialization complex runs TSO/E, you must decide whether or not to treat SYS1.UADS and a sysplex SYS1.BRODCAST as global resources, and how to handle user data sets. If your installation uses the RACF database (requires RACF Version 1 Release 8 or later) in place of SYS1.UADS, you have the option to define *all* TSO/E users through the RACF data set instead of through the UADS, and to delete the SYS1.UADS data set. However, it is recommended that you continue to maintain information for at least one user ID in the UADS. Continuing to maintain the UADS allows you to log on to TSO/E if RACF is not active or the RACF database cannot be accessed.

Note: If you delete the SYS1.UADS data set, you must also delete from the master JCL the DD statement associated with the SYS1.UADS data set.

If your installation is running TSO/E Version 2 Release 3 and shares the SYS1.BRODCAST data set with other systems, you should update the IKJTSOxx parmlib member SEND statement to indicate the environment in which the LISTBC command executes. If all the TSO/E systems are running in the same sysplex (this also implies the same global resource serialization complex), you can avoid I/O to SYS1.BRODCAST for the display of system notices by setting the SYSPLEXSHR keyword on the SEND statement to ON. When the keyword of the SEND parmlib statement is ON, the LISTBC command notifies the other systems of changes to SYS1.BRODCAST using the XCF services.

If your installation shares the SYS1.BRODCAST data set with other systems outside the sysplex, you should set the SYSPLEXSHR keyword of the SEND parmlib statement to OFF. When the keyword of the SEND parmlib statement is OFF, updates are made to the SYS1.BRODCAST data set and are read from the SYS1.BRODCAST data set. This ensures that all updates are observed by all systems sharing the data set.

Because SEND, OPERATOR SEND, and LISTBC access the broadcast data set to store and retrieve messages, you may experience contention for the data set. Instead of storing messages (mail) in the SYS1.BRODCAST data set, you can use individual user logs to store the messages. A user log is a data set that the

SEND command or subcommand processor uses to store messages in, rather than storing the messages in the SYS1.BRODCAST data set. Using user logs may reduce possible contention for the broadcast data set.

To implement the use of user logs, update the IKJTSOxx parmlib member SEND statement LOGNAME and CHKBROD parameters. Note that when user logs are used, the SYS1.BRODCAST data set should not be preallocated in the logon procedure.

The default exclusion RNL includes entries for SYS1.UADS and SYS1.BRODCAST, causing them to be local resources. This may allow a user to be logged on to more than one system at the same time, providing there are no other required resources being globally serialized. For ISPF considerations, see 6.11.9, “ISPF or ISPF/PDF” on page 112.

However, in a JES complex, users cannot log on at the same time to several systems in the same JES complex because of duplicate job names. Treat SYS1.UADS and SYS1.BRODCAST as global resources, if they are used, to gain the following advantages:

- Your installation has only two data sets to maintain, rather than two data sets for each TSO/E system in the complex.
- A user can log on from any system in the complex, allowing a better workload balance, but cannot be logged on to more than one system at a time.
- For foreground-initiated background jobs, a user who specifies NOTIFY always receives the job-ended message, regardless of which system in the complex processed the job.

To cause global resource serialization to treat SYS1.UADS and SYS1.BRODCAST as global resources, you must:

- Merge all existing versions of SYS1.UADS and SYS1.BRODCAST into a single version of each data set.
- Modify the default RNLs by:
  - Deleting the entries for SYS1.UADS and SYS1.BRODCAST from the SYSTEMS exclusion RNL
  - Adding SYSIKJUA as a generic QNAME entry in the inclusion RNL to make SYS1.UADS a global resource

    Note: If multiple logons for the same user ID are desired in a global resource serialization complex, the generic entry in the SYSTEM inclusion RNL for the QNAME SYSIKJUA should serialize only the SYS1.UADS and SYS1.BRODCAST data sets:

    ```
    RNLDEF RNL(INCL) TYPE(GENERIC)
    QNAME(SYSIKJUA) RNAME(SYS1)
    ```

    TSO/E uses the QNAME(SYSIKJUA) RNAME(user_ID) to prevent multiple logons for the same user ID within a system (unless requested otherwise by Logon Pre-Prompt Exit (IKJEFLD(1))
  - Adding SYSIKJBC as a generic QNAME entry in the inclusion RNL to make SYS1.BRODCAST a global resource.

Related information appears in *TSO Extensions Version 2 Customization*. Some of the information is repeated here for your convenience.

If your installation uses the RACF database in place of SYS1.UADS, see 6.11.12, “RACF” on page 114.

The data sets that are used only by TSO/E users include:

- Private user data sets that are not shared by other users or by batch jobs
- Temporary user-related data sets, such as ISPF, log, or recovery data sets
- Shared data sets, such as program libraries

If you use the default RNLs, the SYSDSN entry in the inclusion RNL defines all these data sets as global resources. If your installation wants all of them or some of them to be local resources, you must exclude them from global serialization. You can, if the structure of your user ID allows, place a generic entry in the exclusion RNL to define all TSO/E user data sets as local resources. If, however, a user might log on to different systems at different times, that user′s data sets must be global resources. Place a generic entry for the user ID in the SYSTEM inclusion RNL, and do not place an entry for the user in the SYSTEMS exclusion RNL.

If the structure of your user ID does not allow you to create a generic entry to define all TSO/E user data sets as local resources, you can place a generic entry in the SYSTEMS exclusion RNL for the user ID of each user whose TSO/E data sets are to be local resources. Omit the entry for a user whose TSO/E data sets are to be global resources. This method, however, might cause a very long RNL that you would have to change frequently. If the problem is significant at your installation, you might want to modify the exit search routine (ISGGREX0) to recognize the TSO/E user data sets that are to be local resources and exclude them from global serialization. See *OS/390 Release 2 MVS: Installation Exits* for more information.

6.11.17 VIO Journaling Data Set

The VIO journaling data set is a VSAM data set that contains auxiliary storage management records for virtual I/O (VIO) data sets that the system saves across job steps and between IPLs. The default journaling data set name, SYS1.STGINDEX, is defined in the default SYSTEMS exclusion RNL as a local resource.

In a sysplex of MVS/ESA Version 5 or higher systems, you should define a unique name for the VIO journaling data set on each system. If you do, use the symbolic &SYSNAME on the VIODSN parameter in your IEASYSxx parmlib member. In this case it is not necessary to put SYS1.STGINDEX name in the SYSTEMS exclusion RNL.

In a sysplex that also has MVS/ESA Version 4 systems, the data set SYS1.STGINDEX is required in the RNL exclusion list.

6.11.18 VSAM

Whether you can serialize access to VSAM data sets and VSAM and ICF catalogs as global resources depends on the level of VSAM installed on all systems in the global resource serialization complex. The current level of VSAM is required with global resource serialization. The current level of VSAM is available in MVS/370 DFP Version 1.1 or later, in MVS/DFP Version 2.1 or later, in MVS/DFP Version 3, and in DFSMS/MVS Version 1 or later.

VSAM Data Sets: As a general guideline, treat SYSVSAM the way you treat SYSDSN. If you take no action, SYSVSAM is a global resource. If you make SYSDSN resources local, then make SYSVSAM resources local; delete the entry for SYSDSN from the default inclusion RNL and add an entry for SYSVSAM to the exclusion RNL.

Note: Do not place an entry for SYSVSAM in the SYSTEM inclusion RNL. Every SYSVSAM request that needs the SYSTEMS attribute already has it. Placing an entry for SYSVSAM in the SYSTEM inclusion RNL can degrade performance.

Allocation Serialization: Allocation serialization depends on the existence of a generic QNAME entry for SYSDSN in the inclusion RNL as well as on the user's specifying the correct parameter (DISP=OLD or DISP=SHR) in the JCL. It is thus a less effective method of control than OPEN serialization, which depends on the VSAM share options.

Open Serialization: If, as usually happens, the DD statement for the data set specifies DISP=SHR, then OPEN processing enforces the VSAM cross-region share options 1 and 2. OPEN processing does not affect cross-region or cross-system share options 3 and 4. For share options 3 and 4, OPEN issues an ENQ for major name SYSVSAM with SYSTEM scope. Therefore, for share options 3 and 4, users must provide their own READ/WRITE operation serialization.

GRS extends the serialization afforded by cross-region share options 1 and 2 to all systems in the GRS complex. An ENQ for major name SYSVSAM scope SYSTEMS is issued by OPEN, and whether the SYSVSAM ENQ is exclusive or shared depends on the cross-region SHAREOPTIONS and whether the data set has been OPENed for input or output.

- If a data set is defined with SHAREOPTIONS 1 and OPENed for input, a shared SYSVSAM ENQ is issued.
- If a data set is defined with SHAREOPTIONS 1 and OPENed for output, an exclusive SYSVSAM ENQ is issued.
- If a data set is defined with SHAREOPTIONS 2 and OPENed for input, a shared SYSVSAM ENQ is issued.
- If a data set is defined with SHAREOPTIONS 2 and OPENed for output, a shared SYSVSAM ENQ is issued.

As an example, consider the following scenario:

- A VSAM cluster defined with cross-region SHAREOPTIONS 1 is opened for output on CPU A. CPU B is unable to open the cluster for either input or output.
- A VSAM cluster defined with cross-region SHAREOPTIONS 2 is opened for output on CPU A. CPU B is able to open the cluster for input (without read integrity) but not for output.

For these reasons, global resource serialization automatically treats the VSAM data set as a global resource.

If you do not want global resource serialization to treat a VSAM data set as a global resource, you must add a SYSTEMS exclusion RNL for each VSAM component of the data set; SYSVSAM data set requests are always done at a component level. The RNAME of the SYSTEMS exclusion entry must consist of the component name (with all trailing blanks truncated) concatenated to the catalog name (with all trailing blanks truncated). Because VSAM concatenates additional information after the catalog name, make the entry generic.

READ/WRITE Operation Serialization: Cross-region share options 3 and 4, where serialization is the user's responsibility, require the user to provide serialization for READ/WRITE operations as well as specify DISP=SHR on the DD statement. The usual way to protect the integrity of the data set is to serialize its use by RESERVE and DEQ services.

If your application programs issue RESERVE macros for a VSAM data set, you can convert the RESERVE requests under the following conditions:

- No application programs that run on systems outside the complex ever access the data set.
- Your application programs use consistent and repeatable names for the data set on every system in the sysplex environment.
- Your application programs always acquire and release resources in the same order.
- Your installation converts any RESERVE requests that system services issue to the same device on behalf of your application.

Unless all of these conditions exist, you cannot convert the RESERVE requests.

If you are converting the RESERVE requests, place entries for the data sets in the conversion RNL. If you are not converting the RESERVE requests, place entries for the data sets in the exclusion RNL. The form of the entry (generic or specific) and the QNAME depend on the specific conventions at your installation, but be sure that the entry or entries cover each name the application programs use for each data set.

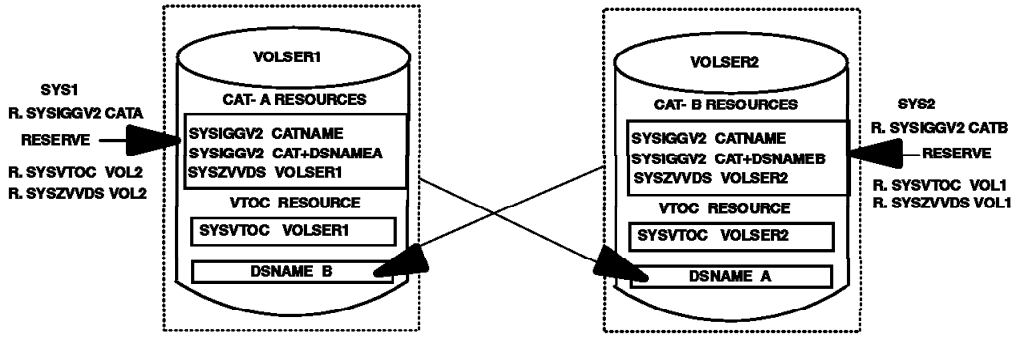
VSAM Catalogs and ICF Catalogs: VSAM catalogs and ICF catalogs are VSAM data sets, but VSAM recognizes catalogs and manages access to them in a special way. If catalog hardware reserve contention is a concern at your installation, replacing VSAM catalogs with ICF catalogs can help. Converting catalog RESERVE requests can reduce contention caused by catalog activity, and also lessens the possibility of catalog lockouts as described in the following scenario.

Catalog management uses reserves with qnames SYSIGGV2 and SYSZVVDS while serializing access to catalogs. The SYSIGGV2 reserve is used to serialize the entire catalog BCS component across all I/O, as well as to serialize access to specific catalog entries. The SYSZVVDS reserve is used to serialize access to associated VVDS records. The SYSZVVDS reserve and the SYSIGGV2 reserve together provide an essential mechanism to facilitate cross-system sharing of catalogs. The RESERVEs are issued only if the catalog is properly defined for sharing (that is, SHARE(3,4) and on a unit defined to MVS as shared).

In a sysplex, this design has a potential exposure when data sets reside on the same volume as other shared catalogs. Deadlocks can occur in the following scenario:

SYS1 and SYS2 share DASD volumes, volser1 and volser2. The SYSIGGV2 reserve is held for CATALOG A (volser1) by SYS1 while trying to obtain a reserve for data set A (volser2). SYS2 has a SYSIGGV2 reserve for CATALOG B (volser2) while trying to obtain a reserve for data set B (volser1). Reserves for data set A or data set B could be for SYSVTOC or SYSZVVDS.

Figure 75 on page 122 shows the scenario.



★ **MVS Catalog and DADSM functions use the following RESERVEs:**

- ► **SYSIGGV2 catname, SYSIGGV2 dsname+catname SYSZVVDS volser, SYSVTOC volser (resource granularity)**
- ► **RESERVE has volume granularity**
- ► **If RESERVEs not converted, interlock may occur**
- ► **Suggested: CONVERT SYSIGGV2 (for Ring and Star)**
- ► **EXCLUDE SYSZVVDS, EXCLUDE SYSVTOC (for Ring)**
- ► **EXCLUDE SYSZVVDS, CONVERT SYSVTOC (for Star)**
- ► **CONVERT SYSZVVDS, CONVERT SYSVTOC (for Star and low enq-delay, 1 msec.)**

Figure 75. VSAM CATALOG RESERVEs

Figure 76 on page 123 shows the processing for dsname SYS4.M.P on volume IODFPK, catalogued on catalog CATALOG.SHRICF1.VIODKBK, which is on volume IODFBK.

You can see that the RESERVE for SYSIGGV2 (dsname+catname) protects all the catalog processing, and that there are nested RESERVEs for SYSIGGV2 (catname), and RESERVEs for SYSZVVDS, and SYSVTOC for volumes IODFPK and IODFBK.

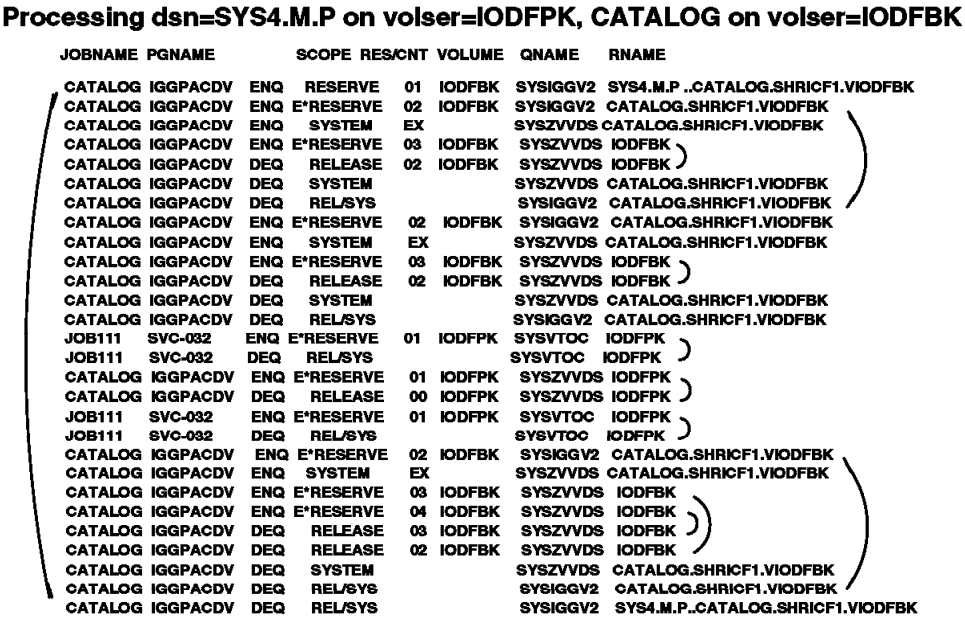


Figure 76. VSAM CATALOG and DADSM RESERVEs Nesting

To prevent such deadlocks, always convert the SYSIGGV2 reserve to a SYSTEMS ENQ. An exception to this is for catalogs shared with systems outside the sysplex. See recommendations for this configuration in topics on VTOC and RNL candidates in this chapter.

If you are using a GRS ring configuration, it is not necessary to convert SYSZVVDS and SYSVTOC reserves, as these reserves are of short duration. Conversion to global enqueues would slow down the catalog and VTOC processes. See 6.11.19, “VTOC” on page 124 for additional information on SYSVTOC.

If you are using the GRS star configuration, the global enqueue delay can be on the average one millisecond (see 6.9.2.2, “Enqueue Delay” on page 97, 6.9.2.3, “RESERVE Requests Behavior” on page 99) and may not affect SYSZVVDS and SYSVTOC requests. With a GRS star configuration, you may consider adding qnames SYSZVVDS and SYSVTOC to the conversion RNL table.

The following considerations apply to catalog and VTOC RESERVEs:

- To avoid data integrity exposure, verify that all systems using an ICF catalog are part of the complex.
- If you are using a GRS ring configuration, place a generic entry for SYSZVVDS in the SYSTEMS exclusion RNL.

  Note: If you are using a GRS star configuration, consider adding qname SYSZVVDS to the conversion RNL table, if the global enqueue delay is not affecting the response time of SYSZVVDS requests.
- Place a generic entry for SYSIGGV2 in the RESERVE conversion RNL.

Note: A generic QNAME entry is recommended for SYSIGGV2 because catalog management uses two naming conventions:

- To serialize the entire catalog, catalog management uses QNAME(SYSIGGV2) RNAME(ucatname)
- To serialize a catalog entry, catalog management uses QNAME(SYSIGGV2) RNAME(userdsn ucatname)

- To avoid the potential interlock problems between catalog and DASD space management, it is suggested to have specific entries qname(SYSVTOC) rname(volser) for volumes containing shared catalogs and data sets catalogued in those catalogs in the RESERVE conversion RNL.

  See 6.11.19, “VTOC” for additional information on SYSVTOC when using GRS star configuration.

If you are sharing volumes with systems outside the GRS complex, the only serialization mechanism is the hardware reserve: global enqueues are not propagated to systems outside the GRS complex. It is suggested you use the GRS RNL user scan exit to manage this configuration. See Appendix C, “ISGGREX0 Sample Exit” on page 199, and 6.10.2, “User Exit for Scanning the RNL” on page 108 for an example.

If the shared volumes have VSAM catalogs, it is recommended you place specific SYSIGGV2 entries for the catalog names in SYSTEMS exclusion RNL.

Note: When using specific entries for SYSIGGV2 and the catalog name is less than 20 bytes, pad it to 20 bytes with blanks; if the name is more than 20, pad it to 44 bytes. If you have a catalog naming convention in place, you can use catalog name prefixes and generic entries. In the following example, the third entry can be used to replace the first two entries.

```
RNLDEF RNL(EXCL) TYPE(SPECIFIC)
QNAME(SYSIGGV2)
RNAME('CATALOG.SHRICF1.VIODFPK              ') <-- padded to 44 bytes

RNLDEF RNL(EXCL) TYPE(SPECIFIC)
QNAME(SYSIGGV2)
RNAME('CATALOG.SHRICF1.VIODFBK              ') <-- padded to 44 bytes

RNLDEF RNL(EXCL) TYPE(GENERIC)
QNAME(SYSIGGV2)
RNAME(CATALOG.SHRICF1)
```

For additional information, see *DFSMS/MVS V1R3 Managing Catalogs*.

6.11.19 VTOC

As a general recommendation for GRS ring configuration, place a generic entry for SYSVTOC in the SYSTEMS exclusion RNL, because RESERVE requests are of short duration.

However, if you plan to run DFDSS DEFRAG programs with MVS/SP 2.2.0 or later in a multisystem environment, you can convert the SYSVTOC RESERVE request by including a generic entry in the conversion RNL. Because converting SYSVTOC RESERVE requests to global ENQs slows down VTOC processing (as the global enqueue delay is significant), you may want to consider using the

dynamic RNL capability to specify one set of RNLs (which convert SYSVTOC RESERVE requests) for DFDSS DEFRAG processing and one set (which leave the RESERVE requests) for normal processing.

In general, SYSVTOC should not be converted. An exception to this rule occurs for the VTOC of devices containing the sysplex couple data sets, and the VTOC of devices containing catalogs shared with systems outside the sysplex (see "VSAM Catalogs and ICF Catalogs" on page 121). Even though SYSVTOC RESERVE requests are ordinarily very short, there are cases where long RESERVE requests might be possible, for example, during a DFDSS backup of the device. However, converting these RESERVE requests could result in longer response times for VTOC accesses for those devices.

In a GRS star configuration, the global enqueue delay can average one millisecond (see 6.9.2.2, "Enqueue Delay" on page 97, 6.9.2.3, "RESERVE Requests Behavior" on page 99) and may not slow down the VTOC processing. Consider adding qname SYSVTOC to the conversion RNL table and avoid the problems previously described.

6.11.20 RNL Candidates

Based on these general recommendations, there are certain resources that are good candidates for a particular RNL. Table 9 shows suggested resources for the inclusion RNL, Table 10 shows suggested resources for the exclusion RNL, and Table 11 shows suggested resources for the conversion RNL.

Figure 116 on page 197 shows an example for the inclusion RNL, Figure 117 on page 198 shows an example for the exclusion RNL, and Figure 118 on page 198 shows an example for the conversion RNL. The exclusion and conversion RNLs are set up for the ISGGREX0 RNL conversion user exit (see Appendix C, "ISGGREX0 Sample Exit" on page 199) to share volumes with systems outside the GRS complex.

For each resource shown in the tables, the figures include information on the resource name and a brief description of why you should consider placing the particular resource in the RNL.

In the tables, a letter indicates the original scope of the resource in the coding: **L** for SYSTEM, **G** for SYSTEMS and **R** for RESERVE. If more than one letter appears, it means that the QNAME can be used with more than one scope.

Note: You must specify the parts of the resource name shown in upper case letters exactly as shown, and you must replace the parts of the resource name shown in lower case letters with your installation-specific information.

Table 9 (Page 1 of 2). System Inclusion RNL Candidates

QNAME	RNAME	Scope	Notes
SYSDSN	dsname (optional)	L	Include a generic QNAME entry for SYSDSN to make data sets that go through MVS allocation global resources.
SYSIKJBC	none	L	Required if the TSO/E data set SYS1.BRODCAST is to be a global resource. Remember to delete the entry for SYSDSN, SYS1.BRODCAST from the default SYSTEMS exclusion RNL.

Table 9 (Page 2 of 2). System Inclusion RNL Candidates

QNAME	RNAME	Scope	Notes
SYSIKJUA	none	L	Required if the TSO/E data set SYS1.UADS is to be a global resource. Remember to delete the entry for SYSDSN, SYS1.UADS from the default SYSTEMS exclusion RNL.
SYSZVOLS	none	L	Include a generic QNAME entry for SYSZVOLS to insure that a tape volume is only used by one system at a time if multiple MVS images are sharing tape drives and volume-serial numbers are unique.
any	any		Include an entry for any request that you want to include. Global resource serialization treats the resource as a global resource. The name can be specific or generic.

Table 10 (Page 1 of 2). SYSTEM Exclusion RNL Candidates

QNAME	RNAME	Scope	Notes
SYSCTLG	none	R	You must include an entry for SYSCTLG. The name must be generic.
SYSDSN	dsname	L	Include an entry for every system data set that is to be a local resource. The default RNLs contain several (see Figure 74 on page 109).
SYSDSN	SYS9	L	To treat temporary data sets created between 1/1/1990 and 12/31/1999 as a local resource. Permanent data sets whose names start with SYS9 are also treated as local. With DFSMShsm automatic space management, in-use temporary data sets can be deleted. Also, IEHPROGM can scratch in-use temporary data sets if they are not globally serialized.
SYSIGGV2	catalog name	R	Put a specific entry for any VSAM catalog used by any system that is not part of GRS complex. See “VSAM Catalogs and ICF Catalogs” on page 121 and Appendix C, “ISGGREX0 Sample Exit” on page 199.
SYSVSAM	dsname (optional)	L-G	It is not generally recommended to include entries for VSAM data set resources. To make all VSAM data sets local resources, use a generic QNAME entry for SYSVSAM. SYSVSAM and SYSDSN data set serialization must be consistent. However, if SYSDSN is in the SYSTEM inclusion RNL, it is not recommended to include SYSVSAM because every SYSVSAM request that needs the SYSTEMS attribute already has it.
SYSVTOC	volser of SYS1.DCMLIB data set	R	For MVS releases earlier than SP2.2, include this entry to IPL with MVS/XA DFP.
SYSVTOC	none	R	In a GRS ring configuration, it is recommended because reserves are of short duration and I/O intensive. In a GRS star configuration, the global enqueue delay can average one millisecond. You may add qname SYSVTOC to the conversion RNL table. See 6.9.2.2, “Enqueue Delay” on page 97, 6.9.2.3, “RESERVE Requests Behavior” on page 99, and SYSVTOC in Table 11.

Table 10 (Page 2 of 2). SYSTEM Exclusion RNL Candidates			
QNAME	**RNAME**	**Scope**	**Notes**
SYSZIAT	volser dsn	R	In a JES3 environment, include an entry for the checkpoint data set.
SYSZJES2	volser dsn	R	In a JES2 environment, include an entry for each checkpoint data set specified in the CKPTDEF initialization statement. For JES2 Version 1.3.6 or JES2 Version 2.1.5, use the volser and dsname specified for PRIMARY and DSNAME in CKPTDEF; for JES2 Version 2.2 or later releases, include a separate entry for each checkpoint data set, and use the volser and dsname specified for CKPT1, CKPT2, NEWCKPT1, and NEWCKPT2 in CKPTDEF.
SYSZVVDS	none	L-R	In a GRS ring, include this entry to treat SYSZVVDS reserves as local resources, because the reserves are I/O intensive and of short duration. In a GRS star configuration, the global enqueue delay can be on the average of one milliseconds. You may add qname SYSZVVDS to the conversion RNL table. See 6.9.2.2, “Enqueue Delay” on page 97 and 6.9.2.3, “RESERVE Requests Behavior” on page 99.
any	any		Include an entry for any reserve that you want to exclude. Global resource serialization treats the resource as a local resource, and the system issues the reserve. The name can be specific or generic.

Table 11 (Page 1 of 2). Reserve Conversion RNL Candidates			
QNAME	**RNAME**	**Scope**	**Notes**
ARCGPA	ARCRJRN (padded to 8 bytes)	R	DFHSM journal data set. If dumped using DFHSM or DFDSS full volume dump, you must include in the conversion list SYSVTOC-volser of the volume containing the journal data set. For more information, refer to *DFHSM Installation and Customization Guide*.
IGDCDSXS	none	R	SMS reserves.
SPFEDIT	data sets (none)	G-R	ISPF reserves as global resources. You must have those data sets in the inclusion list for SYSDSN to serialize with TSO/E and batch. If SPF and ISPF are both used, do not include SPFEDIT in the conversion list.
SPZAPLIB	dsname	R	Include an entry to convert this reserve if all systems that share the resource are part of the complex.
SYSIGGV2	catalog dsn (optional and padded to 20 or 44 bytes)	R	Include this entry to convert the reserves for VSAM and ICF catalogs. Including this entry also avoids catalog deadlocks. If you use specific entries and the dsname is less than 20 bytes, pad it to 20 bytes; if the dsname is more than 20 bytes, pad it to 44 bytes. See “VSAM Catalogs and ICF Catalogs” on page 121.
SYSIEWLP	dsname (optional and padded to 44 bytes)	R	Include an entry to convert this reserve if all systems that share the resource are part of the complex. The entry serializes access between ISPF and the linkage editor.

Table 11 (Page 2 of 2). Reserve Conversion RNL Candidates

QNAME	RNAME	Scope	Notes
SYSVTOC	none	R	In a GRS star configuration, the global enqueue delay can on the average of one msec. See 6.9.2.2, “Enqueue Delay” on page 97 and 6.9.2.3, “RESERVE Requests Behavior” on page 99. Consider including this entry. In a GRS ring configuration, is not recommended because reserves are of short duration and I/O intensive. See SYSVTOC in Table 10, 6.11.5, “DFSMShsm” on page 111. If you plan to run DFDSS DEFRAG programs with MVS/SP 2.2 or a later version in a multisystem environment, include SYSVTOC as a generic entry.
SYSVTOC	volser of couple data set pack	R	Consider converting the reserves against the couple data set packs to global ENQs to reduce the likelihood that a long reserve would cause a status update missing condition. Do this only if there is ordinarily only light activity on the VTOC of these packs.
SYSVTOC	volser of shared catalog pack	R	Consider converting the reserves against the shared catalogs packs to global ENQs to reduce the likelihood of catalog lockout. See “VSAM Catalogs and ICF Catalogs” on page 121.
SYSZRACF	none	R	Include this entry to convert the reserves for the RACF data base. The name should be generic.
SYSZVVDS	none	L-R	In a GRS star configuration, the global enqueue delay can on the average of one msec. You may consider to include this entry. See 6.9.2.2, “Enqueue Delay” on page 97 and 6.9.2.3, “RESERVE Requests Behavior” on page 99. In a GRS ring configuration, is not recommended because reserves are of short duration and I/O intensive. The name is generic.
any	any		Include an entry for any reserve that you want to convert. Global resource serialization treats the resource as a global resource, and the system does not issue the reserve. The name can be specific or generic.

6.11.21 RNL Syntax Checker

To help avoid unnecessary IPLs due to incorrect RNLs, a program, ISGRNLCK, is available to check their syntax. The syntax checker is shipped in source form in SYS1.SAMPLIB and consists of the assembler programs, ISGRNLCK, ISGRNLMS, and ISGRNLPR.

The RNL syntax checker lets you check the syntax of GRSRNLxx lists specified in the JCL. ISGRNLCK performs three major tasks:

- Searches for the specified GRSRNLxx members of SYS1.PARMLIB
- Verifies the syntax of RNLDEF statements in each member
- Issues information and error messages

If ISGRNLCK returns a return code of zero, you can IPL the system successfully.

Note: If the RNLDEF keyword is missing from an RNL statement in the SYS1.PARMLIB member, the RNL syntax checker immediately issues an error message and does not check the rest of the statement. Thus, after correcting an error, execute the RNL syntax checker again on the specified SYS1.PARMLIB members to verify the remaining RNLDEF statements.

Chapter 7. Coupling Facility Failure Policy

This section discusses the enhancements that have been made to the connect and rebuild services and the benefits they can offer your installation. The following items are discussed:

- Rebuild service improvements and how using these can help avoid further problems for an application that suffers connectivity loss.
- Connect service improvements and how an application can request the type of connectivity that it requires.
- Improved diagnostics that are available during connect and rebuild processing.

Full details on these topics can be found in the following manuals:

- *OS/390 MVS Sysplex Services Guide*
- *OS/390 MVS Sysplex Services Reference*
- *OS/390 MVS Setting up a Sysplex*
- *OS/390 MVS System Commands*

7.1 Overview

Cross-System Extended Services (XES) provides the support to manage and use coupling facility structures. The XCF Coupling Facility Failure Policy (CFFP) and the rebuild enhancements improve this support in the following ways:

1. Through better coupling facility (CF) selection during structure allocation.
2. By allowing applications to define their initial connect requirements to the CF.
3. By enhancing the structure rebuild process to ensure that the rebuilt structure has equivalent or better connectivity than the old structure. This is applicable to rebuilds initiated due to:
   - Connectivity failures
   - Operator command
   - Program intervention

These improvements are achieved by enhancing the CF selection algorithm that is used to determine which CF is most optimally connected to systems in the sysplex. This reduces the risk of applications losing their data sharing capability should an inadequately configured CF be chosen. In fact, a rebuild activity is stopped or not initiated at all if it is determined that it is causing or will cause a degradation to applications based on CF connectivity.

7.2 XCF Service Changes

Two Cross-System Coupling Facility services have been enhanced to deliver this capability. They are:

- Connect Service (Macro: IXLCONN)
  - IXLCONN allows you to allocate and connect to a structure in a CF, or to connect to an existing structure. During structure allocation, the CF selection process now uses the system weight values that are specified

in the Sysplex Failure Management (SFM) policy. This information is used by Cross-System Extended Services (XES) to determine the CF that is connected to the most important systems.

  - Applications can now specify their CF structure connectivity requirements. Through a new parameter, CONNECTIVITY, you can indicate the required scope of system connectivity to the CF in which the structure is allocated. When the initial connection to the structure takes place, the connectivity is examined by XES and, if found to be unsatisfactory, the connection is failed. Thus, by not permitting structure allocation in a coupling facility that does not fully meet the application′s connectivity requirements, this helps prevent impact to the application.

- Rebuild Service (Macro: IXLREBLD)
  - A structure can be rebuilt through macro invocation or by issuing the SETXCF START,REBUILD command. After this is done, XES does not process a rebuild that reduces the connectivity of a structure in the sysplex. Instead, CFRM compares the connectivity that existed in the old structure to the available connectivity in the new structure. Based on this information, it can stop or fail to initiate the following causes of rebuilds that would give inadequate connectivity:
    1. Loss of connectivity - if the new structure′s connectivity is less than or equal to that of the old structure at the time of the failure
    2. All other rebuilds - if the new structure′s connectivity is less than that of the old structure

7.3 CF Selection During Structure Allocation

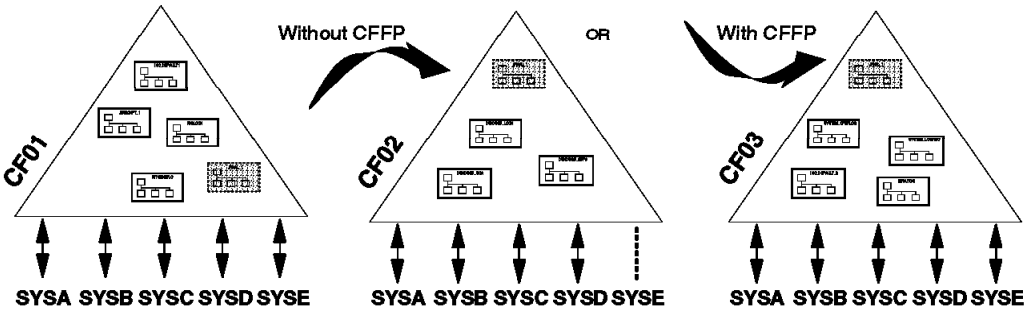
Prior to OS/390 Release 2, the coupling facility resource manager (CFRM) examined the preference list in the policy for a structure when the first IXLCONN request for that particular structure occurred, and attempted to allocate the structure in a CF where both the IXLCONN-requested attributes (CFLEVEL, NONVOLREQ and STRSIZE) and the CFRM policy attributes (SIZE, Preference List, Exclusion List) could be accommodated. If all of the requested attributes were satisfied by all of the CFs in the preference list, then the first CF in the preference list was selected for allocation.

This can lead to less than optimal allocation results. When an IXLCONN is processed from a system which has two CFs available and the first CF in the preference list is not accessible from all other systems in the sysplex, the first CF is always selected by the system performing the allocation, even though it offers worse connectivity for some systems. In this scenario, the other CF is more suitable for the application′s structure, but the selection process does not take into account the current connectivity environment. Allocating the structure in a CF accessible from the majority of systems may also not improve the selection process, as this does not consider the installation′s rated value of importance for that system.

The diagram in Figure 77 shows three CFs and their interconnections between five systems: three development (SYSA, SYSB and SYSD) systems and two production (SYSC and SYSE) systems. CF01 and CF03 can be reached by all production and development systems. CF02 is accessible by all systems except SYSE. It also shows the weights associated with each system. Consider the following:

1. There are three CFs in the preference list for structure JRNAL_1, which is shared between the production systems. If a SETXCF START,REBUILD command is issued, the structure is rebuilt in CF02, and SYSE would lose its connectivity to the structure.
2. In the event of a signalling failure between CF01 and SYSE, certain structures, based on the active CFRM policy, would be rebuilt by the first system that detects the error. This could mean that even after a successful rebuild, SYSE might still have no connectivity to critical structures, because these may have been rebuilt in CF02.

In both of these examples, it would have been more beneficial to choose the CF accessible from both production systems, even when production systems are outnumbered by development systems.



SFM Policy Extracts

```
SYSTEM NAME(*)
  WEIGHT(2)
 ISOLATETIME(30)

SYSTEM
NAME(SYSA)
  WEIGHT(10)

SYSTEM
NAME(SYSC)
  WEIGHT(75)

SYSTEM
NAME(SYSE)
  WEIGHT(65)
```

CFRM Policy Extracts

```
STRUCTURE NAME(JRNAL_1)
SIZE(16128)
   PREFLIST(CF01,CF02,CF03)
```

SETXCF START,REBUILD,STRNAME=JRNAL_1,LOCATION=OTHER

Figure 77. CF Selection During Allocation or Rebuild

The XCF failure policy and rebuild enhancements address this requirement by making the CF selection algorithm use the SFM weights of systems attached to a coupling facility when determining its suitability for allocation. CFRM attempts to maximize the aggregate connector system weight to ensure that the most important systems remain connected to the structure.

Prior to OS/390 Release 2, the attributes specified on IXLCONN were used to allocate the structure at the first attempt to connect to that structure if it was not already allocated in a CF. The structure was allocated in the first CF in the preference list, which met the following default requirements:

- There was connectivity to the system trying to allocate the structure.

- The CF operation level (CFLEVEL) was equal to or greater than the requested CFLEVEL on the IXLCONN macro.
- There was available space equal to or greater than the target structure size.
- The CF met the volatility requirement requested by the connector via the NONVOLREQ keyword on the IXLCONN macro.
- The CF did not contain a structure in this structure′s exclusion list.

If no CFs in the preference list met all the requirements, then XES tried to allocate the structure without the exclusion list requirement, and then without the volatility requirement. If the CFLEVEL requirement could be satisfied, XES tried to allocate the structure in a CF that met or exceeded the CFLEVEL requirement and had the most available space. If the allocation was not successful, XES attempted to allocate the structure in a facility that had any CFLEVEL lower than the required CFLEVEL and also had the most available space. If the structure was successfully allocated, the actual structure size may have been smaller than the target size.

At this point CFRM potentially had a list of CFs that satisfied the application′s structure allocation requirements.

The enhanced CF selection process selects a CF from that list by considering the SFM weights of the systems attached to each CF at the time of the IXLCONN request. As expected, a CF with connectivity to all systems in the sysplex is a better choice than one with connectivity to any subset of systems. However, when several of the CFs in the preference list have only partial connectivity, CFRM selects the CF that is accessible from the set of systems with the highest combined SFM weight. CFRM does this by sorting each grouping of CFs according to the aggregate SFM weight of the systems connected to it.

As CFRM attempts to find a CF that meets all the requirements described previously, it sorts the list of CFs that meet all requirements by aggregate SFM system weights. If it does not find a qualifying CF, it drops the exclusion list requirement and sorts that list by SFM weight, until it finds a CF that:

1. Meets as many of the installation (CFRM policy) and connector (IXLCONN) requirements as possible
2. Has the best available connectivity across the sysplex

Note: If SFM is not active because you do not have an active SFM policy, or because one or more systems do not have access to the SFM couple data sets, CFRM treats every system as having equal weight.

Since the CF selection is now dependent upon the number and weights of systems in the sysplex at the time of the initial connect, the CF that CFRM chooses as the *most optimal* may vary, especially as systems are brought into, or removed from, the sysplex. This happens because the set of systems against which connectivity is being evaluated changes dynamically over time. This is especially noticeable when the entire sysplex is being IPL′d. An IXLCONN done from SYSA, when it is the only system in the sysplex, might have different results when the IXLCONN is attempted after four more systems have been brought in.

Note: The IXLCONN done as part of the REBUILD process is not affected in this way, since the list of current connectors is known by CFRM and is used to optimize the selection of a CF.

7.4 Improvements For Initial Connects

The CF selection algorithm used before OS/390 Release 2 may create undesirable structure connectivity for some applications. Even when the best CF available is selected for a structure, it may not be good enough for the application's needs. One example of this is the GRS/STAR support that was introduced in Release 2. For GRS/STAR to be successful, the GRS structure (ISGLOCK) has to be accessible from all systems in the sysplex.

By means of the new CONNECTIVITY parameter on the IXLCONN macro, applications can indicate their connectivity needs. They can also be informed when the connectivity they require is not available. The following summarizes the various values for the CONNECTIVITY= parameter:

- **SYSPLEX**

  The structure is allocated in a CF that provides connectivity to *all* systems currently in the sysplex. If no CF satisfying this requirement is available, the connection request is rejected with return and reason codes. Conditions causing this type of request to fail are:

  - A system is not running MVS/ESA Version 5 or above.
  - Some systems in the sysplex do not have access to the CFRM couple data sets.
  - Some systems in the sysplex are not capable of CF attachment.
  - Some systems in the sysplex do not have working links to a common CF.

  This option is useful when an application requires that either the entire sysplex participates, or nothing does. Again, GRS/STAR is an example of this, since it does not initialize unless every active system can participate by connecting to the ISGLOCK structure.

- **BESTGLOBAL**

  A CF is selected that provides connectivity to the best combination of systems, as determined by the SFM weights. If the selected CF is not accessible from the system that issued the initial connection, the request fails with return and reason codes indicating why.

- **DEFAULT** (Bestlocal)

  A CF is selected that provides connectivity to the local system. If more than one CF is accessible from the local system, the CF that also provides connectivity to the best combination of systems, based on SFM weighting, is selected.

In the process of CF selection, XES also continues to factor in the other attributes we discussed earlier, in determining the most suitable CF.

This new option reduces the instances in which the first connector to the structure causes it to be allocated in a CF with poor connectivity to other systems in the sysplex, which usually results in other connectors not being able to connect later on. Correction often requires manual intervention involving policy changes and rebuilds to move the structure to a more suitable location.

Note: The CONNECTIVITY parameter only applies to the initial allocation of a structure. The specification is ignored on subsequent connections.

7.5 Improvements For Rebuilds

Structure rebuild allows a connector to a structure to allocate that structure with the same name and reconstruct the data in it. Rebuilding gives connectors the flexibility to change the location or the attributes of the structure without having to disconnect from it. The process can be initiated by an authorized program, the operator, or XCF/XES. The common interface is the IXLREBLD macro.

Prior to OS/390 Release 2, the rebuild process had no checks and balances to determine whether it was improving or worsening the application's execution environment when allocating a new structure. Improvements have been made to IXLREBLD that enable the current connectivity to be evaluated during the rebuild. CFRM fails a rebuild if it cannot find a CF that maintains or improves the application's connectivity at the time the rebuild is started.

The IXLREBLD macro has been updated to include new reason codes for return code IclRcEnvError:

1. Insufficient Connectivity (IxlRsnCodeInsuffConn)

   No CF in the preference list provided equivalent or better connectivity than the current CF.

2. No Better Connectivity (IxlRsnCodeNoBetterConn)

   No CF in the preference list provided better connectivity than the current CF.

System-initiated rebuilds are now failed for the following situations:

- Rebuilds started due to loss of connectivity

  CFRM fails the rebuild with return code IxlRcEnvError and reason code IxlRsnCodeNoBetterConn. This happens when connectivity to the new structure is less than or equivalent to connectivity to the old structure after the connectivity loss.

- Rebuilds started for any other reason

  CFRM fails the rebuild with return code IxlRcEnvError and reason code IxlRsnCodeInsuffConn. This happens when connectivity to the new structure is less than to the old structure.

In addition, when XCF/XES automatically initiates a rebuild for the application because the REBUILDPERCENT value specified in the CFRM policy was reached, the current connectivity is examined and the rebuild is not initiated if the application's connectivity cannot be improved.

This improves the RAS of the rebuild function by stopping or not initiating a rebuild that causes a degradation in CF connectivity. It effectively removes from operations the evaluation decision of whether the rebuild-in-progress is causing connectors to lose connectivity, and the actions that operations would have to take before a point of no return is reached. CFRM now handles these issues. This improves reliability and availability because it avoids forcing the application into a situation when it might lose its data sharing capabilities on one or more systems.

There are situations when the rebuild must proceed despite these conditions. Coding the LESSCONNACTION=CONTINUE parameter allows the rebuild to complete regardless of the effect on the connectors.

Note: Stopping a rebuild requires full application participation, regardless of whether it was initiated by the operator, CFRM, the application, or another program.

7.6 Technical Activities

There are three areas for which certain considerations and actions need to be addressed:

1. **Systems**
   - If you do not already have an active SFM policy, consider establishing one so that system weights can be used for structure allocation decisions.
   - If you are using an SFM policy, decide whether changes to the system weights are necessary, given their new usage in structure allocation decisions.
   - Decide whether CFRM policy preference list changes are necessary to take advantage of the improved structure allocation algorithms.
   - Understand the connectivity requirements of particular exploiters (for example, GRS) and make fully connected resources available to them.
   - Ensure that you understand when it is, or is not, appropriate to use the LESSCONN option on an operator-initiated rebuild.
2. **Applications**
   - Understand and consider exploiting the use of the CONNECTIVITY parameter on the IXLCONN service macro to define the application′s connectivity requirements.
   - Understand and consider exploiting the LESSCONNACTION parameter on the IXLREBLD service macro to override default connectivity checks in the rebuild service.

   Note: Subsequent to OW19718, SFM always considers system WEIGHTs and REBUILDPERCENT. Previously, this was only done if CONNFAIL(YES) had been specified in the SFM policy. This means that system weights are always used for calculating the aggregate connector system weight, provided an SFM policy is active. This is also true for the CF selection process.
3. **Operations**
   - Understand and know when to use, and when not to use, the LESSCONN keyword on an operator-initiated rebuild.

Note: The systems programmer need not take enabling steps for the REBUILD enhancements. However, to take advantage of IXLCONN′s new CONNECTIVITY options, it is necessary to change particular application programs.

7.7 Command Support

The *SETXCF* command has been changed to allow a rebuild to be performed that results in loss of connectivity for one or more connectors. This overrides XCF′s new default action of stopping a rebuild that causes degradation to an application′s structure connectivity. In the following examples, the LESSCONN (or LC) keyword achieves the same result as the LESSCONNACTION parameter on the IXLREBLD macro.

```
SETXCF START,REBUILD,CFNAME=cfname,LESSCONN=CONTINUE|TERMINATE
SETXCF START,REBUILD,CFNAME=cfname,LC=CONTINUE|TERMINATE

SETXCF START,REBUILD,STRNAME=strname,LESSCONN=CONTINUE|TERMINATE
SETXCF START,REBUILD,STRNAME=strname,LC=CONTINUE|TERMINATE
```

CFNAME,LESSCONN=CONTINUE

LESSCONN=CONTINUE with CFNAME indicates that *all* structures in the specified CF are to be rebuilt, regardless of any degradation in connectivity for the connectors to the structure.

STRNAME,LESSCONN=CONTINUE

LESSCONN=CONTINUE with STRNAME indicates that the structure specified is to be rebuilt, regardless of any degradation in connectivity for the connectors to the specified structure.

Note: You can expect additional operator action after this command completes. Using this option to force a rebuild will most probably result in an application losing some processing capacity as some systems are not able to achieve connectivity to the new structure. You will need to bring on additional paths to the CF before applications on systems with impacted connectivity can resume processing.

The DISPLAY XCF,STR command has been updated to include text for the new rebuild stop reason codes.

7.8 Diagnostic Improvements

Several new informational messages have been added, and a number of existing messages include additional information. It is now easier to determine the cause of problems. In addition, the IXLCONN and IXLREBLD macros return more comprehensive diagnostic information via return and reason codes, also making it easier to correct problems.

The following list summarizes the *new* messages and shows some examples.

- IXC526I STRUCTURE *strname* IS REBUILDING FROM

  XCF has chosen a facility to contain the structure that is being rebuilt.

```
IXC526I STRUCTURE ISGLOCK IS REBUILDING FROM
COUPLING FACILITY CF01 TO COUPLING FACILITY CF02.
REBUILD START REASON: OPERATOR INITIATED
INF0108: 0000004D 0000004D.
```

- IXC527I THE REBUILD OF *totalrebstr* STRUCTURES

  The operator requested a rebuild of all the structures in a coupling facility but not all structures could be rebuilt.

```
IXC527I THE REBUILD OF 1 STRUCTURES IN COUPLING FACILITY CF01
COULD NOT BE STARTED. SPECIFIC ERROR MESSAGES FOLLOW.
THREAD: 0000045F
```

- IXC528I THE REBUILD OF STRUCTURE *name* IN COUPLING FACILITY

  An operator entered a SETXCF START/STOP REBUILD command to start or stop rebuilding all structures in one or more CFs, but the command was not successful for one or more of the structures in a CF.

```
IXC528I THE REBUILD OF STRUCTURE IXC_DEFAULT_2 IN COUPLING FACILITY
CF01 COULD NOT BE STARTED:
XCF SIGNALLING STRUCTURES CANNOT BE REBUILT USING CFNAME KEYWORD
THREAD: 0000045F
```

- IXL013I IXLCONN REQUEST FOR STRUCTURE *strname* FAILED

  An attempt to connect to a CF structure via the IXLCONN macro failed.

- IXL014I IXLCONN REQUEST FOR STRUCTURE *strname* WAS SUCCESSFUL

  A connect to a CF structure via the IXLCONN macro was successful.

```
IXL014I IXLCONN REQUEST FOR STRUCTURE ISGLOCK WAS SUCCESSFUL.
JOBNAME: GRS ASID: 0007 CONNECTOR NAME: SC55
CFNAME: CF02
```

- IXL015I STRUCTURE ALLOCATION INFORMATION FOR

  A program attempted to connect to a CF structure.

```
IXL015I STRUCTURE ALLOCATION INFORMATION FOR
 STRUCTURE ISGLOCK, CONNECTOR NAME SC55
  CFNAME     ALLOCATION STATUS/FAILURE REASON
  --------   --------------------------------
  CF01       STRUCTURE ALLOCATED
  CF02       PREFERRED CF ALREADY SELECTED
```

The following list shows the messages with changed text. The highlighted portion of the message is the area of change.

- IXC465I REBUILD REQUEST FOR STRUCTURE strname WAS result WHY REBUILT: rebuildrsn {WHY STOPPED:}**stoppedrsn**
- IXC367I THE SETXCF {START|STOP} REBUILD REQUEST FOR {STRUCTURE|COUPLING FACILITY} name WAS {ACCEPTED|REJECTED}: **text**
- IXC522I REBUILD FOR STRUCTURE strname IS BEING STOPPED DUE TO **text**

  The highlighted portion of the changed messages includes the text for the two rebuild stop reason codes.

  – *NO COUPLING FACILITY PROVIDED BETTER CONNECTIVITY*

    No other CF has better connectivity than the current one.

  – *NO COUPLING FACILITY PROVIDED EQUIVALENT OR BETTER CONNECTIVITY*

    No CF has equivalent or better connectivity than the current one.

Note: IXL012I has been deleted, effectively replaced by IXL013I and IXL014I

7.9 Migration and Coexistence Considerations

If the sysplex includes systems older than OS/390 Release 2, then the improvements provided for the initial connect or the early stages of rebuild are not available when these are done from such a system. Checks for equivalent or better connectivity are not made, as the algorithm in that release is used for CF selection. In addition, the command and macro changes are not recognized when attempted from such earlier systems.

Compatibility APAR (OW19718), retrofitted back to MVS/ESA Version 5, removes the need to code CONNFAIL(YES) in an SFM policy. Previously, this parameter had to be present for REBUILDPERCENT or WEIGHT to be considered. The rebuild decision process illustrated in Figure 78 shows the previous dependence on the CONNFAIL(YES) parameter and how this restriction has been removed in OS/390 Release 2, or systems with the APAR applied.

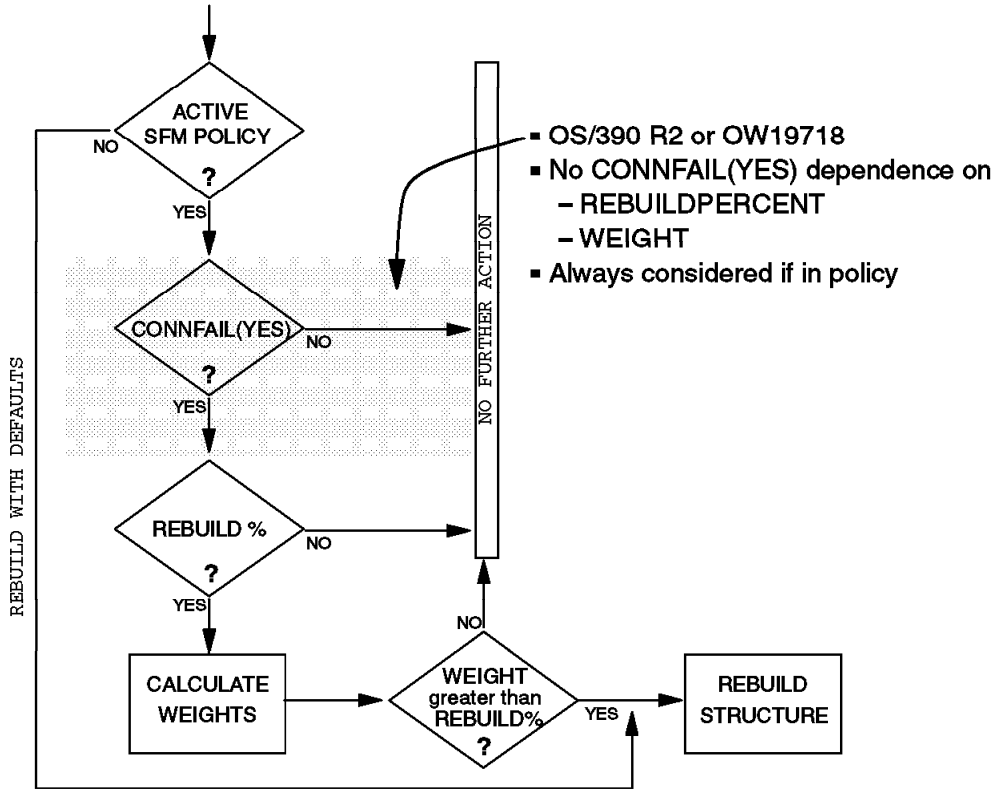


Figure 78. Structure Rebuild Viability

This improvement makes it easier for installations that had concerns about coding CONNFAIL to activate an SFM policy. SFM′s ability to partition certain systems out of the sysplex can continue to be restricted by specifying CONNFAIL(NO), while at the same time allowing the benefits based on REBUILDPERCENT and WEIGHT to be exploited.

There is no impact to ITR, response time, or storage usage. There is a small increase in the time to initially allocate and connect to the structure, due to additional checking to ensure better CF selection. However, certain rebuild

processing is eliminated when it is determined that system connectivity or structure capacity would be reduced.

Chapter 8. SMP/E Enhancements

This section discusses the functional and usability enhancements that have been made to SMP/E at the level packaged with OS/390 Release 2. The following items are discussed:

- The new BUILDMCS command
- Improvements made to BYPASS processing relating to system holds
- FMIDSET selection
- SMPTLIB allocation improvements
- The inclusion of the FIND command to key dialog processes
- The removal of GIMOPCDE from parmlib
- Year 2000 support

Full details on these topics can be found in the following manuals:

- *OS/390 SMP/E User's Guide*
- *OS/390 SMP/E Commands*
- *OS/390 SMP/E Reference*

8.1 BUILDMCS Command

The BUILDMCS command creates the data necessary to reinstall the specified product (or products), which greatly reduces the amount of manual work and therefore also decreases the number of errors that can occur.

This improvement provides a more automated and reliable method for copying products from one pair of target and distribution zones and their libraries into another pair of target and distribution zones and their associated libraries. The BUILDMCS command creates a function SYSMOD image (MCS and JCLIN), used as input to RECEIVE, APPLY, and ACCEPT processing, for reinstallation into another SMP/E environment. These images incorporate all service and user modifications currently installed for the specified FMIDs. Reinstallation allows for the requisite checking needed to ensure that the product is being installed into an appropriate environment.

Part of the output from the BUILDMCS command is a superseding function SYSMOD for each FMID specified on this command, and a superseding function SYSMOD for any dependent functions related to a specified FMID. These superseding functions include all maintenance and user modifications that have been installed in the input zone specified on the BUILDMCS command.

The following MCS statements are generated by the BUILDMCS command:

- ++Element
- ++HFS
- ++JCLIN
- ++MAC
- ++MOD
- ++SRC

A new operand, FROMDS, has been added to these statements. This operand specifies the distribution library in which the element text is located. During RECEIVE processing, these data sets are used as input and copied to SMPTLIB data sets, similar to the way RELFILE data sets are used. Since the data does not originate from a RELFILE, no file number is available to be used as a connector between the RELFILE data set and the corresponding SMPTLIB data set. Instead, the process has been changed to generate a pseudo file number. This number is used as the low-level qualifier of the SMPTLIB data set and as the connector between the distribution and SMPTLIB data sets.

The following example shows the commands that were used to generate a function SYSMOD for the High-Level Assembler, FMID HMQ4120.

```
SET      BOUNDARY(MVST100) .

BUILDMCS FORFMID(HMQ4120) .
```

8.1.1 BUILDMCS Output

The BUILDMCS command produces the following output:

- Function SYSMOD

  This SYSMOD supersedes any function with the same name and also any maintenance that had been applied. Everything, including user modifications, is included within the generated function SYSMOD. These are the statements that are processed by the RECEIVE, APPLY and ACCEPT phases. The generated statements are written to the SMPPUNCH data set.

- Function Summary Report

  This report summarizes the processing that occurs for every specified FMID. The SYSMODs applicable to each processed FMID are also listed.

- Entry Summary Report

  This report summarizes the processing that occurred for every eligible entry. A report is produced for each superseding function created.

The actual output has been included in section D.2, “BUILDMCS Entry Summary Report” on page 211.

8.1.2 BUILDMCS Sequence of Events

This is the sequence you would follow to copy High Level Assembler (5645-001 HMQ4120) to another SMP/E environment:

1. Determine the zones in which the product is currently installed.
2. Determine if the service levels of the target and distribution zones are the same. Use the ACCEPT and RESTORE commands as appropriate to accept or restore any differences.

   Note: ACCEPT raises the service level in the distribution zone. RESTORE lowers the service level in the target zone.

3. Use the distribution zone on the SET BOUNDARY command.

   If you do not ACCEPT JCLIN because ACCJCLIN is not specified in the DLIBZONE entry, then the BUILDMCS command should be run against the

target zone. JCLIN is required to create entries similar to load modules. JCLIN can only exist in the distribution zone if ACCJCLIN is set.

4. Run BUILDMCS.
5. Determine which DDDEFs need to be defined in the new target and distribution zones and define them.

   Use the DDDEF entry information from the BUILDMCS Entry Summary Report.

6. Allocate the new data sets for which DDDEFs have just been defined.
7. RECEIVE, APPLY, and ACCEPT the function SYSMOD generated by the BUILDMCS step.

8.2 Dialog Improvements

This functional improvement provides primary FIND command support on selected table display panels of the SMP/E base dialogs. Prior to this improvement, the only way to find a particular subentry within the Query dialogs was to scan the data visually while scrolling through the display.

The following base dialogs have been modified to deliver this support:

- SYSMOD management dialogs
- Query dialogs
- Receive dialogs

The FIND command works on selected SMP/E CBIPO dialogs to locate a string of characters in a dialog table display panel.

Note: The FIND command only searches for the string of data within the table display part of the panel. It does not search the fixed section of the panel.

8.3 Grouping Function Sysmods

FMIDSETs have been available for some time to group specific FMIDs together. However, with OS/390 Release 2, the use of the FMIDSET has been enhanced to allow it to be used on the APPLY, ACCEPT, RECEIVE, and RESTORE commands.

These commands have been changed so that FMIDSET can be specified on the existing SELECT operand. Specification of FMIDSET indicates that the FMIDs included in the FMIDSET should be processed together.

The following SMP/E areas have been modified to accommodate this change:

- RECEIVE, APPLY, ACCEPT, and RESTORE processing

  A value specified in the select list for RECEIVE, APPLY, ACCEPT, and RESTORE command processing is considered an FMIDSET when it is a 1-8 character (alphanumeric or national) value and the Global zone contains an FMIDSET entry by that name. In this situation, all FMIDs defined in the FMIDSET are processed as if they were explicitly specified in the select list.

  A value specified in the select list is determined to be a SYSMOD if it is not defined as an FMIDSET in the Global zone and has a name of seven characters.

- Dialogs supporting:

- SYSMOD management
- Command generation
- RECEIVE generation

The value entered is first assumed to be an FMIDSET. The Global zone is checked to see if the value entered is a defined FMIDSET. If it is, processing continues unchanged. If it is not defined and it is a seven character value, it is assumed to be an FMID and processing continues unchanged.

8.4 Hold Processing Improvements

The BYPASS operand has been enhanced in the area of system HOLDs. Using the BYPASS(HOLDSYSTEM) operand of the APPLY and ACCEPT commands, you can bypass a HOLD for a specific reason ID that affects a single SYSMOD or group of SYSMODs. Previously, this could only be done on an all-or-nothing basis. Only those SYSMODs specified on the command are eligible to have the system HOLD bypassed. Any other SYSMODs being processed are failed if a HOLD condition is in effect for them. This improves the usability of the APPLY and ACCEPT commands by allowing more granularity when dealing with HOLD information.

This support affects the following areas:

- APPLY and ACCEPT Processing

  The APPLY and ACCEPT commands have had the BYPASS(HOLDSYSTEM) operand extended to include the sysmod_id. This is where you specify the specific SYSMODs that you want bypassed. This is illustrated in Figure 79.

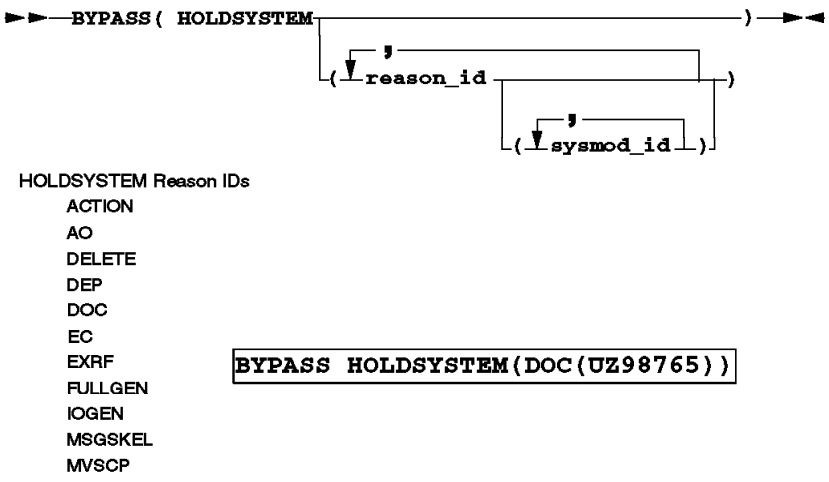


Figure 79. New BYPASS HOLDSYSTEM Enhancements

- SYSMOD Install and Command Generation Dialogs

The dialogs have been changed for both the APPLY and ACCEPT paths of panels. New fields have been added or amended to allow you to indicate that the list of SYSMODs that are not held for the Reason ID are to be entered on a subsequent panel. The end result is a generated command containing the enhanced BYPASS(HOLDSYSTEM) operand.

8.5 SMPTLIB Enhancements

SMP/E can now determine the data set type from which the RELFILE was produced and allocate a data set of the same type. This means that program objects shipped on RELFILEs can be processed with less pre-installation activity.

Note: Remember that PDSEs can only exist in an SMS environment.

During RECEIVE processing of SYSMODs packaged in RELFILE format, SMP/E uses the copy utility to copy the RELFILE data sets to partitioned SMPTLIB data sets. The utility that is used and allocation details for the SMPTLIBs are defined in the Global CSI under OPTIONs and DDDEFs within the Entry Type Selection panel. The DSNTYPE field is only displayed if you select the SMS option on the initial panel. It is specified on the SMS Options panel, but most often, you do not need to set it.

The REFILE data set may be:

- An unloaded PDS or PDSE data set on tape or DASD
- A PDS or PDSE data set on DASD

SMPTLIB data sets may be pre-allocated without SMP/E involvement or may be dynamically allocated by SMP/E during RECEIVE processing. SMP/E′s new ability to determine DSNTYPE enables it to distinguish between PDSs and PDSEs and to allocate them correctly based on the data set type of the RELFILE. This means that more of the organization and consolidation decisions that are usually taken during installation can be automated. This simplifies the steps that were previously performed by a variety of manual methods, which is becoming increasingly important as more program objects and extended object modules are distributed with products.

8.5.1 How DSNTYPE Is Determined

The selection of an appropriate DSNTYPE is critical when the RELFILE contains program objects or load modules. Program objects are created automatically when load modules are copied into a PDSE. They are converted back to load modules when they are copied into a PDS. However, some program objects cannot be converted into load modules because they use features of program objects that do not exist in load modules.

When program objects or load modules are in unloaded format, as they are in a RELFILE (Tape or DASD), IEBCOPY is unable to directly convert between program objects and load modules. This is because their unloaded formats differ. IEBCOPY is able to reload, but the designated reload data set must have the same organization as the original data set (PDS or Library). That is, an unloaded PDSE cannot be reloaded to a PDS (only to another PDSE) and an unloaded PDS cannot be reloaded to a PDSE (only to another PDS).

When RECEIVE processing allocates a new SMPTLIB data set, it uses the original DSNTYPE from the RELFILE data set. If this cannot be determined, SMP/E uses

the DSNTYPE specification in the SMPTLIB DDDEF entry. Otherwise, SMP/E uses a default specified by SMS or the operating system.

8.5.2 Diagnostic Checks

The following situations involving DSNTYPE are checked by SMP/E and require your attention and action.

- The SMPTLIB data set has been preallocated.

  RECEIVE processing checks the DSNTYPE of the preallocated data set against that of the RELFILE. If they are incompatible and the RELFILE contains ++MOD elements, new warning (GIM21500W) and informational (GIM22000I) messages are issued, but the process continues using the preallocated SMPTLIB data set.

```
GIM21500W PREALLOCATED SMPTLIB DATA SET dsname1 HAS A DSNTYPE
          OF dsntype1.
          THIS DIFFERS FROM THE DSNTYPE OF dsntype2 FOR THE
          CORRESPONDING RELFILE DATA SET dsname2

GIM22000I AN ERROR MAY OCCUR WHEN RELFILE DATA SET dsname1 IS COPIED
          TO THE SMPTLIB DATA SET dsname2.
```

- SMP/E cannot determine the original DSNTYPE.

  The SMPTLIB data set is allocated with the DSNTYPE value that is specified in the SMPTLIB DDDEF. If it is not specified there, the DSNTYPE text unit is not passed to dynamic allocation. In this situation, the system default or the SMS subsystem determines the DSNTYPE value.

- The original DSNTYPE of the RELFILE cannot be determined.

  When this occurs and the RELFILE contains ++MOD elements, SMP/E issues new warning message GIM21700W and new informational message GIM22000I. Processing continues.

```
GIM21700W SMP/E COULD NOT DETERMINE THE DSNTYPE OF RELFILE DATA SET
          dsname.

GIM22000I AN ERROR MAY OCCUR WHEN RELFILE DATA SET dsname1 IS COPIED
          TO THE SMPTLIB DATA SET dsname2.
```

- SMP/E cannot allocate a PDSE.

  This occurs when RECEIVE processing determines that the SMPTLIB data set should be allocated with a DSNTYPE of LIBRARY but the system does not support this type of data set.

```
GIM22500W SMP/E CANNOT ALLOCATE SMPTLIB DATA SET dsname WITH A
          DSNTYPE OF LIBRARY
```

8.6 Removal of Parmlib Dependencies

The GIMOPCDE member, which SMP/E previously used to determine valid OPCODES while processing JCLIN, has been removed from parmlib. Instead, a default set of OPCODE definitions is held internally within SMP/E. This eliminates the need for SMP/E to allocate parmlib whenever processing JCLIN assembler steps.

With this enhancement, during JCLIN processing, SMP/E checks for a new data set, SMPPARM. If it exists, a check is made for member GIMOPCDE. If this member exists, it is used to override SMP/E′s internal default values.

Your own member would also be placed in the SMPPARM data set. It would be accessed by the OPCODE operand of the JCLIN command or ++JCLIN statement.

8.7 Year 2000 Support

SMP/E′s date processing has been aligned with the capability to process dates into the 21st century. The main areas of change so that a 4-digit year can be specified are:

- Dates entered as input
  - Dialog
  - Command
- Tape and data set expiration dates
  - CBIPO and ServerPac dialogs

For 21st century date processing, a sliding window technique has been implemented to perform date comparisons for LIST LOG, REPORT ERRSYSMODS processing and in ordering the displays in CBIPO dialogs.

A window defines how a specific 2-digit year is evaluated and whether it is greater than or less than a particular year. A sliding window compares a 2-digit date with the current (system) date rather than a fixed point in time.

Note: For Year 2000 compatibility and migration considerations, see Appendix D, “SMP/E BUILDMCS Output Elements” on page 211.

Chapter 9. SDSF

SDSF is an efficient way to control JES2 job processing and devices. It allows you to monitor jobs while they are running and browse output without printing it. You can also browse the system logs, including the sysplex-wide operations log. SDSF does this with an interactive panel interface that provides immediate, up-to-date information about jobs, output, printers, initiators, and other system resources.

SDSF Release 1.6.0 is an optional feature of OS/390 Release 2 and it is the last release orderable outside of OS/390. Any further functional enhancements to SDSF are available only through OS/390. SDSF Release 1.6.0 provides the following enhancements:

- New displays

  New panels have been added to SDSF to enable the ability to initiate the spool offload function, and to display and manage JES2 lines and associated resources, and JES2 nodes.

- Date support

  The dates can be displayed in user definable formats. The year 2000 and beyond is recognized.

- Cancel confirmation

  When any of the destructive commands (Cancel, Purge, Restart, or Erase) are issued, the user can confirm the request prior to its execution.

- Column locate

  Users can locate a column in a tabular display by its title.

- Cursor positioning

  The cursor is returned to the last entry processed in a list.

- JCL Edit/Submit

  Users can issue a command that extracts an exact copy of the submitted JCL for a job. They can then edit the extracted data, and resubmit the job.

- SAF Control

  A SAF resource can be used to determine a user′s membership in groups defined by ISFGRP macros in ISFPARMS.

- Dynamic ISFPARMS

  Systems programmers can dynamically update ISFPARMS, reducing the effort associated with ISFPARMS management.

- OPERLOG

  Filtering has been added, making it easier to find specific information in the operlog in a sysplex environment.

- WHO command enhancements

  The WHO command has been enhanced to display information such as levels of MVS, JES, RMF, and ISPF.

9.1 Enhanced Commands and Displays

The changes to the user commands available in SDSF Release 1.6.0 are shown in Table 12.

Table 12. Summary of User Command Changes

Command	New/Changed	Description
FILTER	Changed	Changed to filter the operlog panel as well as tabular panels.
LI	New	Displays the LI panel, which shows JES2 NJE and RJE lines.
LOGLIM	New	Limits the amount of operlog data SDSF searches when filters are in effect.
LOCATE	Changed	Changed to locate column names, as well as line numbers, on tabular panels. Also accepts a 4-digit year on the log.
NO	New	Displays the NO panel, which shows JES2 NJE nodes.
NEXT	Changed	Changed to accept minutes and seconds as parameters when scrolling the OPERLOG display.
PREV	Changed	Changed to accept minutes and seconds as parameters when scrolling the OPERLOG display.
PRINT	Changed	Changed to accept PT as an abbreviation, and to accept a 4-digit year.
QUERY AUTH	New	Displays the commands the user is authorized for.
SELECT	Changed	Changed to override settings for PREFIX, OWNER, or DEST, if the user is authorized to use those commands.
SET CONFIRM	New	Sets confirmation of action characters on or off.
SET CURSOR	New	Controls placement of the cursor on tabular panels.
SET DATE	New	Sets the date format used on displays and commands.
SO	New	Displays the SO panel, which shows SPOOL offloaders and offload devices.
SJ action character	New	Displays just the JCL for a job and allows editing.
, overtype character	New	Erases an output descriptor on the JDS and OD panels.

9.1.1 New Primary Option Menu

SDSF primary option menu includes the LINE, NODE, and SO options. Figure 80 on page 153 is an example of the Primary Option Menu.

```
  Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
V1R6M0 ------------------  SDSF PRIMARY OPTION MENU  --------------------------
COMMAND INPUT ===>                                            SCROLL ===> PAGE

 Type an option or command and press Enter.

    LOG       - Display the system log
    DA        - Display active users in the sysplex
    I         - Display jobs in the JES2 input queue
    O         - Display jobs in the JES2 output queue
    H         - Display jobs in the JES2 held output queue
    ST        - Display status of jobs in the JES2 queues
    PR        - Display JES2 printers on this system
    INIT      - Display JES2 initiators on this system
    MAS       - Display JES2 members in the MAS
    LINE      - Display JES2 lines on this system
    NODE      - Display JES2 nodes on this system
    SO        - Display JES2 spool offload for this system
    ULOG      - Display user session log

    TUTOR     - Short course on SDSF (ISPF only)
PF 1=HELP      2=SPLIT     3=END       4=RETURN    5=IFIND     6=BOOK
PF 7=UP        8=DOWN      9=SWAP     10=LEFT     11=RIGHT    12=RETRIEVE
```

Figure 80. SDSF Primary Option Menu

9.1.2 LINES Display

SDSF shows the status of each JES2 NJE and RJE line, and its associated transmitters and receivers. It also shows information about the job currently being processed by a transmitter or receiver, such as the number of lines that have been processed. This information allows you to quickly monitor activity. With filtering, you can limit the number of devices that SDSF displays. With action characters, you can start, drain, and restart the devices. By typing over fields, you can change characteristics of the devices, such as:

- Duplex
- Line and page limits
- Resistance
- Speed
- Node the line is connected to

Figure 81 on page 154 is an example of the Lines display. Options were set to also display the action characters with descriptions.

```
  Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF LINE DISPLAY                                        LINE 1-19 (153)
COMMAND INPUT ===>                                          SCROLL ===> PAGE
ACTION=//-Block,=-Repeat,+-Extend,C-Cancel,D-Display,E-Restart,I-Interrupt,
ACTION=P-Stop,Q-Quiesce,S-Start
NP   DEVICE   STATUS   UNIT  NODE     JOBNAME  JOBID    OWNER    PROC-L
     LINE1    ACTIVE    SNA
     L1.JR1   INACTIVE
     L1.JT1   INACTIVE
     L1.SR1   INACTIVE
     L1.SR2   INACTIVE
     L1.SR3   INACTIVE
     L1.SR4   INACTIVE
     L1.SR5   INACTIVE
     L1.SR6   INACTIVE
     L1.SR7   INACTIVE
     L1.ST1   INACTIVE
     L1.ST2   INACTIVE
     L1.ST3   INACTIVE
     L1.ST4   INACTIVE
     L1.ST5   INACTIVE
     L1.ST6   INACTIVE
     L1.ST7   INACTIVE
```

Figure 81. SDSF Lines Display

9.1.3 NODE Display

You can see the status of each NJE node and modify its characteristics, such as what the node transmits and receives. With action characters, you can start node communication on a line, and display information about the network connections or paths for a node. JES2 $TNODE command functions are available to set:

- Node name
- Authority
- What the node receives and transmits to the local node
- Resistance rating of the connection

Figure 82 on page 155 is an example of the Nodes display. Options were set to also display the action characters with descriptions.

```
  Display  Filter  View  Print  Options  Help
--------------------------------------------------------------------------------
SDSF NODE DISPLAY  WTSCPLX1 N1                          LINE 1-19 (999) V   HOLD
COMMAND INPUT ===>                                        SCROLL ===> PAGE
ACTION=//-Block,=-Repeat,+-Extend,D-Display,S-Start
NP   NUMBER NODENAME STATUS               AUTHORITY         TRANS  RECV   HOLD
     N1     WTSCPLX1 OWNNODE             (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N2     WTSCMXA  VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N3     WTSCPLX3 VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N4     WTSCNET  VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N5     WTSCPOK  VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N6     WTSCTEST UNCONNECTED         (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N7     WTSC58   VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N8     WTSC59   VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N9     WTSC60   VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N10    WTSCICF  VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N11    WTSCESA  VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N12    WTSCSL2  VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N13    EPLMVSA  VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N14    BLDISDMC VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N15    WTSCVMXA VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N16    PLPSC    VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N17    KGNVMC   VIA SC47            (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
     N18    WTSCPLX9 UNCONNECTED         (D=Y,J=Y,N=N,S=Y) BOTH   BOTH   NONE
```

Figure 82. SDSF Nodes Display

9.1.4 Spool Offload Display

A Spool Offload display makes it easier to control the spool offload devices. SDSF displays the status of each offloader and its associated transmitters and receivers, and shows information about the jobs being processed. You can start, drain, and restart the devices. Simply typing over fields lets you set characteristics of the devices and control the jobs that are selected. Overtypeable fields allow JES2 $TOFFLOAD and $TOFF command functions, including:

- Data set name
- Modify values like burst, class, destination
- Selection values like burst, class, destination

Figure 83 on page 156 is an example of the Spool Offload display. Options were set to also display the action characters with descriptions.

```
  Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF SPOOL OFFLOAD DISPLAY                              LINE 1-19 (20)
COMMAND INPUT ===>                                        SCROLL ===> PAGE
ACTION=//-Block,=-Repeat,+-Extend,C-Cancel,D-Display,E-Restart,P-Stop,S-Start
NP   DEVICE   STATUS    TYPE     JOBNAME  JOBID    OWNER    PROC-LINES  TOT-LI
     OFFLOAD1 DRAINED
     OFF1.JT  STARTABLE
     OFF1.ST  STARTABLE
     OFF1.JR  STARTABLE
     OFF1.SR  STARTABLE
     OFFLOAD2 DRAINED
     OFF2.JT  STARTABLE
     OFF2.ST  STARTABLE
     OFF2.JR  STARTABLE
     OFF2.SR  STARTABLE
     OFFLOAD3 DRAINED
     OFF3.JT  STARTABLE
     OFF3.ST  STARTABLE
     OFF3.JR  STARTABLE
     OFF3.SR  STARTABLE
     OFFLOAD4 DRAINED
     OFF4.JT  STARTABLE
     OFF4.ST  STARTABLE
     OFF4.JR  STARTABLE
```

Figure 83. SDSF Spool Offload Display

9.1.5 Filtering on the Operlog Display

Filtering on the Operlog display allows users to limit the amount of operlog data displayed. This is an enhancement to the FILTER function. Filters can be set based on:

- System name
- Date and time
- Jobname and job ID
- Console name
- Message ID and message text

Figure 84 on page 157 is an example of the FILTER pop-up. The defined filter only displays messages for system SC54. Depending on the number of operlog messages, filtering on a certain system may require many messages to be read. Response times are therefore difficult to predict. If you want to display the operlog of just one system, it may be faster to use the SYSLOG function on that system. Alternatively, you can use the LOGLIM command to limit the amount of operlog messages SDSF searches.

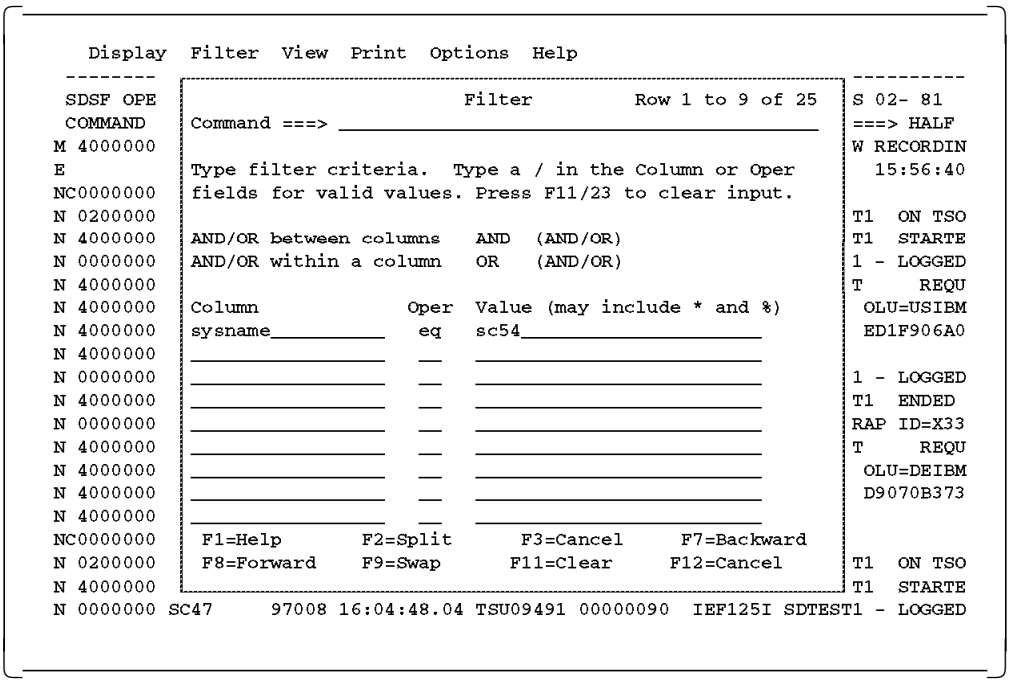


Figure 84. FILTER Pop-Up

9.1.6 WHO Command

The WHO command and the pop-up display additional information about the SDSF server and the level of software. Figure 85 is an example of the WHO command response.

```
  Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
V1R6M0 ------------------  SDSF PRIMARY OPTION MENU  --------------------------
COMMAND INPUT ===> who                                       SCROLL ===> PAGE
USERID=TSOUSER,PROC=IKJACCNT,TERMINAL=SCT3056C,GRPINDEX=1,GRPNAME=ISFSPROG,
MVS=OS/390 01.02.00,JES2=OS 1.1.0,JESNAME=JES2,ISPF=4.2,RMF/DA=6.0.2,
SERVER=YES,SERVERNAME=SDSF
    LOG       - Display the system log
    DA        - Display active users in the sysplex
    I         - Display jobs in the JES2 input queue
    O         - Display jobs in the JES2 output queue
    H         - Display jobs in the JES2 held output queue
    ST        - Display status of jobs in the JES2 queues
    PR        - Display JES2 printers on this system
    INIT      - Display JES2 initiators on this system
    MAS       - Display JES2 members in the MAS
    LINE      - Display JES2 lines on this system
    NODE      - Display JES2 nodes on this system
    SO        - Display JES2 spool offload for this system
    ULOG      - Display user session log

    TUTOR     - Short course on SDSF (ISPF only)
    END       - Exit SDSF
```

Figure 85. WHO Pop-Up

9.1.7 Editing of the JCL for a JOB

SJ action character allows users to edit just the JCL for a job. A job′s JCL can be edited and resubmitted.

This is a useful enhancement, because it allows editing of a job′s JCL without leaving SDSF. This is a very fast way to change JCL for a job which was, for example, created by a long dialog like SMP/E or ISMF. Now small JCL errors can be fixed in flight. The changes are not saved when you exit.

Figure 86 is an example of how to issue the action characters. Options were set to also display action characters with descriptions.

```
  Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF HELD OUTPUT DISPLAY ALL CLASSES  LINES 656,080     DATA SET DISPLAYED
COMMAND INPUT ===>                                          SCROLL ===> PAGE
ACTION=//-Block,=-Repeat,+-Extend,?-JDS,A-Release,C-Cancel,H-Hold,L-List,
ACTION=O-Release,P-Purge,Q-Outdesc,S-Browse,X-Print
NP   JOBNAME  JOBID    OWNER    PRTY C ODISP DEST                TOT-REC  TOT-
     HAIMO1   JOB21425 DONNAS     96 T HOLD  LOCAL                 4,333
     HAIMOA   JOB25615 HAIMO      16 T HOLD  LOCAL                38,767
     HAIMOD   JOB25631 HAIMO     144 T HOLD  LOCAL                   330
     HAIMOE   JOB25694 HAIMO     144 T HOLD  LOCAL                   348
     HAIMOG   JOB25703 HAIMO     144 T HOLD  LOCAL                   261
     HAIMOI   JOB25722 HAIMO     144 T HOLD  LOCAL                   348
     HAIMOJ   JOB25725 HAIMO     128 T HOLD  LOCAL                   615
     HAIMOK   JOB25727 HAIMO     144 T HOLD  LOCAL                   198
     HAIMOL   JOB25728 HAIMO     144 T HOLD  LOCAL                   150
sj   HAIMOM   JOB25729 HAIMO     144 T HOLD  LOCAL                   300
     HAIMON   JOB25736 HAIMO     144 T HOLD  LOCAL                   321
     HAIMOO   JOB25737 HAIMO     144 T HOLD  LOCAL                   297
     HAIMOP   JOB25740 HAIMO     144 T HOLD  LOCAL                   293
     HAIMOQ   JOB25748 HAIMO     128 T HOLD  LOCAL                   783
     HAIMOR   JOB25751 HAIMO     144 T HOLD  LOCAL                   160
     HAIMOU   JOB25777 HAIMO       0 T HOLD  LOCAL                41,707
     HAIMOV   JOB25785 HAIMO     144 T HOLD  LOCAL                   219
```

Figure 86. SJ Action Characters

Figure 87 on page 159 is an example of the JCLEDIT panel, which allows editing of the JCL.

```
SDSF EDIT    HAIMOM   (JOB25729) JCLEDIT                  Columns 00001 00072
Command ===>                                               Scroll ===> PAGE
****** ***************************** Top of Data ******************************
000001 //HAIMOM JOB (999,POK),'BOB HAIMOWITZ',CLASS=A,MSGCLASS=T,
000002 // MSGLEVEL=(1,1),NOTIFY=&SYSUID,REGION=0M,TIME=1440
000003 /*JOBPARM L=999,SYSAFF=SC47
000004 //*
000005 //S1       EXEC PGM=GIMSMP,
000006 //         PARM='PROCESS=WAIT',
000007 //         DYNAMNBR=120
000008 //*
000009 //* NOTE:      THIS JCL CREATED BY THE COMMAND GENERATION DIALOGS.
000010 //*
000011 //*            SMP ZONE-RELATED FILES ARE DYNAMICALLY ALLOCATED,
000012 //*            THIS INCLUDES THE SMPPTS, SMPLOG, AND SMPTLIB DATA SETS,
000013 //*            IF APPLICABLE.
000014 //*
000015 //* SMP FILES
000016 //*
000017 //SMPCSI   DD DISP=SHR,DSN=TOT.SMP.GLOBAL.CSI
000018 //*
000019 //*
000020 //SMPCNTL  DD *
000021   SET    BOUNDARY (TGTM02A)
```

Figure 87. SJ JCL Edit Display

9.1.8 Action Character Confirmation

A confirmation option keeps you from accidentally destroying data. The confirmation pop-up is displayed for all destructive action characters: cancel, purge, and restart.

This option can be controlled either through ISFGRP definition or by using the SET CONFIRM ON|OFF command. Figure 88 on page 160 is an example of the confirmation pop-up.

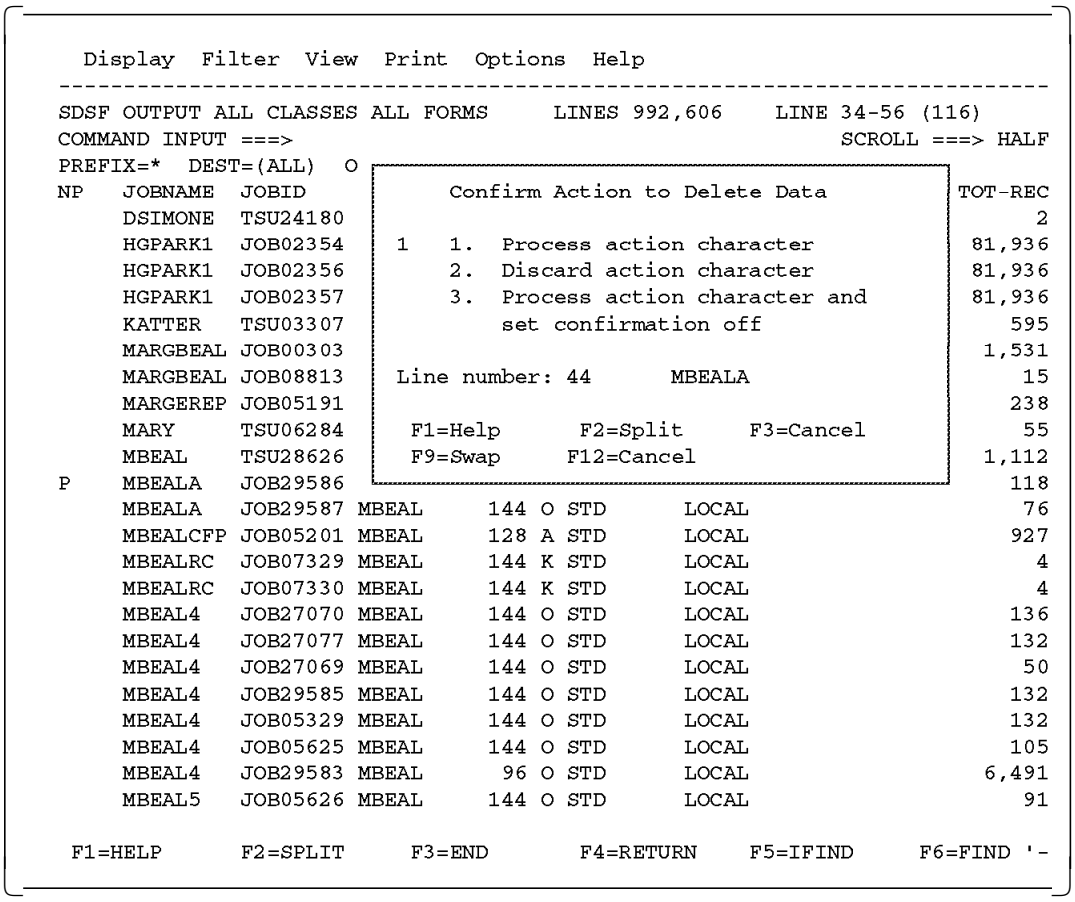


Figure 88. Action Character Confirmation Pop-Up

9.1.9 Other Enhancements

SDSF Release 1.6.0 contains several other enhancements:

- SET CURSOR ON, entered on the command line, makes the cursor return to the current line instead of jumping to the command line.
- QUERY AUTH command displays the SDSF commands you are authorized to use. Only commands requiring authorization are displayed.
- On tabular panels, you can scroll the panel directly to a specific line or column.
- Output descriptors are now overtypeable also on the Job Data Set Panel. (Previously they were overtypeable only on the Output Descriptors panel.)
- Year 2000 is now supported. Commands accept 4-digit years, and dates are displayed using 4-digit years.

9.2 SDSF Installation

Depending on the Customized Offering you ordered, the installation of SDSF is different. You should use the appropriate SDSF Program Directory or Installation Guide to install the product. The following post-installation steps are required:

1. Authorize SDSF load libraries:

   SISFLOAD

SISFLINK

SISFLPA

2. Add SISFLPA to Link Pack Area.
3. Update execution environment.

   SDSF can execute from the link list, link pack area, or steplib.
4. Update TSO procedures.

   Add the panels, messages, and tables libraries into your TSO Logon sequence.
5. Update ISPF Primary Options Menu to invoke SDSF.
6. Enable RMF for sysplex DA.

   If you are running sysplex and RMF, you can use the sysplex DA function in SDSF. Make the modules in SISFLOAD accessible to the RMF started task.
7. Optionally, define the SDSF Server.
8. Copy the softcopy books to sequential data sets.
9. Run the UCLIN jobs.

   Sample UCLIN jobs cause SMP/E to automatically reassemble affected SDSF modules when JES2 maintenance is applied.
10. Verify the installation.

After you have run the post-installation steps and verified the basic functions of SDSF, you can migrate SDSF to meet the authorization and security requirements of your installation.

9.3 Migrating to SDSF Release 1.6.0

SDSF Release 1.6.0 introduces new functions that may impact your installation:

- ISFPARMS statements provide an alternative to ISFPARMS assembler macros. They eliminate the need to reassemble and link edit ISFPARMS when you make changes. You can convert your existing ISFPARMS to the new statements with the ISFACP utility provided by SDSF.
- Security Authorization Facility (SAF) can be used to determine a user's membership in groups defined with ISFPARMS.

When you migrate from an earlier SDSF release, you may need to make changes in ISFPARMS. SDSF Release 1.6.0 adds parameters to the ISFPMAC, ISFGRP, and ISFFLD macros.

Important

You must reassemble your modified ISFPARMS module using the SDSF Release 1.6.0 macro library. The ISFPARMS load module for SDSF Release 1.6.0 is incompatible with earlier releases of SDSF.

If you are using SAF Resource Profiles for security, SAF resources defined for previous releases continue to work. However, you should review your SAF resource definitions if there are new resources you want to protect.

If you have written exit routines, review them to ensure they are still appropriate for the SDSF Release 1.6.0 environment. You must reassemble them with the SDSF Release 1.6.0 macro library.

The SVC in SDSF Release 1.6.0 is compatible with the SVC installed by prior releases of SDSF. If you plan on running different levels of SDSF in the same system, you should use the SDSF Release 1.6.0 level of the SVC.

The HASPINDX data set can be shared across multiple systems, provided all the following are true:

- All systems are running SDSF Release 1.6.0.
- The systems are in a sysplex.
- JES2 levels are the same for all members in the MAS. If you are running a mixed environment, the HASPINDX data set may be shared between any combination of JES2 5.1.0, JES2 5.2.0, or JES2 OS/390 1.1 or 1.2 running in the same MAS.

9.3.1 Using SAF for Security

The Security Authorization Facility (SAF) interface is an alternative to using ISFPARMS to provide security for SDSF. SAF is part of the OS/390 operating system and is always present. SDSF uses the SAF interface to route authorization requests to the external security manager. The benefits of using the SAF interface for SDSF security are:

- Dynamic change of security profiles
- Single image of security information
- Simple introduction of security philosophy
- Improved auditability
- Improved protection

In order to use SAF security, you need a security product. OS/390 Release 2 has an optional Security Server feature that contains the RACF component.

9.3.1.1 Relationship of SAF and ISFPARMS

You may choose to use SAF to protect some functions while using ISFPARMS to protect others. Even if you use SAF for all SDSF security, you need ISFPARMS to control:

- Global values (ISFPMAC macro or OPTIONS statement).
- Any values for groups that are not related to security (ISFGRP macro or GROUP statement).
- Code page (ISFTR macro or TRTAB statement).
- If you want to customize the columns on SDSF panels, you also need ISFFLD macros or FLD statements.

ISFPARMS are used to determine authorization when SAF returns an indeterminate result, that is, when SAF cannot make a security decision. SAF returns an indeterminate result when the resource class is inactive or no profiles are defined to protect a resource. When using ISFPARMS to back up a SAF security scheme, you should have at least two groups defined: one for all users except the system programmer, and another for the system programmer.

For information on converting to SAF security, see *System Display and Search Facility Customization and Security*.

9.3.2 Summary of Macro Changes for SDSF Release 1.6.0

SDSF Release 1.6.0 adds parameters to the ISFPMAC, ISFGRP, and ISFFLD macros. It also adds SAF resources and commands. The changes to the ISFPMAC macro for SDSF Release 1.6.0 are shown in Table 13.

Table 13. ISFPMAC Change Summary

Keyword	Default	New/Changed	Description
LOGLIM	0	New	Sets a default limit for searching the operlog when filters are in effect.
SERVER	SDSF	New	Identifies the default server to be used by SDSF for use with ISFPARMS.

The changes to the ISFGRP macro for SDSF Release 1.6.0 are shown in Table 14.

Table 14. ISFGRP Change Summary

Keyword	Default	New/Changed	Description
label		Changed	Used to SAF-check membership in the group.
AUTH		Changed	LI, SO, and NO added.
CONFIRM	ON	New	Controls action character confirmation.
CURSOR	ON	New	Controls how SDSF places the cursor on tabular panels.
DATE	MMDDYYYY	New	Sets the date format used by SDSF.
LINEFLDS		New	Names on the ISFFLD macro that define the primary field list for the LI panel.
LINEFLD2		New	Names on the ISFFLD macro that define the alternate field list for the LI panel.
NODEFLDS		New	Names on the ISFFLD macro that define the primary field list for the NO panel.
NODEFLD2		New	Names on the ISFFLD macro that define the alternate field list for the NO panel.
SOFLDS		New	Names on the ISFFLD macro that define the primary field list for the SO panel.
SOFLD2		New	Names on the ISFFLD macro that define the alternate field list for the SO panel.

The ISFFLD macro is changed to support the new columns on the LI, NO, and SO displays.

The ISFTR macro is changed to support the new code pages.

As part of the support for the year 2000, several date columns on SDSF panels were expanded.

A new SAF resource, GROUP.group-name.server-name, is added to control membership in groups defined in with ISFGRP macros in ISFPARMS. Also, new resources are added to control the use of the new panels and the fields of the panels.

9.3.3 Defining ISFPARMS

ISFPARMS defines global options, the functions each user is authorized to perform, and the format of the panels.

ISFPARMS is required, but you can use SAF to control user authorization. If you use SAF for user authorization, you need the ISFPARMS module only to define your global options and panel formats. You might also use ISFPARMS as a backup to SAF.

9.3.3.1 ISFPARMS Migration

ISFPARMS contains sections that define global options, group membership, panel definitions, and code page definitions. SDSF provides two alternatives for ISFPARMS:

- Assembler macros that you define, assemble, and then link into the SDSF load library
- Statements that are introduced with SDSF Release 1.6.0

ISFPARMS statements are easier to code and are more dynamic than the assembler macros. They can be updated without reassembling or link-editing. The statements are processed by an SDSF server, which is controlled by MVS operator commands. SDSF provides a utility for converting ISFPARMS assembler macros to ISFPARMS statements.

To assist you in defining your ISFPARMS, SDSF provides a sample ISFPARMS in assembler macros and two sample ISFPARMS in statements. You can modify the appropriate sample to meet the needs of your installation. Table 15 summarizes ISFPARMS.

Table 15 (Page 1 of 2). ISFPARMS Summary

Assembler Macro	Statement	Required	Description
ISFPMAC	OPTIONS	Assembler only	Specifies global SDSF initialization parameters. If you use statements, you need the ISFPMAC macro to identify the default server.
ISFGRP	GROUP	No	Defines a group of users and the SDSF functions that are available to a member of the group. Also includes initialization parameters. You can use SAF along with your group definitions to control membership and authorization.

Table 15 (Page 2 of 2). ISFPARMS Summary			
ISFFLD	FLD + FLDENT	No	Customizes the fields shown on an SDSF primary or alternate panel for members of a group. Associated with an ISFGRP macro or GROUP statement.
ISFNTBL	NTBL + NTBLENT	No	Specifies such things as user IDs, job names, and destination names to further qualify group membership and authority. Associated with an ISFGRP macro or GROUP statement.
ISFTR	TRTAB + TRDEF	Assembler only	Specifies the code page that SDSF uses for members of the group. Associated with an ISFGRP macro or GROUP statement.

If you decide to use the ISFPARMS module, you have to define the macros, and assemble and link-edit it as you did with the previous SDSF releases.

However, the ISFPARMS statements can be updated without reassembling and link-editing. Using ISFPARMS statements is the recommended implementation method to define ISFPARMS.

9.3.3.2 Converting ISFPARMS Assembler Macros to Statements

SDSF provides a conversion utility. You invoke the utility by typing the ISFACP command on the ISPF command line. ISFACP is located in SDSF's SISFEXEC data set. The utility displays a panel on which you type the names of the input data set (ISFPARMS assembler macros) and output data set (statements), as well as the assembler macro library you use when assembling SDSF.

This is the conversion utility Pop-Up.

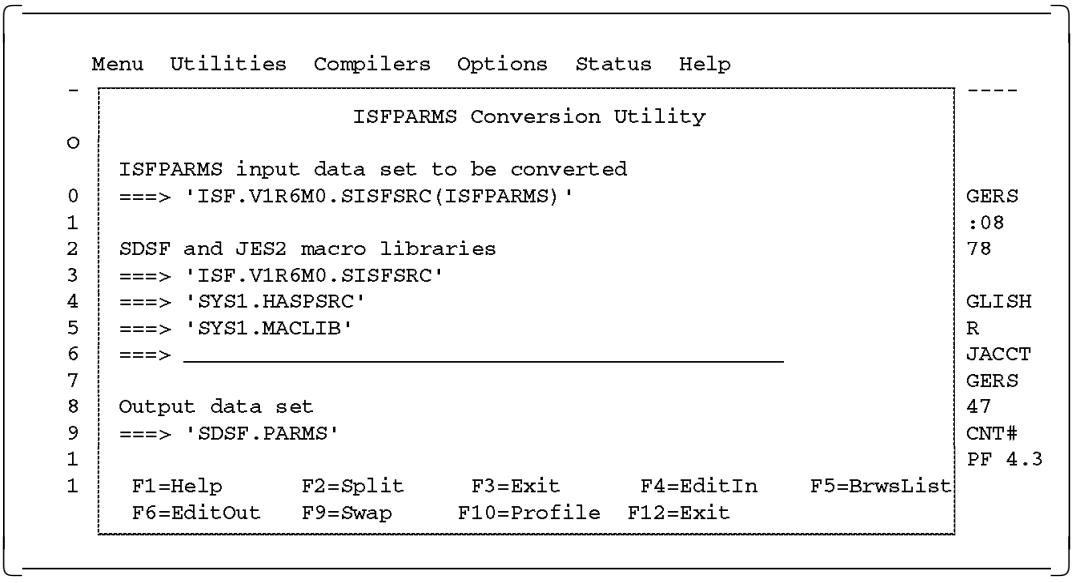


Figure 89. Conversion Utility Pop-Up

Because the macros in ISFPARMS can be coded in different ways, the conversion may not be exact. You may need to make corrections. For example, the name of the SDSF help panel data set may need an update to refer to OS/390

Release 2. Also, the format of the date fields can be set to either MMDDYYYY or DDMMYYYY.

9.3.3.3 SDSF Server

SDSF uses a new address space called the SDSF server to process the statements. You control the server through the MVS operator START, STOP, and MODIFY commands.

SDSF reads the statements from concatenated SYS1.PARMLIB or, optionally, from a member of a partitioned data set allocated to the server when the server is started.

The server offers a test mode, which allows you to check the syntax of the statements without activating them. MODIFY SDSF,REFRESH,TEST is the command that allows you to do just the syntax checking without putting the statements into effect.

The default server name used by SDSF is defined in the ISFPMAC macro of ISFPARMS, allowing you to restrict use of a particular server. This may be desirable in a test environment, or when SDSF maintenance is applied.

A server supports a single ISFPARMS. To run multiple levels of ISFPARMS, you must have a unique server for each level. New parameters can be activated at any time with the operator MODIFY command directed to the SDSF server. As with ISFPARMS, changes take effect the next time users access SDSF.

When you use statements, SDSF uses the ISFPARMS defined with assembler macros only for the name of the default server, specified on the SERVER parameter of the ISFPMAC macro. SDSF is connected to the server during initialization and uses the server to process the ISFPARMS defined with statements. However, if the server is not active or if no statements are in effect, SDSF reverts to the ISFPARMS defined with the assembler macros and initialization proceeds.

If you have migrated to an ISFPARMS defined with statements, you may want to restrict or prevent SDSF from reverting to the ISFPARMS defined with assembler macros. That process is described in *System Display and Search Facility Customization and Security*. The book also describes methods that can be used to protect your SDSF server.

After you have created your ISFPARMS statements, you should check the server procedure in proclib and its performance definitions. You should either include the start command of the server in the COMMNDxx member of the parmlib, or include the server in your system automation procedures. Then you can start your server and start to use SDSF.

Chapter 10. RMF Enhancements in OS/390 Release 2

OS/390 Release 2 incorporates changes to all monitors of RMF. The enhancements provide:

New functions The cache RMF reporter is now integrated into RMF. A new report displays the activity of a kernel of MVS OpenEdition.

New facilities From a MONITOR II session, it is possible to switch to any system in the sysplex.

Sysplex control New fields in the Coupling Facility Activity Report provide information on delays.

New reports Two new reports have been added to Monitor III (Channel and IOQUEUE).

10.1 New Functions in RMF

Two new sets of reports have been added to track the following:

- Cache Subsystem Activity
- OMVS Kernel Activity

The Coupling Facility report has been enhanced.

10.1.1 Cache Subsystem Activity Report

The Cache Subsystem Activity report integrates into RMF the functions previously included in the Cache RMF Reporter (CRR) product. Data is gathered in SMF record type 74.5 instead of record type 245.

Users not familiar with CRR are advised to use the Cache Subsystem Device Overview report (see Figure 91 on page 169) as a quick assessment of cache-related performance problems.

RMF reporting is at the volume level. Data set level caching and service time information for SMS data sets can be found in SMF records 42 subtype 6.

10.1.1.1 Data Gathering

One of the following options must be specified to request or unselect the gathering of cache data.

CACHE Monitor I gathers data for this report, with the default option CACHE, as SMF record type 74.5.

NOCACHE If you want to suppress gathering, you have to specify option NOCACHE.

10.1.1.2 Producing Reports

To produce a report, specify:

```
REPORTS(CACHE(options))
```

The options are:

SUBSYS This generates a Cache Subsystem Activity report with three sections:

- Cache Subsystem Status
- Cache Subsystem Overview
- Cache Subsystem Device Overview

This report is shown in Figure 90 and Figure 91 on page 169.

DEVICE This generates a Device Level report with two sections:

- Cache Device Status
- Cache Device Activity

```
                                          C A C H E   S U B S Y S T E M   A C T I V I T Y
                                                                                                                        PAGE    1

            OS/390                  SYSTEM ID SYS1            DATE  04/15/1996       INTERVAL  12.00.000
            REL 01.02.00            RPT VERSION 1.2.0         TIME  08.12.00

SUBSYSTEM  3990-03     CU-ID  02A0     SID  0010     CDATE  04/15/1996     CTIME 08.12.00      CINT  12.00

------------------------------------------------------------------------------------------------------------------------------------
                                                       CACHE SUBSYSTEM STATUS
------------------------------------------------------------------------------------------------------------------------------------

SUBSYSTEM STORAGE           NON-VOLATILE STORAGE        STATUS

CONFIGURED       64.0M      CONFIGURED        4.0M      CACHING                  - ACTIVE
AVAILABLE        63.5M      PINNED            0.0       NON-VOLATILE STORAGE     - ACTIVE
PINNED            0.0                                   CACHE FAST WRITE         - ACTIVE
OFFLINE           0.0                                   IML DEVICE AVAILABLE     - YES

------------------------------------------------------------------------------------------------------------------------------------
                                                      CACHE SUBSYSTEM OVERVIEW
------------------------------------------------------------------------------------------------------------------------------------

TOTAL I/O    48411   CACHE I/O    48408   CACHE OFFLINE        0
TOTAL H/R    0.955   CACHE H/R    0.955


CACHE I/O    -------------READ I/O REQUESTS-------------    ----------------------WRITE I/O REQUESTS----------------------      %
REQUESTS       COUNT     RATE      HITS     RATE     H/R      COUNT     RATE      FAST     RATE      HITS     RATE     H/R    READ

NORMAL         33731     46.8     31868     44.3   0.945      11669     16.2     11669     16.2     11368     15.8   0.974    74.3
SEQUENTIAL      1427      2.0      1426      2.0   0.999       1581      2.2      1581      2.2      1547      2.1   0.978    47.4
CFW DATA           0      0.0         0      0.0     N/A          0      0.0         0      0.0         0      0.0     N/A     N/A

TOTAL          35158     48.8     33294     46.2   0.947      13250     18.4     13250     18.4     12915     17.9   0.975    72.6


--------------------CACHE MISSES------------------------          ------------MISC------------        ------NON-CACHE I/O--- -
REQUESTS      READ    RATE   WRITE    RATE  TRACKS    RATE                        COUNT   RATE                       COUNT   RATE
                                                                  DFW BYPASS        117    0.2        ICL                 0    0.0
NORMAL        1863     2.6     301     0.4    2072     2.9        CFW BYPASS          0    0.0        BYPASS              3    0.0
SEQUENTIAL       1     0.0      34     0.0    3000     4.2        DFW INHIBIT         0    0.0        TOTAL               3    0.0
CFW DATA         0     0.0       0     0.0                        ASYNC (TRKS)     1887    2.6

TOTAL         2199    RATE     3.1


---CKD STATISTICS---     ---RECORD CACHING---

WRITE            1229     READ MISSES        0
WRITE HITS       1219     WRITE PROM         0
```

Figure 90. Cache Subsystem Activity Report: Status and Overview

The Device Level report provides detailed information for each single device attached to the selected control unit. The status section shows whether caching and DASD fast write are active, or whether the current device is part of a duplex pair. The report helps you to analyze cache usage in detail on the basis of the information about the applications that access these volumes.

```
                                        C A C H E   S U B S Y S T E M   A C T I V I T Y
                                                                                                                        PAGE    2

          OS/390                   SYSTEM ID SYS1             DATE  04/15/1996       INTERVAL  12.00.000
          REL 01.02.00             RPT VERSION 1.2.0          TIME  08.12.00

SUBSYSTEM  3990-03     CU-ID  02A0     SID  0010     CDATE  04/15/1996     CTIME 08.12.00      CINT  12.00

------------------------------------------------------------------------------------------------------------------------------------
                                                     CACHE SUBSYSTEM DEVICE OVERVIEW
------------------------------------------------------------------------------------------------------------------------------------

VOLUME     DEV  DUAL     %     I/O   ---CACHE HIT RATE--   ----------DASD I/O RATE----------  ASYNC   TOTAL    READ   WRITE      %
SERIAL     NUM  COPY   I/O    RATE    READ     DFW    CFW   STAGE  DFWBP    ICL    BYP  OTHER   RATE     H/R     H/R     H/R   READ

*ALL                 100.0    67.2    46.2    17.9    0.0     2.9    0.2    0.0    0.0    0.0    2.6   0.955   0.947   0.975   7|.6
*CACHE-OFF             0.0     0.0
*CACHE               100.0    67.2    46.2    17.9    0.0     2.9    0.2    0.0    0.0    0.0    2.6   0.955   0.947   0.975   7|.6
TS3399    02A0         0.0     0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0   0.625   0.625     N/A  100.0
SY3304    02A1        21.3    14.3    14.3     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0   1.000   1.000   1.000  100.0
SY3302    02A2         3.7     2.5     2.3     0.1    0.0     0.2    0.0    0.0    0.0    0.0    0.0   0.938   0.936   1.000   9|.1
RZ3301    02A3         6.2     4.1     2.4     1.6    0.0     0.2    0.0    0.0    0.0    0.0    0.1   0.957   0.954   0.962   5|.9
DB3303    02A4         0.0     0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0   1.000   1.000     N/A  100.0
DB3304    02A5         0.0     0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0   0.250   0.250     N/A  100.0
DB3305    02A6         0.0     0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0   0.800   0.800     N/A  100.0
DB3306    02A7         0.0     0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0   0.250   0.250     N/A  100.0
RZ3313    02A8         0.1     0.1     0.0     0.1    0.0     0.0    0.0    0.0    0.0    0.0    0.0   0.917   0.615   1.000   2|.7
RZ3312    02A9         3.8     2.5     0.9     1.1    0.0     0.6    0.0    0.0    0.0    0.0    0.4   0.781   0.621   0.987   5|.4
RZ3311    02AA         0.0     0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0   0.750   0.750     N/A  100.0
RZ3310    02AB         2.3     1.6     1.0     0.5    0.0     0.0    0.0    0.0    0.0    0.0    0.1   0.968   0.970   0.964   6|.1
DL3303    02AC         0.0     0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0   0.571   0.571     N/A  100.0
DL3304    02AD         0.0     0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0     N/A     N/A     N/A    N/A
SY3301    02AE         0.0     0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0   0.400   0.400     N/A  100.0
TS3392    02AF         0.1     0.1     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0   0.708   0.667   0.889   8|.2
RZ3384    02B4         6.4     4.3     2.3     1.3    0.0     0.6    0.0    0.0    0.0    0.0    0.6   0.845   0.787   0.968   6|.6
RZ3380    02B5         0.8     0.5     0.2     0.3    0.0     0.0    0.0    0.0    0.0    0.0    0.0   0.906   0.843   0.941   3|.1
RZ3381    02B6        18.1    12.2     4.2     7.4    0.0     0.5    0.1    0.0    0.0    0.0    0.7   0.953   0.915   0.976   3|.6
RZ3391    02B7         8.3     5.6     4.3     1.2    0.0     0.1    0.0    0.0    0.0    0.0    0.1   0.976   0.970   0.998   7|.6
DB3307    02B8         0.0     0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0   1.000   1.000     N/A  100.0
RZ3393    02B9         7.7     5.2     2.6     2.2    0.0     0.4    0.0    0.0    0.0    0.0    0.5   0.925   0.896   0.962   5|.2
DB3308    02BA         0.0     0.0     0.0     0.0    0.0     0.0    0.0    0.0    0.0    0.0    0.0   0.400   0.400     N/A  100.0
RZ3392    02BB        21.1    14.2    11.8     2.1    0.0     0.2    0.0    0.0    0.0    0.0    0.2   0.982   0.982   0.984   8|.9
```

Figure 91. Cache Subsystem Activity Report: Device-Level Overview

```
                                C A C H E   S U B S Y S T E M   A C T I V I T Y
                                                                                                                    PAGE    21

           OS/390                  SYSTEM ID SYS1             DATE  04/15/1996       INTERVAL  12.00.000
           REL 01.02.00            RPT VERSION 1.2.0          TIME  08.12.00

SUBSYSTEM  3990-03     CU-ID  02A0     SID  0010     CDATE  04/15/1996     CTIME 08.12.00      CINT  12.00
VOLSER     RZ3381      NUM    02B6
-------------------------------------------------------------------------------------------------------------------------------
                                                CACHE DEVICE STATUS
-------------------------------------------------------------------------------------------------------------------------------

CACHE STATUS                                    DUPLEX PAIR STATUS

CACHING             - ACTIVE                    DUPLEX PAIR        - NOT ESTABLISHED
DASD FAST WRITE     - ACTIVE                    STATUS             - N/A
PINNED DATA         - NONE                      DUAL COPY VOLUME   - N/A

-------------------------------------------------------------------------------------------------------------------------------
                                                CACHE DEVICE ACTIVITY
-------------------------------------------------------------------------------------------------------------------------------

TOTAL I/O     8772   CACHE I/O     8772   CACHE OFFLINE   N/A
TOTAL H/R    0.953   CACHE H/R    0.953


CACHE I/O    -------------READ I/O REQUESTS-------------    ----------------------WRITE I/O REQUESTS----------------------   %READ
REQUESTS       COUNT     RATE      HITS     RATE     H/R      COUNT     RATE      FAST     RATE      HITS     RATE     H/R

NORMAL          1872      2.6      1594      2.2   0.851       4136      5.7      4136      5.7       4038     5.6   0.976    31.1
SEQUENTIAL      1425      2.0      1424      2.0   0.999       1339      1.9      1339      1.9       1308     1.8   0.977    51.6
CFW DATA           0      0.0         0      0.0     N/A          0      0.0         0      0.0          0     0.0     N/A     N/A

TOTAL           3297      4.6      3018      4.2   0.915       5475      7.6      5475      7.6       5346     7.4   0.976    37.6


----------------------CACHE MISSES-------------------------           ------------MISC------------        ------NON-CACHE I/O-----
REQUESTS      READ    RATE   WRITE    RATE  TRACKS    RATE                          COUNT   RATE                    COUNT   RATE
                                                                      DFW BYPASS         52    0.1        ICL            0     .0
NORMAL         278     0.4      98     0.1     374     0.5           CFW BYPASS          0    0.0        BYPASS         0     .0
SEQUENTIAL       1     0.0      31     0.0    3000     4.2           DFW INHIBIT         0    0.0        TOTAL          0     .0
CFW DATA         0     0.0       0     0.0                          ASYNC (TRKS)      509    0.7

TOTAL          408    RATE     0.6


---CKD STATISTICS---     ---RECORD CACHING---

WRITE           1048     READ MISSES        0
WRITE HITS      1041     WRITE PROM         0
```

Figure 92. Cache Subsystem Activity Report: Device-Level Reporting

```
                                C A C H E   S U B S Y S T E M   A C T I V I T Y
                                                                                                                    PAGE    01

           OS/390                  SYSTEM ID SYS1             DATE  04/15/1996       INTERVAL  12.00.000
           REL 01.02.00            RPT VERSION 1.2.0          TIME  07.12.00

SUBSYSTEM  3990-03     CU-ID  02A0     SID  0016     CDATE  04/15/1996    CTIME  07.12.00     CINT  12.00
VOLSER     RZ3381      NUM    02B6
-------------------------------------------------------------------------------------------------------------------------------
                                                CACHE DEVICE STATUS
-------------------------------------------------------------------------------------------------------------------------------

CACHE STATUS                                    DUPLEX PAIR STATUS

CACHING             - DEACTIVATED               DUPLEX PAIR NOT ESTABLISHED
DASD FAST WRITE     - DEACTIVATED
PINNED DATA         - NONE

-------------------------------------------------------------------------------------------------------------------------------
                                                CACHE DEVICE ACTIVITY
-------------------------------------------------------------------------------------------------------------------------------

TOTAL I/O     ---     CACHE I/O     ---     CACHE OFFLINE     8772
TOTAL H/R     ---     CACHE H/R     ---
```

Figure 93. Cache Subsystem Activity Report: Cache Offline

Note: Cache statistics are collected from the DASD control units, so it is redundant to collect the information on every MVS image. This applies not only to Parallel Sysplex, but also to any shared DASD environment. The only

precautions you must take are to be sure that the MVS collecting image has a path to all control units and is up during the whole measurement time period.

10.2 Coupling Facility Report Enhancements

The following enhancements are introduced in the Coupling Facility Report:

- Activity against cache structures
- Availability of logical processors
- Reasons for delays on requests to the coupling facility

To request the Coupling Facility Report, Monitor III gathers data for this report automatically. If you want to suppress gathering, you have to disable writing SMF record type 74.4. To produce this report, specify:

```
SYSRPTS(CF)
```

Note: The Coupling Facility Report is only available as an interval report, not as a duration report.

Figure 94 shows the changes to the Coupling Facility Report as follows:

A Indicates the number of logical processors defined for the coupling facility.

B Indicates the number of effective available logical processors in a shared environment. This value is only useful in a CFCC environment. The CFCC measures the time of real command execution as well as the the time waiting for work. The reported value shows the ratio between the LPAR dispatch time (CFCC execute and wait time) and the RMF interval length.

For example, if a CFCC CEC contains six LPs, and the measured CF LPAR has two logical processors and is limited to 5%, the number of effective LPs is 0.3.

Note: Refer to the CPU Activity report when using an ICMF LPAR.

```
                                C O U P L I N G   F A C I L I T Y   A C T I V I T Y
                                                                                                                                PAGE   1
     OS/390                  SYSPLEX UTCPLXJ8           DATE 04/11/1996          INTERVAL 030.00.000
     REL. 01.02.00           RPT VERSION 1.2.0          TIME 12.00.00            CYCLE 01.000 SECONDS

-------------------------------------------------------------------------------------------------------------------------------
 COUPLING FACILITY NAME = CF1
 TOTAL SAMPLES(AVG) =  1781  (MAX) =  1799  (MIN) =  1671
-------------------------------------------------------------------------------------------------------------------------------
                                          COUPLING  FACILITY  USAGE  SUMMARY
-------------------------------------------------------------------------------------------------------------------------------
 STRUCTURE SUMMARY
-------------------------------------------------------------------------------------------------------------------------------

                                                  % OF                 % OF      AVG     LST/DIR  DATA      LOCK     DIR
        STRUCTURE                       ALLOC     CF            #      ALL       REQ/    ENTRIES  ELEMENTS  ENTRIES  RECLAIMS
 TYPE   NAME              STATUS CHG    SIZE      STORAGE     REQ     REQ       SEC     TOT/CUR  TOT/CUR   TOT/CUR
 LIST   DSNDB1G_SCA       ACTIVE         16M        0.8%     3395    0.1%      1.89      26K       52K       N/A       N/A
                                                                                          92       158       N/A
        IXCPLEX_PATH1     ACTIVE         59M        2.9%   257609    8.6%    143.12      14K       14K       N/A       N/A
                                                                                           1        48       N/A

...

 LOCK   DSNDB1G_LOCK1     ACTIVE         31M        1.5%    64373    2.2%     35.76     114K        0      4194K       N/A
                                                                                          41        0      2730
        IRLMLOCK1         ACTIVE         31M        1.5%     1676K  56.0%    930.89     114K        0      4194K       N/A
                                                                                           6        0       256

 CACHE  DSNDB1G_GBP0      ACTIVE         21M        1.0%     3289    0.1%      1.83      22K      4321       N/A         0
                                                                                          58       35       N/A
...

                                        -------      -----   -------   ------   -------
            STRUCTURE TOTALS             263M       12.9%    3077K     100%   1709.2

 STORAGE SUMMARY
-------------------------------------------------------------------------------------------------------------------------------

                                          ALLOC     % OF CF     ------- DUMP SPACE -------
                                          SIZE      STORAGE     % IN USE   MAX % REQUESTED

 TOTAL CF STORAGE USED BY STRUCTURES       263M      12.9%
 TOTAL CF DUMP STORAGE                       6M       0.3%         0.0%              0.0%
 TOTAL CF STORAGE AVAILABLE                  2G      86.8%
                                          -------
 TOTAL CF STORAGE SIZE                       2G


                                          ALLOC     % ALLOCATED
                                          SIZE

 TOTAL CONTROL STORAGE DEFINED               2G      13.2%
 TOTAL DATA STORAGE DEFINED                  0K       0.0%

 PROCESSOR SUMMARY
-------------------------------------------------------------------------------------------------------------------------------

 AVERAGE CF UTILIZATION  (% BUSY)           7.8      LOGICAL PROCESSORS:   DEFINED   6    EFFECTIVE   6.0
                                                                                  A            B
```

Figure 94. Coupling Facility Activity Report: Usage Summary

Figure 95 on page 174 shows delayed requests. The following columns list the possible contention reasons for requests sent to the coupling facility:

REASON The reason for a delayed request can be either a subchannel contention (NO SCH) or a dump serialization (DUMP).

 # REQ / % of REQ The total number and the percentage of requests delayed in the interval.

AVG TIME - /DEL The average delay time, in microseconds, over all delayed requests.

AVG TIME - STD_DEV The standard deviation to the average delay time.

AVG TIME - /ALL The average delay time, in microseconds, over all requests, whether delayed or not.

Figure 95 on page 174 also shows external request contentions. These values are available for all serialized list structures as follows:

REQ TOTAL The number of requests against this structure.

REQ DEFERRED The number of requests running into a lock contention.

DATA ACCESS, as shown in Figure 95 on page 174, shows information for cache structures as follows:

READS The number of occurrences the coupling facility returned data on a read request by any connector (read hit). Directory-only caches always have a zero value reported since there is no data to be returned.

WRITES The number of occurrences data has been written to the cache structure. Directory-only caches always have a zero value reported since there are no data writes possible.

CASTOUTS The number of times CASTOUT processing occurs. This is the process of writing changed cache data to permanent storage. This counter is of interest for store-in cache structures (for example, DB2 global buffer pool structures) in determining the volume of changed data being removed from the structure.

XI′S The number of times a data item residing in a local buffer pool was marked invalid by the coupling facility. XI′s count values are seen for directory, store-in, and store-thru caches. This count reflects the amount of data sharing among the users of the cache and the amount of write or update activity against the databases.

```
                                C O U P L I N G   F A C I L I T Y   A C T I V I T Y
                                                                                                                                  PAGE   3
     OS/390                 SYSPLEX PLEXPERF           DATE 04/11/1996          INTERVAL 030.00.000
     REL. 01.02.00          RPT VERSION 1.2.0          TIME 12.00.00            CYCLE 01.000 SECONDS

------------------------------------------------------------------------------------------------------------------------------------
 COUPLING FACILITY NAME = CF1
------------------------------------------------------------------------------------------------------------------------------------
                                          COUPLING  FACILITY  STRUCTURE  ACTIVITY
------------------------------------------------------------------------------------------------------------------------------------

    STRUCTURE NAME = COUPLE_CKPT1      TYPE = LIST
              # REQ    -------------- REQUESTS -------------  -------------- DELAYED REQUESTS -------------
    SYSTEM    TOTAL             #     % OF  -SERV TIME(MIC)-    REASON    #   % OF  ---- AVG TIME(MIC) -----  EXTERNAL REQUEST
    NAME      AVG/SEC           REQ    ALL     AVG   STD_DEV            REQ    REQ   /DEL    STD_DEV   /ALL   CONTENTIONS

    J8O        8463    SYNC   2927    5.8%   306.6    127.8                                                    REQ TOTAL     833
               4.70    ASYNC  5535   11.0%  1502.1   1263.2    NO SCH 1240  22.4%  508.4      597.7  113.9   REQ DEFERRED   12
                       CHNGD     1    0.0%  INCLUDED IN ASYNC
                                                               DUMP      0   0.0%    0.0        0.0

...

   ---------------------------------------------------------------------------------------------------------------------------------
    TOTAL     50370    SYNC     17K  34.3%   342.2    124.5                                                    REQ TOTAL       5K
              27.98    ASYNC    33K  65.5%  557551G     0.0     NO SCH 8797  26.6%   1047       1245  278.4   REQ DEFERRED   49
                       CHNGD    69    0.1%
                                                               DUMP      0   0.0%    0.0        0.0    0.0
...
    STRUCTURE NAME = IRLMLOCK1         TYPE = LOCK
              # REQ    -------------- REQUESTS -------------  -------------- DELAYED REQUESTS -------------
    SYSTEM    TOTAL             #     % OF  -SERV TIME(MIC)-    REASON    #   % OF  ---- AVG TIME(MIC) -----  EXTERNAL REQUEST
    NAME      AVG/SEC           REQ    ALL     AVG   STD_DEV            REQ    REQ   /DEL    STD_DEV   /ALL   CONTENTIONS

...

    JFO         128K   SYNC    128K   7.7%   221.9     37.5                                                    REQ TOTAL      12K
               71.26   ASYNC      0   0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED    8
                       CHNGD      0   0.0%  INCLUDED IN ASYNC                                                  -CONT           8
                                                                                                               -FALSE CONT     2
...

   ---------------------------------------------------------------------------------------------------------------------------------
    TOTAL      1676K   SYNC   1676K   100%   198.1     44.7                                                    REQ TOTAL     167K
               930.9   ASYNC      0   0.0%     0.0      0.0     NO SCH    0   0.0%    0.0        0.0    0.0   REQ DEFERRED   90
                       CHNGD      0   0.0%                                                                    -CONT          90
                                                                                                               -FALSE CONT    22
...
    STRUCTURE NAME = DSNDB1G_GBP3      TYPE = CACHE
              # REQ    -------------- REQUESTS -------------  -------------- DELAYED REQUESTS -------------
    SYSTEM    TOTAL             #     % OF  -SERV TIME(MIC)-    REASON    #   % OF  ---- AVG TIME(MIC) -----
    NAME      AVG/SEC           REQ    ALL     AVG   STD_DEV            REQ    REQ   /DEL    STD_DEV   /ALL
...
    JAO         1032    SYNC   1025   19.1%   448.8     84.3
                0.57    ASYNC     0    0.0%  4508.9   1409.1    NO SCH    7   100%   2060       1671   2060
                        CHNGD     7    0.1%  INCLUDED IN ASYNC
                                                               DUMP      0   0.0%    0.0        0.0

...

   ---------------------------------------------------------------------------------------------------------------------------------
    TOTAL      5356    SYNC   5342   100%   432.8     89.0                                                    -- DATA ACCESS -
               2.98    ASYNC     0    0.0%  3685.2   1634.2    NO SCH   14   100%   1510       1374   1510    READS         133
                       CHNGD    14    0.3%                                                                    WRITES        170
                                                                  DUMP      0   0.0%    0.0        0.0    0.0    CASTOUTS        4
```

Figure 95. Coupling Facility Activity Report: Structure Activity

Figure 96 on page 175 shows the subchannel activity. For DELAYED REQUESTS, the following section provides a list of possible contention reasons for requests sent to the coupling facility:

 # REQ The total number of requests delayed in the RMF interval.

% OF REQ The percentage of requests delayed.

AVG TIME - /DEL The average delay time, in microseconds, over all delayed requests.

AVG TIME - STD_DEV The standard deviation to the average delay time.

AVG TIME - /ALL The average delay time, in microseconds, over all requests, whether delayed or not.

```
                              C O U P L I N G   F A C I L I T Y   A C T I V I T Y
                                                                                                          PAGE   6
 OS/390                 SYSPLEX PLEXPERF            DATE 04/11/1996           INTERVAL 030.00.000
 REL. 01.02.00          RPT VERSION 1.2.0           TIME 12.00.00             CYCLE 01.000 SECONDS
-----------------------------------------------------------------------------------------------------------------------------
COUPLING FACILITY NAME = CF1
-----------------------------------------------------------------------------------------------------------------------------
                                                    SUBCHANNEL  ACTIVITY
-----------------------------------------------------------------------------------------------------------------------------
         # REQ                                ----------- REQUESTS -----------   -------------- DELAYED REQUESTS ------------
SYSTEM   TOTAL                  --BUSY--               #   -SERVICE TIME(MIC)-          #    % OF  ------ AVG TIME(MIC) ----
NAME     AVG/SEC -- CONFIG --  -COUNTS-               REQ     AVG    STD_DEV           REQ    REQ   /DEL    STD_DEV     /ALL

J80      639197  SCH GEN    4  PTH  525     SYNC    602326   190.5      34.7   SYNC   271   0.0%    0.0        0.0      0.0
          355.1  SCH USE    4  SCH  271     ASYNC    36166  1940.7      1528   ASYNC 2723   7.5%   1106       1258     82.9
                 SCH MAX    4               CHANGED    167  INCLUDED IN ASYNC  TOTAL 2994   0.5%
                 PTH        2               UNSUCC       0     0.0       0.0
J90      568493  SCH GEN    4  PTH  360     SYNC    519833   194.0      40.0   SYNC   175   0.0%    0.0        0.0      0.0
          315.8  SCH USE    4  SCH  175     ASYNC    46272  1942.8     29558   ASYNC 2980   6.4%  20288     142.7K     1304
                 SCH MAX    4               CHANGED    102  INCLUDED IN ASYNC  TOTAL 3155   0.6%
                 PTH        2               UNSUCC       0     0.0       0.0
JA0      385380  SCH GEN    4  PTH    0     SYNC    352722   232.0      46.2   SYNC   715   0.2%    0.0        0.0      0.0
          214.1  SCH USE    4  SCH  715     ASYNC    30487  3655.0      2562   ASYNC 4470  14.5%   2426       2474    351.1
                 SCH MAX    4               CHANGED    406  INCLUDED IN ASYNC  TOTAL 5185   1.4%
                 PTH        2               UNSUCC       0     0.0       0.0
JB0       67806  SCH GEN    4  PTH    0     SYNC     34883   247.4      72.3   SYNC    40   0.1%   1468       2204      1.7
           37.7  SCH USE    4  SCH   40     ASYNC    30548  603152G      0.0   ASYNC 2679   8.8%   2510       2387    219.9
                 SCH MAX    4               CHANGED     31  INCLUDED IN ASYNC  TOTAL 2719   4.2%
                 PTH        2               UNSUCC       0     0.0       0.0
```

Figure 96. Coupling Facility Activity Report: Subchannel Activity

10.3 Postprocessor OpenEdition Reporting

Monitor III gathers data for this report automatically. If you want to supress gathering, you have to disable writing SMF record type 74.3. To produce the report, specify:

REPORTS(OMVS)

The OMVS Kernel Activity report, shown in Figure 97 on page 176, has the following parts:

- OMVS System Call Activity
- OMVS Process Activity
- OMVS Inter-Process Communication
- OMVS Memory Map

All average values derived from accumulated fields are marked with an asterisk (*) if the OMVS kernel address space was reinstated during the interval. If the OMVS process limits (MAXIMUM line) have changed, they are reported as ****.

```
                                    O M V S   K E R N E L   A C T I V I T Y
                                                                                                                    PAGE   1

          OS/390                 SYSTEM ID AQTS            DATE 03/08/1996      INTERVAL 30.00.000
          REL. 01.02.00          RPT VERSION 1.2.0         TIME 13:00:00        CYCLE 1.000 SECONDS

TOTAL SAMPLES = 1,800

                                                OMVS SYSTEM CALL ACTIVITY
-------------------------------------------------------------------------------------------------------------------------------
                 MINIMUM  AVERAGE  MAXIMUM
-------------------------------------------------------------------------------------------------------------------------------
 SYSCALLS (N/S)     23.5     2300*    5699
 CPU TIME (H/S)       16       47*      88

                                                OMVS PROCESS ACTIVITY
-------------------------------------------------------------------------------------------------------------------------------
                          PROCESSES                        USERS                  PROCESSES PER USER
 MAXIMUM  (TOT)              1200                            50                          12
-------------------------------------------------------------------------------------------------------------------------------
                 MINIMUM  AVERAGE  MAXIMUM      MINIMUM  AVERAGE  MAXIMUM      MINIMUM  AVERAGE  MAXIMUM
-------------------------------------------------------------------------------------------------------------------------------
 CURRENT  (TOT)       99      854     1200           12       13       23
 OVERRUNS (N/S)        0      5.5*     333            0      1.5*     4.8            0      1.0*     5.3

                                             OMVS INTER-PROCESS COMMUNICATION
-------------------------------------------------------------------------------------------------------------------------------
                   MESSAGE QUEUE IDS               SEMAPHORE IDS            SHARED MEMORY IDS          SHARED MEMORY PAGES
 MAXIMUM  (TOT)              500                        500                        500                      262144
-------------------------------------------------------------------------------------------------------------------------------
                 MINIMUM  AVERAGE  MAXIMUM    MINIMUM  AVERAGE  MAXIMUM    MINIMUM  AVERAGE  MAXIMUM   MINIMUM  AVERAGE  MAXIMUM
-------------------------------------------------------------------------------------------------------------------------------
 CURRENT  (TOT)      100      300      500        100      300      500        100      300      500       100   131072   26214
 OVERRUNS (N/S)        0      10*      100          0      10*      100          0      10*      100         0      10*      10

                                                OMVS MEMORY MAP
-------------------------------------------------------------------------------------------------------------------------------
                 MEMORY MAP STORAGE PAGES        SHARED STORAGE PAGES
 MAXIMUM  (TOT)              2048                       4096
-------------------------------------------------------------------------------------------------------------------------------
                 MINIMUM  AVERAGE  MAXIMUM    MINIMUM  AVERAGE  MAXIMUM
-------------------------------------------------------------------------------------------------------------------------------
 CURRENT  (TOT)      111     1111     2048        222     2222     4096
 OVERRUNS (N/S)        1      10*      100          2       20*     200

 UNITS:   (TOT) = TOTAL VALUE, (N/S) = NUMBER PER SECOND, (H/S) = HUNDREDTHS OF SECONDS PER SECOND
```

Figure 97. OMVS Kernel Activity Report

The fields in the OMVS Kernel Activity report, shown in Figure 97, are defined as follows:

OMVS SYSTEM CALL ACTIVITY

SYSCALLS (N/S)
Number of system calls per second processed by the OMVS kernel address space in this interval; reported as MINIMUM, AVERAGE, and MAXIMUM.

CPU TIME (H/S)
Time spent to process system calls in hundredths of seconds per second; reported as MINIMUM, AVERAGE, and MAXIMUM.

OMVS PROCESS ACTIVITY

MAXIMUM
The maximum number of processes, users, and processes per user defined by OMVS kernel address space initialization parameters (in SYS1.PARMLIB(OMVSPMxx)). If one of these values has changed (due to an OMVS restart), it is reported as ****.

CURRENT PROCESSES	The number of OMVS processes controlled by OMVS during this interval; reported as MINIMUM, AVERAGE, and MAXIMUM.
CURRENT USERS	The number of OMVS users controlled by OMVS during this interval; reported as MINIMUM, AVERAGE, and MAXIMUM.
OVERRUNS PROCESSES	The rate of processes that could not be created by OMVS because the maximum number of processes would have been exceeded; reported as MINIMUM, AVERAGE, and MAXIMUM.
OVERRUNS USERS	The rate of OMVS users that could not be created by OMVS because the maximum number of users would have been exceeded; reported as MINIMUM, AVERAGE, and MAXIMUM.
OVERRUNS PROCESSES PER USER	The rate of processes per user that could not be created by OMVS because the maximum number of processes per user would have been exceeded; reported as MINIMUM, AVERAGE, and MAXIMUM.

OMVS INTER-PROCESS COMMUNICATION

MAXIMUM	The maximum number of message queue IDs, semaphore IDs, shared memory IDs, and shared memory pages defined by OMVS kernel address space initialization parameters (in SYS1.PARMLIB(OMVSPMxx)).
CURRENT MESSAGE QUEUE IDS	The number of message queue IDs during this interval; reported as MINIMUM, AVERAGE, and MAXIMUM.
CURRENT SEMAPHORE IDS	The number of semaphore IDs during this interval; reported as MINIMUM, AVERAGE, and MAXIMUM.
CURRENT SHARED MEMORY IDS	The number of shared memory IDs during this interval; reported as MINIMUM, AVERAGE, and MAXIMUM.
CURRENT SHARED MEMORY PAGES	The number of shared memory pages during this interval, reported as MINIMUM, AVERAGE, MAXIMUM.

OVERRUNS MESSAGE QUEUE IDS The rate of message queue IDs that could not be created by OMVS because the maximum number of message queue IDs would have been exceeded; reported as MINIMUM, AVERAGE, and MAXIMUM.

OVERRUNS SEMAPHORE IDS The rate of semaphore IDs that could not be created by OMVS because the maximum number of semaphore IDs would have been exceeded; reported as MINIMUM, AVERAGE, and MAXIMUM.

OVERRUNS SHARED MEMORY IDS The rate of shared memory IDs that could not be created by OMVS because the maximum number of shared memory IDs would have been exceeded; reported as MINIMUM, AVERAGE, and MAXIMUM.

OVERRUNS SHARED MEMORY PAGES The rate of shared memory pages that could not be created by OMVS because the maximum number of shared memory pages would have been exceeded; reported as MINIMUM, AVERAGE, and MAXIMUM.

OMVS MEMORY MAP

MAXIMUM The maximum number of memory map storage pages and shared storage pages defined by OMVS kernel address space initialization parameters (in SYS1.PARMLIB(OMVSPMxx)).

CURRENT MEMORY MAP STORAGE PAGES The number of memory map storage pages during this interval; reported as MINIMUM, AVERAGE, and MAXIMUM.

CURRENT SHARED STORAGE PAGES The number of shared storage pages during this interval; reported as MINIMUM, AVERAGE, and MAXIMUM.

OVERRUNS MEMORY MAP STORAGE PAGES The rate of memory map storage pages that could not be created by OMVS because the maximum number of memory map storage pages would have been exceeded; reported as MINIMUM, AVERAGE, and MAXIMUM.

OVERRUNS SHARED STORAGE PAGES The rate of shared storage pages that could not be created by OMVS because the maximum number of shared storage pages would have been exceeded; reported as MINIMUM, AVERAGE, and MAXIMUM.

Notes:

1. If the OMVS address space was ended and reinitialized during the measurement interval, all average values derived from accumulative counts are marked with an asterisk (*). This reminds the user that these values are calculated from an elapsed time that is smaller than the interval time.
2. If the OMVS address space was ended and reinitialized during the measurement interval, and the process limits defined in SYS1.PARMLIB(OMVSPMxx) have changed, the corresponding fields in the report contain ****.

10.4 Monitor II Changes

Corresponding to the remote reporting capability in Monitor III (introduced with RMF 5.1), Monitor II now also provides the capability to report on any system in a sysplex. On each report panel, you can select the system to report on either by the new command SYSTEM smf-id, or by overwriting the contents of the new System= field in the header area (ISPF-session only).

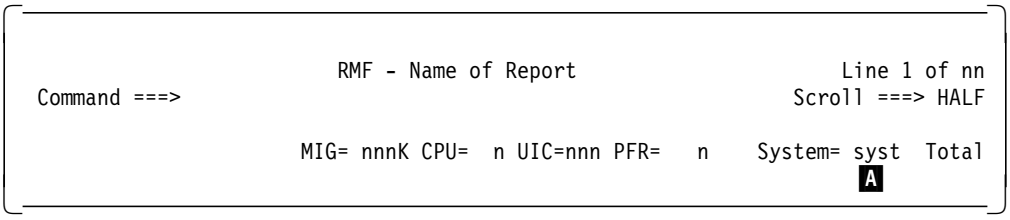


Figure 98. Header of a Monitor II ISPF Session Report

A Overtype SMF-ID

10.4.1 Sysplex Data Server Reporting

A SDS Report can be obtained in ISPF by specifying 8 on the Resource Report Selection menu or by using the command interface and specifying SDS.

The report provides statistics about the usage of the SMF data buffer and the usage of Sysplex Data Services.

The numbers presented in the first section aid in finding the optimal size of the SMF buffer to hold as many SMF records as an installation might want to keep for immediate sysplex reporting. The second section contains statistics about the exploitation of the callable services. You can use these numbers to optimize the usage of the callable services within applications other than RMF.

The SDS report requires that the RMF address space has been started. Otherwise, no statistics can be provided from the reporting system or from any remote system in the sysplex.

```
                  RMF - SDS RMF Sysplex Data Server            Line 1 of 13
Command ===>                                                  Scroll ===> HALF

                      MIG= 767K CPU= 40 UIC=254 PFR=   0    System= SYS1  Total

RMF Sysplex Data Server Statistics
    Report Start 11/29/1995 16:49:24, End 12/06/1995 12:28:21, Duration   6d 19h


SMF Buffer Statistics, Start 11/29/1995 16:49:24, Duration    6d 19h
    Buffer     Records    Record     Avg Queue   Avg Rec   Records    Buffer
    Size (b)   Arrived    Rate (/h)  Time (ms)   Length    in Buffer  Wrap Time
         1M     23473        143         51          1K       615      04:23:18

Callable Services Statistics
    Service    Requests   Request    Avg Srv    Avg Sys    Avg Amnt
    Name       Arrived    Rate (/h)  Time (ms)     /Req    Data /Req
    ERBDSQRY         0          0          0          0          0
    ERBDSREC         0          0          0          0          0
    ERB2XDGS         0          0          0          0          0
    ERB3XDRS      3058         18       2463          3         21K
```

Figure 99. SDS Report

The RMF SDS Sysplex Data Server Statistics fields are as follows:

SMF Buffer Statistics

Buffer size The size in bytes as specified in the SPACE subparameter and UF parameter for the RMF address space.

Records Arrived The number of records that arrived during the buffer duration.

Record Rate (/h) SMF data arrival rate in records per hour.

Avg Queue Time Data server internal processing time for SMF data, in milliseconds.

Avg Rec Length Average record length for the buffer statistics duration, in bytes.

Records in buffer Number of records currently in the SMF data buffer.

Buffer Wrap Time Wrap-around time for the SMF data buffer (data residency time); this time is estimated before the second wrap and measured afterwards. The format can be in days and hours (6d 19h) or in hh:mm:ss.

Callable Services Statistics

Service Name Name of the sysplex data service module:

ERBDSQRY RMF Query Available Sysplex SMF Data Service

ERBDSREC RMF Request Sysplex SMF Record Data Service

ERB2XDGS RMF Monitor II Sysplex Data Gathering Service

	ERB3XDRS RMF Monitor III Sysplex Data Retrieval Service
Requests Arrived	Number of calls to the data service
Request Rate (/h)	Data services call rate in calls per hour
Avg Srv Time (ms)	Average response time (in milliseconds) for data services
Avg Sys /Req	Average number of systems a request was propagated to
Avg Amnt Data /Req	Average amount of data returned by a request, in bytes

10.5 Monitor III Changes

Monitor III has added two new reports to track the following:

- Channel Path Activity
- I/O Queuing Activity

10.5.1 Channel Path Activity Report

To request the Channel Path Activity report, select **3** from the primary menu and then **12** on the Resource Report menu or you can enter the CHANNEL command.

The Channel Path Activity report (CHANNEL) gives you information about channel path activity for all channel paths in the system. The report contains data for every channel path that is online during data gathering.

You can use channel path activity information together with I/O device activity and I/O queuing activity information to identify performance bottlenecks associated with channel paths.

To find out which logical control unit is using the channel, look in the I/O Queuing Activity report. From there you can go to check device response times. For example, if a channel path to a device shows excessive use, you could define additional paths to the device or introduce a different job mix to produce better performance.

You can obtain the report whether or not a Monitor I session that is measuring channel path activity is active.

Data for total utilization and partition utilization is gathered independently. Because the internal interval used to gather this data is a few seconds, the total utilization and the sum of the utilizations of the partitions sharing that channel might differ if a short RMF interval is specified. If the interval is too small or the appropriate data cannot be gathered, dashes(---) are reported instead of data.

Figure 100 on page 182 and Figure 101 on page 182 are examples of the Channel Path Activity report in BASIC and LPAR mode. The graphic form of this report shows the percent of total utilization for each channel.

```
                          RMF 1.2.0  Channel Path Activity      Line 1 of 120
Command ===>                                                  Scroll ===> HALF

Samples:     100 System: MVS1  Date: 04/20/96  Time: 08.53.20   Range:  100 Sec

Channel Path     Utilization(%)
ID   Type SHR   Partition  Total

00    BY                   0.00
01    BL                   0.63
03    BL                   0.00
05    BL                   0.57
07    BL                   0.00
08    BL                   3.73
0A    BL                   0.00
0B    BL                   0.63
0D    BL                   0.60
0E    BL                   0.00
10    BL                   0.00
11    BL                   0.00
14    BL                   1.53
```

Figure 100. CHANNEL Report - BASIC Mode

```
                          RMF 1.2.0  Channel Path Activity      Line 14 of 82
Command ===>                                                  Scroll ===> HALF

Samples:     100 System: MVS3  Date: 04/20/96  Time: 17.04.00   Range:  100 Sec

Channel Path     Utilization(%)
ID   Type SHR   Partition  Total

15    BL           0.00     0.00
17    BL           0.00     0.00
18    BL           7.13     7.13
1A    BL           0.00     0.00
1C    BL           9.50     9.50
20    CN D  Y     11.03    21.07
22    BL           8.03     8.03
24    CN    Y      0.57     0.57
25    BL           3.83     3.83
27    CV           0.00     0.00
28    BL           6.20     6.20
```

Figure 101. CHANNEL Report - LPAR Mode

10.5.2 I/O Queuing Activity Report

To request the I/O Queuing Activity report, select **3** from the primary menu, and then **13** on the Resource Report menu, or you can enter the IOQUEUE command.

The I/O Queuing Activity report (IOQUEUE) provides information, grouped by LCU (logical control unit), on the I/O configuration. The information includes contention rate, queue lengths, and percentages of time when one or more I/O components were busy. Information about the LCU is useful because the LCU is the focus of I/O configuration and path management measurements for a related group of I/O devices.

Note: This report is available on a 3090 or an ES/9000 series processor only.

An LCU is the set of devices attached to the same physical control unit (or group of control units that have one or more devices in common). Each device belongs to only one LCU, but the I/O processor (IOP), which is part of the channel subsystem, manages and schedules I/O work requests to the various devices within the LCU.

This report can tell you about the cause of performance problems associated with channel paths and devices. You could, for example, find the reason for an unusually long pending time reported on the device report. Check the relationship between the percentage of requests deferred for device busy and control unit busy for the LCU.

10.5.2.1 Special Considerations of Report Output

The report depends on data that the Monitor I session collects. To get this report, the Monitor I I/O queuing activity measurement must be active. The Monitor I gatherer gets a new set of model-dependent data every second or every cycle, whichever time period is greater.

Data items that are not valid are marked by dashes (---) in the output display.

In contrast to the Monitor II IOQUEUE report, an LCU without any activity during the interval is not omitted from the report. If there are changes in the status or in the configuration during the interval, the report is shown with the new configuration, but without measurement data.

Note: In a VM guest system environment, the report for an MVS system that is authorized via the VM RMCHINFO directory option shows only static configuration data. Measurement data is not available.

10.5.2.2 Contents of the Report

Figure 102 on page 184 is an example of the I/O Queuing Activity report. The description of the fields is the same as in Monitor I. The graphic form of this report shows the contention rate of each LCU.

```
                          RMF 1.2.0  I/O Queuing Activity        Line 1 of  6
Command ===>                                                  Scroll ===> HALF

Samples:     100 System: MVS1  Date: 04/20/96  Time: 08.53.20     Range:  100 Sec

                                          Cont  Del Q   %All    CHPID  %DP   % CU
Chan Paths Control Units       LCU       Rate  Lngth  Ch busy  Taken  busy  Busy

2F         200A                016A       0.0   0.00   0.00     0.99  2.47  0.00
32         200A                016A                             0.86  0.72  0.00
33         200B                016A                             0.85  0.73  0.00
35         200B                016A                             0.89  1.37  0.68
33         400A                016D       0.0   0.00   0.00     1.69  0.00  0.00
35         400B                016D                             1.69  0.00  0.00
```

Figure 102. IOQUEUE Report

Chapter 11. Small Programming Enhancement for CFLEVEL=2

OS/390 Release 2 includes the Small Programming Enhancement for CFLEVEL=2 functions of the coupling facility. This SPE is also known as CF Batched Unlocks.

An enhanced level of coupling facility architecture, a level 2 coupling facility, is available. It provides enhancements for cache and lock structure functions.

A level 2 coupling facility includes the following additional functions:

- Cache structure enhancements

  A cache structure user can:

  - Specify a list of data items to be registered in a single coupling facility operation.
  - Write data to the cache structure conditionally, if and only if the data item is currently registered to the user with a particular specified local cache vector index.

- Lock structure enhancements

  A lock structure user can:

  - Request that a list of lock entries be released in one operation, rather than as individual operations.

The following IBM CPCs support a level 2 coupling facility:

2003 ICMF support only (all ECs).

9672 R4 CPCs (all ECs)

9672 R3 CPCs at EC E12923 with Licensed Internal Code (LIC) updates.

9672 R2 CPCs at EC E45548.

9672 R1, E, and P CPCs at EC D79759 with LIC updates (also available at EC D57262 with LIC updates).

9674 Model C04 (all ECs).

9674 Model C03 at EC E12971 with LIC updates.

9674 Model C02 at EC E45548.

9674 Model C01 at EC D79786 with LIC updates (also available at EC D57264 with LIC updates).

9021 711-based models at EC 236422 with LIC updates.

9121 511-based models ICMF support only at EC C35956 with LIC updates.

11.1 Use of Coupling Facility Level in MVS

MVS attempts to allocate a structure in a coupling facility of the CFLEVEL requested, that is, a coupling facility that provides at least the level of architected function that was requested. If necessary, the structure is allocated in a coupling facility with a CFLEVEL lower than requested. It is the responsibility of the connector to check and verify that the level is acceptable.

The following is a list of functions introduced in different CFLEVELs on the top of the basic functions.

- CFLEVEL=1
  - Maximum of 255 data elements per data item
  - IXLALTER requests for altering structure size and entry-to-element ratio
  - IXLLIST request types that support entry version number comparison, automatic list key assignment, list cursor manipulation, entry key comparison, and conditional processing based on list authority
- CFLEVEL=2
  - IXLCACHE REQUEST=REG_NAMELIST

    Specifies a list of items to be registered in a single operation.
  - IXLCACHE REQUEST=WRITE_DATA, WHENREG=YES, with VECTORINDEX specified.

    Write data to the cache structure conditionally, if the data item is currently registered to the user with a particular specified local cache vector index.
  - IXLLOCK REQUEST=PROCESSMULT for batched release requests. A lock structure user requests that a list of lock entries be released in one operation.

OS/390 V1R2.0 MVS Sysplex Services Guide provides more information about Parallel Sysplex services.

11.2 Use of CFLEVEL=2 in Subsystems

In order for these functions to be available for coupling facility users, the installation must upgrade one or more coupling facilities to support CFLEVEL=2 and have the functions available to authorized programs running in the sysplex. There is no implied ordering between upgrading CFs to CFLEVEL=2 and installing the MVS APARs to support CFLEVEL=2; however, both must be done before a coupling facility user may exploit the functions of CFLEVEL=2.

11.2.1 DB2 Example

APAR PN72413 in DB2 V4.1 exploits the IXLCACHE register name list to improve prefetch performance in the Data Sharing environment by taking advantage of IXLCACHE register name list.

When prefetching pages for Group Buffer Pool dependent pagesets, DB2 has to register interest in the page and retrieve the page from the GBP. This currently requires a separate coupling facility interaction for each requested page. By making use of the “Register-Name-List” request, a set of up to 32 pages may be registered with a single coupling facility interaction. If the Register-Name-List function is available, determined by the ability to connect to the Group Buffer Pool with a CF level of 2, the prefetch code uses it to register groups of pages. If the function is not available, the old logic of issuing a separate “read-and-register” for each page is used instead.

11.2.2 IRLM Example

IRLM APAR PN83549 exploits the batch unlock feature by consolidating the release of resources into a single request thus giving a performance improvement. IRLM can take advantage of this feature without any change to the DBMSs using IRLM.

11.3 Implementing CFLEVEL=2 SPE

OS/390 Release 2 includes the SPE for CFLEVEL=2. The SPE is also available for MVS/ESA 5.2 as PTF UW23470, and for OS/390 Release 1 as PTF UW23471.

Installation should make CFLEVEL=2 CFs available for exploiters who wish to use these functions. If the available coupling facilities provide different levels of functions, this may require changes to CFRM policies.

The implementation of CFLEVEL=2 type coupling facilities also changes the algorithms used to estimate the structure sizes. *Processor Resource/Systems Manager Planning Guide* provides information about CFLEVEL=2 planning and structure sizing considerations.

Appendix A. GRS Star Implementation Checklist

The following checklist makes an inventory of the analysis and activities to be followed while planning a migration to a GRS star configuration.

Table 16 (Page 1 of 2). GRS Star Implementation Checklist

	Related Information
A coupling facility connected to all systems in the complex is available. There is no special CFCC (coupling facility control code) level required by GRS.	None
A second fully connected coupling facility is available for recovery.	None
The required coupling facility links are available.	None
All MVS systems are OS/390 Release 2 or above.	None
The sysplex complex is matching the GRS complex. **Note:** A mixed complex is not supported.	None
All systems in the sysplex operate within the GRS star complex. **Note:** GRS ring and star complex cannot be interconnected.	None
All connections to shared DASD are correct.	None
If some DASD are shared outside the sysplex, the DASD resources serialization is guaranteed only with a hardware RESERVE.	Appendix C, “ISGGREX0 Sample Exit” on page 199
Sysplex couple data sets, primary and alternate, have been recreated with the GRS ITEM parameter.	6.5, “GRS Star Implementation and Migration” on page 75
The COUPLExx parmlib member has been updated to reflect the newly-created sysplex couple data sets.	6.5, “GRS Star Implementation and Migration” on page 75
The CFRM policy has been updated for the GRS lock structure (ISGLOCK).	6.5, “GRS Star Implementation and Migration” on page 75
The lock structure size is set up according to the the number of outstanding GRS global requests during peak load measured with utility ISGSCGRS.	6.5, “GRS Star Implementation and Migration” on page 75
Verify the CFRM and the SFM policies for appropriate REBUILDPERCENT and system weights values.	6.5, “GRS Star Implementation and Migration” on page 75
IEASYSxx and GRSCNFxx parmlib members have been updated for the GRS star configuration, and, for fallback possibility, additional members are available for GRS ring configuration.	6.5, “GRS Star Implementation and Migration” on page 75
Data describing the behavior of GRS ring complex is collected. • Average GRS ring delay and behavior. • Average elapsed time for converted and excluded reserves. • Average storage and processor utilization for GRS address spaces.	Appendix B, “ENQ/DEQ/RESERVE Analysis Aid Reports” on page 191
RMF sysplex data server implemented, and RMF JCLs and procedures to monitor the behavior of the coupling facility structures are available.	None
Operational procedure updated to manage the GRS lock structure.	None

Table 16 (Page 2 of 2). GRS Star Implementation Checklist	
	Related Information
Automation procedures using GRS messages updated for the new and changed GRS messages.	6.6, “GRS Star Operation” on page 82
Check if the limitation introduced by GRS star for ENQ=TEST service affects any application. See 6.3.2.1, “ENQ RET=TEST” on page 71.	*OS/390 MVS: Authorized Assembler Services Reference*, GC28-1647
Check if the change introduced by GRS star for GQSCAN service affects any application. See 6.3.4, “Global GQSCAN Processing” on page 72.	*OS/390 MVS: Authorized Assembler Services Reference*, GC28-1647

Appendix B. ENQ/DEQ/RESERVE Analysis Aid Reports

The tool monitors supervisor calls 56 and 48 (ENQ/RESERVE/DEQ) and collects data about the resources serialized and the requestors.

The objectives of the tool are:

1. Assist in planning the RNLs for global resource serialization implementation.
   - Find the name and the scope of ENQ requests used.
   - Measure the rate and time of the RESERVE requests used.
2. Help to detect the shared DASD device contention and interlock exposures.
   - Measure the total and the maximum reserve time of the DASD volumes with RESERVE activities.
   - List the resources that contributed to the reserve time.
   - Verify the results when RNLs are implemented.
3. Global resource serialization tuning.
   - Measure the global resource serialization delay for global ENQ requests and the actual RESMIL value for GRS ring configuration.

The reports produced by the tool help the systems programmer to understand ENQ/DEQ activity in global resource serialization complex and to set up RNLs. Examples of post-processing reports are included on the following pages.

The monitor collects the data into a dataspace. An ISPF application allows you to browse interactively the accumulated data. Sample of ISPF panels are shown on the following pages.

Note: The ENQ/DEQ/RESERVE Analysis Aid Reports program is available in some countries as an IBM service. The package is available from a TOOLSRUN based repository. The address is OFFERING disk CETOROMA at ROMEPPC. The package is referred to as BOCA-AUD PACKAGE. The repository is for IBM employees involved with value-added service offerings. If you wish to subscribe or get access to the repository, send a note to ESDP at ROMEPPC. When authorized, you can use the following commands to get or to subscribe for the package:

```
TOOLS SENDTO ROMEPPC CETOROMA OFFERING GET BOCA-AUD PACKAGE

TOOLS SENDTO ROMEPPC CETOROMA OFFERING SUB BOCA-AUD PACKAGE
```

```
DATE=(94001.1600:94365.1100)                          ** TRACE  REPORT **

    TIME     JOBNAME PROGRAM  AUT SVC RNL SCOPE     VOLSER   MAJOR     MINOR                                                  DEV SMF  ML

DATE  94 161

09 05.41.26  USER12L  IEWL      L ENQ    RESERVE 01 TOTTS1  SYSIEWLP USER12.AUDIT.LINKLIB.NAUTH                          0FCCS005 44
09 05.41.37  USER12L  SVC-019     ENQ    SYSTEMS EX         SYSZDSCB TOTTS1AUSER12.AUDIT.LINKLIB.NAUTH                      S005 33
09 05.41.44  USER12L  SVC-130     ENQ    SYSTEM  SH         SYSZRAC2 SYS1.RACFESA                                           S005 12
09 05.41.44  USER12L  SVC-130     ENQ  C*SYSTEMS SH TOTSM1  SYSZRACF SYS1.RACFESA                                         0FC2S005 12
09 05.41.48  USER12L  SVC-130     DEQ  C*SYSTEMS    TOTSM1  SYSZRACF SYS1.RACFESA                                           S005 12
   ////
09 05.42.09  USER12L  SVC-099     ENQ  C*SYSTEMS EX TOTPUB  SYSVTOC  TOTPUB                                                0FCDS005 06
09 05.42.26  USER12L  SVC-099     DEQ    REL/SYS            SYSVTOC  TOTPUB                                                  S005 06
09 05.42.29  USER12L  IEFIIC    L DEQ  I*SYSTEMS            SYSDSN   USER12.AUDIT.LINKLIB.NAUTH                            S005 26
09 05.42.32  USER12L  IEFIIC    L DEQ    SYSTEM  G          SYSZVOLS    a                                                   S005 04
09 05.42.38  USER12L  IEFIIC    L DEQ    SYSTEM  G          SYSZVOLS    a                                                   S005 04
09 05.42.43  US11     ISFMAIN   L ENQ    SYSTEMS EX         SYSZSDSF ISF.HASPINDX                                 TOTRS1    S005 50
09 05.42.44  USER12L  IEFIIC    L DEQ  I*SYSTEMS G          SYSDSN      a                                                   S005 04
09 05.42.59  USER12L  SVC-099     ENQ    SYSTEM  SH         SYSIEFSD Q4                                                     S005 02
09 05.42.59  USER12L  SVC-099     DEQ    SYSTEM             SYSIEFSD Q4                                                     S005 02
09 05.42.59  USER12L  SVC-099     ENQ  C*SYSTEMS EX TOTPUB  SYSVTOC  TOTPUB                                                0FCDS005 06
   ////
09 06.04.24  JES2     HASJES20 AL DEQ    SYSTEM             SYSZJES2 SVJ LOCK                                               S005 08
09 06.04.31  USE15    SVC-035     ENQ    SYSTEM  SH         SYSZMCS  MPFTABLE                                               S005 08
09 06.04.31  USE15    SVC-035     DEQ    SYSTEM             SYSZMCS  MPFTABLE                                               S005 08
09 06.04.36  USE15    IEFSD060  L ENQ  I*SYSTEMS SH         SYSDSN   ESA.SYS1.CLIST                                        S005 14
09 06.04.38  USE15    SVC-035     ENQ    SYSTEM  SH         SYSZMCS  MPFTABLE                                               S005 08
09 06.04.38  USE15    SVC-035     DEQ    SYSTEM             SYSZMCS  MPFTABLE                                               S005 08
09 06.04.38  CATALOG  IGGPACDV AL ENQ  C*SYSTEMS SH TOTCAT  SYSIGGV2 CATALOG.TOTICFM.VTOTCAT                               0FC7S005 44
09 06.04.42  CATALOG  IGGPACDV AL ENQ    SYSTEM  EX         SYSZVVDS CATALOG.TOTICFM.VTOTCAT                                S005 44
09 06.04.42  CATALOG  IGGPACDV AL ENQ  C*SYSTEMS SH TOTCAT  SYSZVVDS TOTCAT                                                0FC7S005 06
09 06.04.48  CATALOG  IGGPACDV AL DEQ  C*SYSTEMS    TOTCAT  SYSZVVDS TOTCAT                                                  S005 06
09 06.04.48  CATALOG  IGGPACDV AL DEQ    SYSTEM             SYSZVVDS CATALOG.TOTICFM.VTOTCAT                                S005 44
09 06.04.48  CATALOG  IGGPACDV AL DEQ    REL/SYS            SYSIGGV2 CATALOG.TOTICFM.VTOTCAT                                S005 44
    TIME     JOBNAME  PROGRAM AUT SVC RNL SCOPE     VOLSER   MAJOR     MINOR               DEV SMF-ID

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

SCOPE -> RESERVE  XX = DEVICE RESERVE COUNT               C*SYSTEMS  ->  RESERVE CONVERTED

      -> RELEASE  XX = DEVICE RESERVE COUNT               I*SYSTEMS  ->  SYSTEM  INCLUDED

      -> REL/SYS  XX = DEQ ASSOCIATED WITH RESERVE        E*SYSTEM   ->  SYSTEMS EXCLUDED

                                                          E*RESERVE  ->  RESERVE EXCLUDED (SYSTEM)
      -> SYSTEM(S)SH = SHARED
                  EX = EXCLUSIVE
                  G  = GENERIC DEQ
                 ' ' = SYSTEM  DEQ

      -> REL/SYS ' ' = SYSTEMS DEQ
```

Figure 103. Trace Report

```
                                                ** RESOURCES USAGE REPORT **
-------------------------------------------------------------------------------------------------------------------------------------------
 COUNT              MAJOR                                                    VOLUME AV-MSEC MIN-MSEC MAX-MSEC  TOT-SEC  RATE/MIN

00000001            SYSIEWLP            SCOPE=RESERVE                SMFID=SO05

RNL (ML)   COUNT     MAJOR     MINOR

     44   00000001  SYSIEWLP MERONI.AUDIT.LINKLIB.NAUTH                      TOTTS1  000549  000549  00000549  00000000  000000
-------------------------------------------------------------------------------------------------------------------------------------------
 COUNT              MAJOR

00000985            SYSZVVDS            SCOPE=SYSTEM                 SMFID=SO05

RNL (ML)   COUNT     MAJOR     MINOR

     44   00000720  SYSZVVDS CATALOG.TOTICFM.VTOTCAT
     44   00000143  SYSZVVDS CATALOG.TOTICF1.VTOTCAT
-------------------------------------------------------------------------------------------------------------------------------------------
 COUNT              MAJOR

00000976            SYSIGGV2            SCOPE=SYSTEMS                SMFID=SO05

RNL (ML)   COUNT     MAJOR     MINOR

C*   44   00000052  SYSIGGV2 CATALOG.TOTICF1.VTOTTP1
C*   88   00000003  SYSIGGV2 NITZ.LO6RSO05.SPFLOG1.LIST                 CATALOG
-------------------------------------------------------------------------------------------------------------------------------------------
 COUNT              MAJOR

00000400            SYSDSN              SCOPE=SYSTEMS                SMFID=SO05

RNL (ML)   COUNT     MAJOR     MINOR

I*   12   00000005  SYSDSN   ICQ.ICQAATAB
I*   41   00000001  SYSDSN   SYS1.DUMP.D940608.T192346.LO6RSO05.SO0001
-------------------------------------------------------------------------------------------------------------------------------------------
 COUNT              MAJOR

00000176            IGDCDSXS            SCOPE=SYSTEMS                SMFID=SO05

RNL (ML)   COUNT     MAJOR     MINOR

C*   44   00000176  IGDCDSXS SYS1.COMMDS10
-------------------------------------------------------------------------------------------------------------------------------------------
 COUNT              MAJOR

00000107            SYSZRACF            SCOPE=SYSTEMS                SMFID=SO05

RNL (ML)   COUNT     MAJOR     MINOR

C*   12   00000102  SYSZRACF SYS1.RACFESA
C*   15   00000005  SYSZRACF SYS1.RACF.BKUP1
-------------------------------------------------------------------------------------------------------------------------------------------
 COUNT              MAJOR

00000059            SPFEDIT             SCOPE=SYSTEMS                SMFID=SO05

RNL (ML)   COUNT     MAJOR     MINOR

     52   00000006  SPFEDIT  USER12.ISPF.ISPPROF                        PDFEDIT
C*   44   00000003  SPFEDIT  USER12.ISPF.ISPPROF
-------------------------------------------------------------------------------------------------------------------------------------------
 COUNT              MAJOR

00000041            SYSVSAM             SCOPE=SYSTEMS                SMFID=SO05

RNL (ML)   COUNT     MAJOR     MINOR

     51   00000003  SYSVSAM  CNM.V2R4MO.CNMMSGF.INDEXCATALOG.TOTICF1.VTOTTP1   0
I*   08   00000001  SYSVSAM  VSICHAIN
-------------------------------------------------------------------------------------------------------------------------------------------
```

Figure 104. ENQ/DEQ Resources Usage Report

```
                                ** VOLUMES RESERVE-TIME REPORT **

-------------------------------------------------------------------------------------------------------------------------
START=94.161 09.05.41   END=94.161 09.05.41   VOLUME=TOTTS1    SMFID=SO05                    ELAPSED SECONDS=00000000

 COUNT   TIME OF MAX                            DEV    MAX-COUNT      VOLUME  AV-MSEC   MIN-MSEC  MAX-MSEC  TOTAL-SEC RATE/MIN

00000001 94.161 09.05.41.26                     0FCC      01          TOTTS1  00000549  00000549  00000549  00000000  00000001

00000001 SYSIEWLP USER12.AUDIT.LINKLIB.NAUTH                          TOTTS1  00000549  00000549  00000549  00000000  00000000

-------------------------------------------------------------------------------------------------------------------------
```

Figure 105. Volume Reserve Time Report

The monitor collects the data in a dataspace, and an ISPF application allows you to navigate the contents. The following are samples of the ISPF panels:

```
                ENQ/DEQ Monitor V 3.1.0 - Main Menu

Select an option:

__  1.  MAJOR Names                 Date &   Time      : 96.211   10:49
    2.  Resource Name List          Monitor started at : 96.209   16:57
    3.  Volume List                 Elapsed seconds    :         150622
    4.  Filter List                 SMF System ID      :           SC52
-------------------------------------------------------------------------------
GRS Ring -> From: SC42      To: SC49      This: SC52      NUMSYS: 10
-------------------------------------------------------------------------------
                                Time of Delay High. . : 96.211 07:47:22
Global Requests . . . :    34105  Enqueue Delay Hi - Low:     5049    3
Local  Requests . . . :   212085  Enqueue Delay     msec:       22

Major Names . . . . . :       45  ACCELSYS. . . . . . . :        2
Minor Names . . . . . :      612  RESMIL  . . . . . msec:        0
Volumes . . . . . . . :      151  Data Space Used .bytes:   121128    5 %
Number of Events. . . :   493569  Active Filter. . . . .:       08
Lost Events . . . . . :        0  Events Rate  . . . . .:       93
```

Figure 106. ENQ/DEQ Monitor - Main Menu for GRS Ring Configuration

```
                ENQ/DEQ Monitor V 3.0.1 - Main Menu

Select an option:

__  1.  MAJOR Names                 Date &   Time      : 96.211   11:29
    2.  Resource Name List          Monitor started at : 96.209   16:57
    3.  Volume List                 Elapsed seconds    :         152992
    4.  Filter List                 SMF System ID      :           SC52
-------------------------------------------------------------------------------
GRS Star -> Number of Lock Entries:  1048576             NUMSYS: 10
-------------------------------------------------------------------------------
                                Time of Delay High. . : 96.220 14:18:45
Global Requests . . . :    34731  Enqueue Delay Hi - Low:   464347    304
Local  Requests . . . :   215828  Enqueue Delay  mic-sec:      481

Major Names . . . . . :       45
Minor Names . . . . . :      625
Volumes . . . . . . . :      151  Data Space Used .bytes:   123076    6 %
Number of Events. . . :   502327  Active Filter. . . . .:       08
Lost Events . . . . . :        0  Events Rate  . . . . .:       93
```

Figure 107. ENQ/DEQ Monitor - Main Menu for GRS Star Configuration

```
                    ENQ/DEQ Monitor - Major Name List     ROW 16 TO 30

Enter S to select a Major Name for details     .
      L major on command line to locate a Major.   Elapsed seconds:  150622

   Sel.   ----------   ---------   --------   -------  -average-   -Reserved-
   Field  Major Name      RNL       Scope     Counter     msec       seconds
   _      SYSZWTOR     NO           SYSTEM          4
   _      SYSZVVDS     EXCLUDED    *RESERVE      1870       13           24
   _      SYSZVVDS                  SYSTEM        1614
   _      SYSZVARY                  SYSTEM           3
   _      SYSZSDSF                  SYSTEM           3
   _      SYSZRAC2                  SYSTEM        1331
   _      SYSZRACF     CONVERTED   *SYSTEMS       291       42           11
   _      SYSZMCS      NO           SYSTEM           2
   S      SYSZMCS      NO           SYSTEMS         39
   _      SYSZMCS                   SYSTEM       16418
```

Figure 108. ENQ/DEQ Monitor - Major Name List

```
                    ENQ/DEQ Monitor - Major Name List     ROW 31 TO 45

Enter S to select a Major Name for details     .
      L major on command line to locate a Major.   Elapsed seconds:  150622

   Sel.   ----------   ---------   --------   -------  -average-   -Reserved-
   Field  Major Name      RNL       Scope     Counter     msec       seconds
   _      SYSIGGV2     FORCED       RESERVE        41       50            1
   _      SYSIGGV2     CONVERTED   *SYSTEMS      1564       36           58
   _      SYSIEWLP     CONVERTED   *SYSTEMS        16      396            6
   _      SYSIEA01                  SYSTEM           8
   _      SYSDSN                    SYSTEM           2                    24
   _      SYSDSN       INCLUDED    *SYSTEMS       505
   _      SPFEDIT      CONVERTED   *SYSTEMS       120       13
   _      SPFEDIT                   SYSTEMS        234
   _      IGDCDSXS     CONVERTED   *SYSTEMS     14970     1132
   _      ENQDELAY                  SYSTEM         130                    11
   _      BBM/RMGR                  SYSTEM       30179
   _      AUDIT056                  SYSTEM           1
   _      ARCGPA       CONVERTED   *SYSTEMS       378
```

Figure 109. ENQ/DEQ Monitor - Major Name List (continued)

```
                      ENQ/DEQ Monitor - Minor Name List       ROW 1 TO 5

 Minor Name list for:              Major Name  : SYSZMCS
                                   RNL . . . . : NO
                                   Scope . . . : SYSTEMS

 Enter S to select a Minor Name for Jobnames    .
       L min.  on command line to locate a Minor.

   Sel.   ------------------------------------------------  --------
   Field  Minor Name:                                      Counter:
   _      SYSMCS#MCS                                              7
   _      SYSMCS#EMCS                                             7
   S      SYSMCS#CL3                                             10
   _      SYSMCS#CL2                                              7
   _      SYSMCS#CL1                                              8
:****************************** Bottom of data ************************
```

Figure 110. ENQ/DEQ Monitor - Minor Name List

```
                        ENQ/DEQ Monitor - Jobname List         ROW 1 TO 5

 List for Major Name  : SYSZMCS
          Minor Name  : SYSMCS#CL3
          Minor Length: 10
    --------  --------   ----------       -------- ---       ----------
    Job_name  User_ID    Enqs x Job       Pgm_name E/S       Enqs x PGM
    RONN      RONN                1       ISFMAIN   E                 1
    MERONI    MERONI              5       ISFSTOP   E                 3
              MERONI                      ISFMAIN   E                 2
    IOSAS     STC                 1       IOSVISWR  E                 1
    CONSOLE   +CONSOLE            3       IEAVG604  E                 3
 ****************************** Bottom of data ************************
```

Figure 111. ENQ/DEQ Monitor - Jobname List

```
                        ENQ/DEQ Monitor - VOLUME List          ROW 1 TO 7

 Enter S to  select a Volume for details
       A for active Reserves on  Volume
       L volume on command line to locate a Volume
   *   indicates volume where reserves are not converted
-- - ------ ------- Dev. Max  --------- ------------ Reserve Time -------------
S.   Volume Tot.Res nbr   Res Elap(sec)  Avg.(ms)  Min.(ms)  Max.(ms) Tot.(sec)
 S   ITS001  14145  0CDA   01    151642        82        27     86880     11882
 _   MVS002    129  0FD5   04    139516       103         3      2734        12
 _   TOTCAT   1559  0FC7   02    151640         3         3        48         6
 _   TOTPUB    107  0FCD   01     72110        47        22       190         4
 _   TOTSMS     44  0FCE   01    151209        30        24        60         1
 _ * BOOK01      9  0CC6   02    150720        19        17        56         0
 S * IODFPK     99  0CD0   02     11031        19        19        19         0
 ****************************** BOTTOM OF DATA ******************************
```

Figure 112. ENQ/DEQ Monitor - VOLUME List

```
                    ENQ/DEQ Monitor - VOLUME Entry List      ROW 1 TO 2

 Volser. . . . . : ITSO01      Average Reserve Time (ms) : 82
 Tot.nr of Reserve : 141459    Minimum Reserve Time (ms) : 27
 Dev.nr. . . . . : 0CDA        Maximum Reserve Time (ms) : 86880
 Max Reserve Cnt. .: 01        Total   Reserve Time (sec): 11882
 Elapsed Time (sec): 151642    Volume  Reserve Rate (min): 55

 Interval
- Rate -- ----- -------- ----------------------  ------------ Time ------------
S  min.   Count MajName  Minor name (max 22 ch)  Avg ms  Min ms  Max ms Tot sec
_    55  141782 SYSZJES2 ITSO01SYS1.JES2.CKPT2       82      38   86880   11909
_     0       1 SYSVTOC  ITSO01                      27      27      27       0
 ******************************** BOTTOM OF DATA ********************************
```

Figure 113. ENQ/DEQ Monitor - VOLUME Entry List for ITSO01

```
                    ENQ/DEQ Monitor - VOLUME Entry List      ROW 1 TO 3

 Volser. . . . . : IODFPK      Average Reserve Time (ms) : 19
 Tot.nr of Reserve : 99        Minimum Reserve Time (ms) : 19
 Dev.nr. . . . . : 0CD0        Maximum Reserve Time (ms) : 19
 Max Reserve Cnt. .: 02        Total   Reserve Time (sec): 0
 Elapsed Time (sec): 11031     Volume  Reserve Rate (min): 0

 Interval
- Rate -- ----- -------- ----------------------  ------------ Time ------------
S  min.   Count MajName  Minor name (max 22 ch)  Avg ms  Min ms  Max ms Tot sec
_     0      61 SYSZVVDS IODFPK                       7       2     122       0
_     0       1 SYSVTOC  IODFPK                      19      19      19       0
_     0      37 SYSIGGV2 CATALOG.SHRICF1.VIODFP      51      13     178       1
```

Figure 114. ENQ/DEQ Monitor - VOLUME Entry List for IODFPK, Always Reserved

```
                          ENQ/DEQ Monitor - RNL Menu

 Select an option:

 __  1.  RNL SYSTEM Inclusion Table
     2.  RNL SYSTEMS Exclusion Table
     3.  RNL RESERVE Conversion Table
```

Figure 115. ENQ/DEQ Monitor - RNL Menu

```
                  ENQ/DEQ Monitor - SYSTEM Inclusion Table      ROW 1 FROM 24

 RNL SYSTEM Inclusion Table:

 ------------------------------------------------------------------------
 GENERIC     SYSDSN
 GENERIC     SPFEDIT
 GENERIC     SYSIKJUA    SYS1
 GENERIC     SYSZVOLS
 GENERIC     SYSIKJBC
 ******************************** BOTTOM OF DATA ************************
```

Figure 116. ENQ/DEQ Monitor - SYSTEM Inclusion Table

```
                ENQ/DEQ Monitor - SYSTEMS Exclusion Table      ROW 6 FROM 24

RNL SYSTEMS Exclusion Table:

-------------------------------------------------------------------------
SPECIFIC    SYSDSN      PASSWORD
SPECIFIC    SPFEDIT     PASSWORD
SPECIFIC    SYSDSN      SYS1.DCMLIB
SPECIFIC    SPFEDIT     SYS1.DCMLIB
GENERIC     SYSDSN      SYS1.JES3
SPECIFIC    SYSDSN      SYS1.NUCLEUS
SPECIFIC    SPFEDIT     SYS1.NUCLEUS
SPECIFIC    SYSDSN      SYS1.SVCLIB
SPECIFIC    SPFEDIT     SYS1.SVCLIB
GENERIC     SYSZJES2
GENERIC     SYSCTLG
GENERIC     SYSZVVDS                            <--used with GRS RING
GENERIC     SYSVTOC                             <--used with GRS RING
SPECIFIC    SYSIGGV2    UCAT.VBOOK01            <--used with ISGGREX0 user exit
SPECIFIC    SYSIGGV2    UCAT.VOS3R2B            <--used with ISGGREX0 user exit
SPECIFIC    SYSIGGV2    CATALOG.SHRICF1.VIODFPK <--used with ISGGREX0 user exit
SPECIFIC    SYSIGGV2    CATALOG.SHRICF1.VIODFBK <--used with ISGGREX0 user exit
****************************** BOTTOM OF DATA ************************
```

Figure 117. ENQ/DEQ Monitor - SYSTEMS Exclusion Table

```
                ENQ/DEQ Monitor - RESERVE Conversion Table    ROW 16 FROM 24

RNL RESERVE Convertion Table:

-------------------------------------------------------------------------
GENERIC     SYSZRACF
GENERIC     SYSIEWLP
GENERIC     SPZAPLIB
GENERIC     SYSIGGV2
GENERIC     IGDCDSXS
GENERIC     DSPURI01
GENERIC     SPFEDIT
GENERIC     ARCGPA
GENERIC     SYSZVVDS                            <--used with GRS STAR
GENERIC     SYSVTOC                             <--used with GRS STAR
SPECIFIC    HWRESERV    IODFPK                  <--used with ISGGREX0 user exit
SPECIFIC    HWRESERV    IODFBK                  <--used with ISGGREX0 user exit
SPECIFIC    HWRESERV    BOOK01                  <--used with ISGGREX0 user exit
SPECIFIC    HWRESERV    BOOK02                  <--used with ISGGREX0 user exit
SPECIFIC    HWRESERV    OS3R2B                  <--used with ISGGREX0 user exit
****************************** BOTTOM OF DATA ************************
```

Figure 118. ENQ/DEQ Monitor - RESERVE Conversion Table

Appendix C. ISGGREX0 Sample Exit

When some DASD devices are shared with one or more MVS systems outside a global resource serialization complex, it is mandatory, for data integrity reasons, to have *all* RESERVE/DEQ requests for the devices result in a hardware reserve/release. For example, a generic entry in a CONVERSION RNL for QNAME(SYSVTOC) converts all VTOC serialization requests to global ENQ requests and cannot be tolerated in this environment.

The attached ISGGREX0 exit is based on the ISGGREXS exit in SYS1.SAMPLIB. The sample exit detects volumes shared by systems outside the global resource serialization complex and prevents RESERVE conversion for those volumes.

The shared “special” volumes are defined to the exit by adding a specific entry in the RNL conversion table with an unused QNAME and the shared volser as the RNAME, for example:

```
RNLDEF RNL(CON) TYPE(SPECIFIC)
QNAME(HWRESERV)
RNAME(VOLSER)
```

When a RESERVE/DEQ request is issued, the exit gets control and verifies whether the target volume for the RESERVE/DEQ request matches the special QNAME(HWRESERV) RNAME volser. If a match is found, return code 4 is passed back to the caller. Return code 4 indicates that global resource serialization should *not* convert the RESERVE request (a SYSTEMS scope ENQ and the hardware reserve are to be issued). The target volser is found using the UCB address passed in the input parameters. If the RESERVE/DEQ target volser was not one of the special QNAME(HWRESERV) RNAME(VOLSER) volumes, normal CONVERSION RNL scan processing takes place.

Because the CONVERSION RNL table is used, any change to the table can be activated through operator commands.

To install the exit see *OS/390 MVS INSTALLATION EXITS*.

Note: Because this sample exit is using a qname - rname combination that is not use by the system, it can be installed and activated without having to cold start the sysplex. The suggested installation sequence is the following:

- Assemble and linkedit the sample ISGGREX0 exit into the MVS nucleus.
- Stop and re-IPL all the MVS systems in the sysplex, one by one. It is not required to stop the sysplex.
- Update the RNL conversion table with specific entries, qname(HWRESERV) rname(volser), for shared volumes.
- If the shared volumes have VSAM catalog, update the RNL exclusion table with specific entries, qname(SYSIGGV2) rname(catname), for the catalogs. For more information see following note.
- Check that the shared volumes have no activity, and that they be logically offline.
- Dynamically activate the updated RNL table using command SET GRSRNL=xx.

Note: The volumes dynamically added or deleted to the QNAME(HWRESRV) with the RNL update should not be in use; they should be offline.

Note: If the shared volumes have VSAM catalogs, it is also recommended to place specific SYSIGGV2 entries for the catalog names in SYSTEMS exclusion RNL.

When using specific entries for SYSIGGV2 and the catalog name is less then 20 bytes, pad it to 20 bytes with blanks; if the name is more than 20, pad it to 44 bytes. If you have a catalog naming convention in place, you can use catalog name prefixes and generic entries. In the following example, the third entry can be used to replace the first two entries; a naming convention should be in place.

```
RNLDEF RNL(EXCL) TYPE(SPECIFIC)
QNAME(SYSIGGV2)
RNAME('CATALOG.SHRICF1.VIODFPK                    ') <-- padded to 44 bytes

RNLDEF RNL(EXCL) TYPE(SPECIFIC)
QNAME(SYSIGGV2)
RNAME('CATALOG.SHRICF1.VIODFPK                    ') <-- padded to 44 bytes

RNLDEF RNL(EXCL) TYPE(GENERIC)
QNAME(SYSIGGV2)
RNAME(CATALOG.SHRICF1)
```

Figure 119 shows the conversion exit logic.

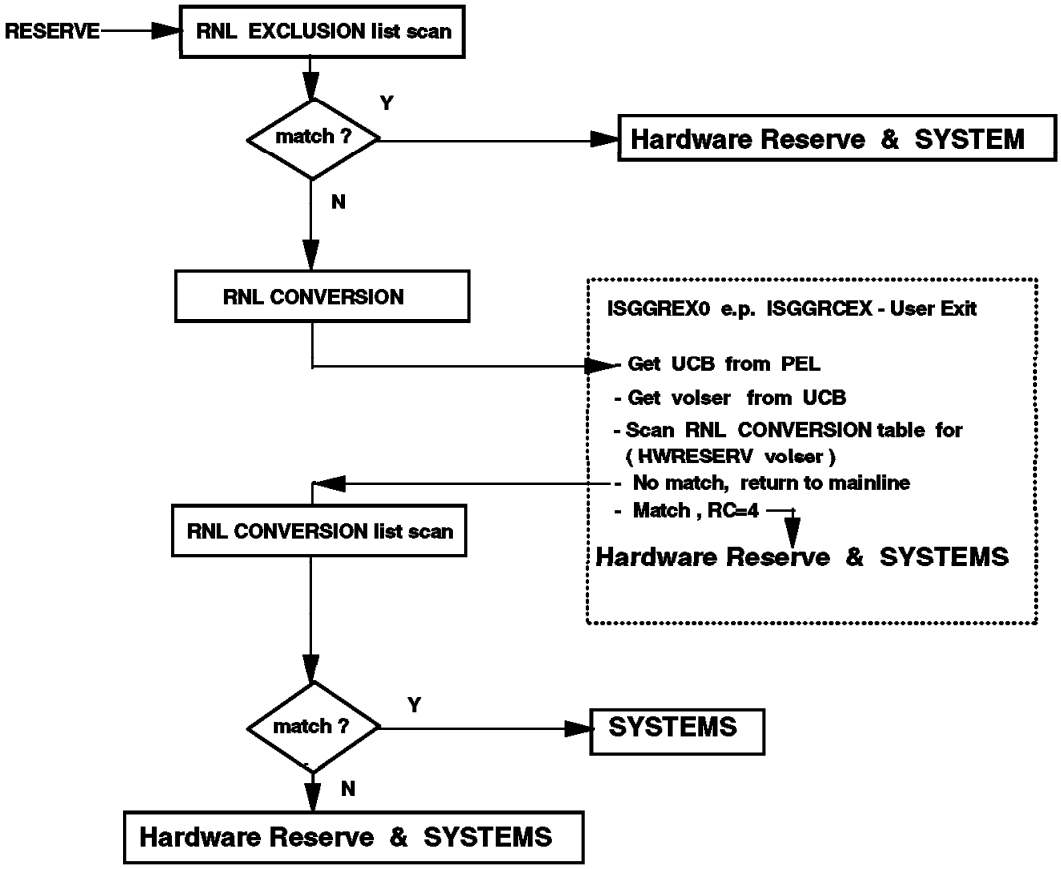


Figure 119. Conversion Exit Logic

```
        TITLE 'ISGGREXO - GRS RESOURCE EXIT ROUTINE'
ISGGREXO CSECT
*/*  START OF SPECIFICATIONS ****
*--------------------------------------------------------------------*
* ALWAYS RESERVE FOR VOLUMES SHARED WITH MVS SYSTEMS OUTSIDE         *
* THE GRS RING.                                                      *
*                                                                    *
* THIS ISGGREXO EXAMPLE IS BASED ON THE ISGGREXS IN THE              *
* SYS1.SAMPLIB, WITH SUPPORT FOR SUPPRESSION OF TEMPORARY            *
* DATA SETS IN GLOBAL SHARING REMOVED.                               *
*                                                                    *
* THE LINE OF CODE MODIFIED OR ADDED ARE INDICATED WITH      *AAA *
*                                                                    *
*--------------------------------------------------------------------*
* LOGIC:                                                             *
*                                                                    *
* IT IS MANDATORY, TO SUPPORT SHARED DASD WITH MVS SYSTEMS           *
* OUTSIDE GRS, THAT ALL RESERVE REQUESTS THAT ADDRESS THESE          *
* VOLUMES RESULT IN AN H/W RESERVE WHATEVER THE RESOURCE NAME IS.    *
*                                                                    *
* THE SAME RESOURCES, THAT ADDRESS OTHER VOLUMES, SHOULD HAVE THE    *
* POSSIBILITY TO BE FILTERED THROUGH THE CONVERSION RNLS AND HAVE    *
* THE HARDWARE RESERVE ELIMINATED.                                   *
*                                                                    *
* BY ADDING TO THE CONVERSION TABLE A QNAME NOT USED BY THE          *
* SYSTEM AND RNAMES THAT IDENTIFY THE VOLUMES TO SHARE OUTSIDE       *
* GRS, THE RESERVE REQUEST FOR THE VOLUMES INCLUDED IN THE           *
* FOLLOWING DEFINITION WILL ALWAYS RESULT IN A HARDWARE RESERVE.     *
*                                                                    *
*      RNLDEF RNL(CON) TYPE(SPECIFIC)                                *
*      QNAME(HWRESERV)                                               *
*      RNAME(VOLSER)                                                 *
*                                                                    *
* ISGGRCEXO RECEIVES CONTROL FOR ALL RESERVE/DEQ  REQUESTS           *
* (SYSTEMS+H/W RESERVE), LOCATES THE UCBVOLI (VOLSER) AND CHECKS     *
* IF THE RESOURCE 'HWRESERV' 'VOLSER' IS PRESENT IN THE 'CON'        *
* TABLE. IF FOUND RETURNS RC=4 TO GRS THAT LEAVES THE RESOURCE       *
* AS IS (SYSTEMS+H/W RESERVE). IF NOT NORMAL RNL SCAN IS RESUMED.    *
*                                                                    *
* BECAUSE THE CONVERSION RNL TABLE IS USED, ANY CHANGE TO THE        *
* TABLE CAN BE ACTIVATED THROUGH OPERATOR COMMANDS.                  *
*                                                                    *
* NOTE: THE VOLUME(S) BEHIND THE QNAME(HWRESERV) SHOULD NOT BE       *
* DYNAMICALLY CHANGED UNLESS THE DEVICES ARE OFFLINE.                *
*                                                                    *
* THE NAME 'HWRESERV' IS HARD-CODED AND IS DEFINED AT LABEL          *
* HRDWNAME.                                                          *
*                                                                    *
*--------------------------------------------------------------------*
*                CONVERSION LIST EXAMPLE                             *
*--------------------------------------------------------------------*
*                                                                    *
* RNLDEF RNL(CON) TYPE(GENERIC)                                      *
* QNAME(SYSVTOC)                                                     *
*                                                                    *
* RNLDEF RNL(CON) TYPE(GENERIC)                                      *
* QNAME(SYSIGGV2)                                                    *
*                                                                    *
* RNLDEF RNL(CON) TYPE(SPECIFIC)                                     *
* QNAME(HWRESERV)                    /*SPECIAL NAME*/              *
* RNAME(XA9RES)                      /*ALWAYS RESERV  XA9RES*/     *
*                                                                    *
* RNLDEF RNL(CON) TYPE(SPECIFIC)                                     *
* QNAME(HWRESERV)                    /*SPECIAL NAME*/              *
* RNAME(CIX321)                      /*ALWAYS RESERV  CIX321*/     *
*                                                                    *
**********************************************************************
***                        END OF                                  *
***   RESERVE CONVERSION RESOURCE NAME LIST - SAMPLE               *
**********************************************************************
*--------------------------------------------------------------------
*
*
*01*  MODULE-NAME = ISGGREXO
*
*02*    CSECT-NAME = ISGGREXO
*
*01*  DESCRIPTIVE-NAME = GRS RESOURCE EXIT ROUTINE
*
*01*  COPYRIGHT =
*        5740-XC6 COPYRIGHT IBM CORP 1981,
*        LICENSED MATERIAL-PROGRAM, PROPERTY OF IBM,
*        REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083.
```

```
*
*01* STATUS = OS/VS2 HBB2102
*
*01*  FUNCTION =
*               TO SCAN THE INPUT RESOURCE NAME LIST (RNL) FOR
*               THE RESOURCE NAME SPECIFIED IN THE INPUT PEL.
*               THE RETURN CODE INDICATES WHETHER OR NOT THE
*               INPUT RESOURCE NAME IS CONTAINED IN THE RNL.
*
*02*    OPERATION =
*                 THE INPUT RESOURCE NAME LIST IS SEARCHED AS
*                 FOLLOWS:
*
*                 1.  COMPARE THE INPUT ARGUMENT TO THE RNL ENTRY.
*
*                 2.  IF A MATCH IS FOUND, SET INDICATIVE RETURN
*                     CODE.
*
*                 3.  IF A MATCH IS NOT FOUND, INDEX TO NEXT RNL
*                     ENTRY AND CONTINUE SCAN.
*
*                 4.  IF THE ENTIRE RNL IS SEARCHED WITHOUT A
*                     MATCH, SET INDICATIVE RETURN CODE.
*
*01*  NOTES =
*            1.  TO FACILITATE REPLACEMENT OF THE RESOURCE EXITS
*                PROVIDED BY GRS, THIS ROUTINE HAS BEEN DESIGNED
*                TO SCAN ANY OF THE 3 GRS RNL'S. THOUGH ONE
*                ROUTINE IS PROVIDED TO PERFORM THE SCAN, IT IS
*                INVOKED USING 3 DIFFERENT EXTERNAL NAMES:
*
*                ISGGSIEX - INVOKED TO SCAN THE SYSTEM (SCOPE)
*                           INCLUSION RESOURCE NAME LIST.
*
*                ISGGRCEX - INVOKED TO SCAN THE RESERVE
*                           CONVERSION RESOURCE NAME LIST.
*
*                ISGGSEEX - INVOKED TO SCAN THE SYSTEMS (SCOPE)
*                           EXCLUSION RESOURCE NAME LIST.
*
*                EACH OF THESE EXTERNAL ENTRY POINT NAMES AND
*                ITS CORRESPONDING RNL ADDRESS HAS BEEN
*                DEFINED IN THE GVT.  THE ENQ/DEQ/RESERVE MAINLINE
*                ROUTINE INVOKES THE APPROPRIATE EXIT ROUTINE
*                (E.G., RESERVE PROCESSING INVOKES ISGGRCEX)
*                ALTHOUGH EACH ENTRY POINT MAPS TO THE COMMON
*                SCAN ROUTINE PROVIDED BY GRS.  THEREFORE THE
*                INSTALLATION CAN REPLACE THE EXIT ROUTINE
*                AND/OR RNL(S) WITHOUT CHANGES TO THE GVT OR
*                THE ENQ/DEQ/RESERVE MAINLINE ROUTINE.
*            2.  MODULE ISGGREXO CAN BE REPLACED WITH WHATEVER
*                AMODE OR RMODE THE INSTALLATION WISHES. THE
*                DEFAULTS USED ARE AMODE(31) AND RMODE(ANY).  @G860PWE
*
*            3.  EACH PROCESSOR IN THE GRS RING MUST HAVE
*                IDENTICAL RNL'S.  ALSO, THE EXIT ROUTINE
*                IN EACH GRS SYSTEM MUST YIELD THE SAME SCAN
*                RESULTS.  REFERENCE ISGCQMRG FOR FURTHER
*                INFORMATION.
*
*            4.  SERIALIZATION WHEN INVOKED AS ISGGSIEX,
*                ISGGSEEX, OR ISGGRCEX IS THE LOCAL LOCK
*                OF THE INVOKER'S ADDRESS SPACE AND
*                CMSEQDQ.
*
*02*    DEPENDENCIES =
*                1.  THE RECOVERY ROUTINE FOR THIS MODULE,
*                    ISGGFRRO, DEPENDS ON THE FOLLOWING LABELS
*                    TO APPEAR IN THIS MODULE:
*                       ISGGREXO  CSECT NAME
*                       ISGGREXE  END OF MODULE ISGGREXO
*                       ISGGREXM  LABEL ON THE MODID MACRO
*
*                    IF THE USER REPLACES THIS MODULE, IT MUST
*                    BE INSURED THAT THE ABOVE LABELS APPEAR
*                    IN THE APPROPRIATE LOCATIONS IN THE
*                    EXIT ROUTINE.
*
*                2.  THIS MODULE WILL FUNCTION ON ALL HARDWARE
*                    AND SOFTWARE CONFIGURATIONS WHICH SATISFY
*                    THE REQUIREMENTS OF THE PRODUCT IDENTIFIED
*                    IN THE STATUS FIELD.
*                3.  ANY INPUT ADDRESSES TO ISGGREXO MUST BE
```

```
*                  24-BIT ADDRESSES TO ALLOW THE MODULE TO
*                  BE REPLACED IN ANY AMODE OR ANY RMODE AN
*                  INSTALLATION WISHES.                    @G860PWE
*
*02*    CHARACTER-CODE-DEPENDENCIES = EBCDIC CHARACTER SET
*
*02*    RESTRICTIONS = NONE.
*
*02*    REGISTER-CONVENTIONS =
*
*03*      REGISTERS-SAVED =
*                        R0 - R12, R14, R15
*
*03*      REGISTER-USAGE =
*               R0     = CONTAINS RNL ADDRESS ON ENTRY. CONTAINS
*                        RETURN CODE WHILE SCANNING LISTS
*               R1     = ADDRESS OF A PEL ON ENTRY
*               R2     = USED TO ADDRESS RNLES
*               R3-R12 = USAGE UNPREDICTABLE
*               R13    = SAVE AREA ADDRESS
*               R14    = RETURN ADDRESS
*               R15    = ENTRY POINT ADDRESS/BASE REGISTER.
*                        UPON EXIT, RETURN CODE.
*
*03*      REGISTERS-RESTORED =
*                           R0 - R12, R14
*
*02*    PATCH-LABEL = NONE (NUCLEUS RESIDENT - USE IEAPATCH)
*
*01*  MODULE-TYPE = PROCEDURE
*
*02*    PROCESSOR = PL/S-III
*
*02*    MODULE-SIZE = SEE EXTERNAL SYMBOL DICTIONARY
*
*02*    ATTRIBUTES =
*
*03*      LOCATION      = NUCLEUS
*03*      STATE         = SUPERVISOR
*03*      KEY           = 0
*03*      MODE          =
*
*04*        WORK UNIT = TASK
*04*        HASID     = ANY
*04*        PASID     = ANY
*04*        SASID     = ANY
*04*        SAC       = ON OR OFF
*04*        AMODE     = 31
*04*        RMODE     = ANY
*
*03*      SERIALIZATION = SERIALIZATION WHEN INVODED AS ISGGSIEX,
*                        ISGGSEEX, OR ISGGRCEX IS THE LOCAL LOCK
*                        OF THE INVOKER'S ADDRESS SPACE AND
*                        CMSEQDQ.
*
*03*      TYPE          = REENTRANT
*
*01*  ENTRY-POINT = ISGGSIEX (ENTRY POINT DEFINED IN GVTGSIEX)
*
*02*    PURPOSE = TO DETERMINE WHETHER THE INPUT RESOURCE NAME
*                 SHOULD BE INCLUDED IN GLOBAL SHARING BY
*                 SCANNING THE SYSTEM (SCOPE) INCLUSION RNL.
*
*03*      OPERATION = SEE MODULE OPERATION SECTION
*
*02*    LINKAGE = BALR
*
*03*      CALLERS = ISGGQWBI, ISGLNQDQ
*
*03*      ENTRY-REGISTERS =
*                  R0     = ADDRESS OF SYSTEM INCLUSION
*                           RESOURCE NAME LIST
*                  R1     = ADDRESS OF A PEL
*                  R2-R12 = UNDEFINED
*                  R13    = SAVEAREA ADDRESS
*                  R14    = RETURN ADDRESS
*                  R15    = ENTRY POINT ADDRESS
*
*02*    INPUT =
*                R0 = ADDRESS OF RNL TO BE SEARCHED
*                R1 = PEL ENTRY  CONTAINING RESOURCE NAME TO BE
*                     USED AS THE SEARCH ARGUMENT.
*
```

```
*02*    OUTPUT =
*               R15 = 0 - NAME FOUND
*                     4 - NAME NOT FOUND
*
*02*    EXIT-NORMAL = RETURN TO CALLER VIA BR14.
*
*03*      CONDITIONS = INPUT LIST SCANNED AS REQUESTED.
*
*04*        EXIT-REGISTERS =
*                 R0-R12 = AS ON INPUT
*                 R13    = SAVEAREA ADDRESS
*                 R14    = RETURN ADDRESS
*                 R15    = RETURN CODE
*
*03*      RETURN-CODES =
*                       R15 = 0 - NAME FOUND
*                             4 - NAME NOT FOUND
*
*02*    EXIT-ERROR = NONE
*
*02*    WAIT-STATE-CODES = NONE
*
*01*  ENTRY-POINT = ISGGRCEX (ENTRY POINT DEFINED IN GVTGRCEX)
*
*02*    PURPOSE = TO DETERMINE WHETHER THE INPUT RESOURCE NAME
*                 SHOULD BE CONVERTED FROM A HARDWARE RESERVE
*                 RESOURCE NAME TO A GRS CONTROLLED GLOBAL ENQ
*                 RESOURCE NAME BY SCANNING THE RESERVE CONVERSION
*                 RNL.
*
*03*      OPERATION = SEE MODULE OPERATION SECTION.
*
*02*    LINKAGE = BALR
*
*03*      CALLERS = ISGGQWBI
*
*03*      ENTRY-REGISTERS =
*                   R0     = ADDRESS OF SYSTEM RESERVE CONVERSION
*                            RESOURCE NAME LIST
*                   R1     = ADDRESS OF A PEL
*                   R2-R12 = UNDEFINED
*                   R13    = SAVEAREA ADDRESS
*                   R14    = RETURN ADDRESS
*                   R15    = ENTRY POINT ADDRESS
*
*02*    INPUT =
*                 R0 = ADDRESS OF RNL TO BE SEARCHED
*                 R1 = PEL ENTRY  CONTAINING RESOURCE NAME TO BE
*                      USED AS THE SEARCH ARGUMENT.
*
*02*    OUTPUT =
*                R15 = 0 - NAME FOUND
*                      4 - NAME NOT FOUND
*
*02*    EXIT-NORMAL = RETURN TO CALLER VIA BR14.
*
*03*      CONDITIONS = INPUT LIST SCANNED AS REQUESTED.
*
*04*        EXIT-REGISTERS =
*                   R0-R12 = AS ON INPUT
*                   R13    = SAVEAREA ADDRESS
*                   R14    = RETURN ADDRESS
*                   R15    = RETURN CODE.
*
*03*      RETURN-CODES =
*                       R15 = 0 - NAME FOUND
*                             4 - NAME NOT FOUND
*
*02*    EXIT-ERROR = NONE
*
*02*    WAIT-STATE-CODES = NONE
*
*01*  ENTRY-POINT = ISGGSEEX  (ENTRY POINT DEFINED IN GVTGSEEX)
*
*02*    PURPOSE = TO DETERMINE WHETHER THE INPUT RESOURCE NAME
*                 SHOULD BE EXCLUDED FROM GLOBAL SHARING BY
*                 SCANNING THE SYSTEMS (SCOPE) EXCLUSION RNL.
*
*03*      OPERATION = SEE MODULE OPERATION SECTION.
*
*02*    LINKAGE = BALR
*
*03*      CALLERS = ISGGQWBI
```

```
*
*03*      ENTRY-REGISTERS =
*                  R0     = ADDRESS OF SYSTEMS EXCLUSION RESOURCE
*                           NAME LIST
*                  R1     = ADDRESS OF A PEL
*                  R2-R12 = UNDEFINED
*                  R13    = SAVEAREA ADDRESS
*                  R14    = RETURN ADDRESS
*                  R15    = ENTRY POINT ADDRESS
*
*02*    INPUT =
*             R0 = ADDRESS OF RNL TO BE SEARCHED
*             R1 = PEL ENTRY  CONTAINING RESOURCE NAME TO BE
*                  USED AS THE SEARCH ARGUMENT.
*
*02*    OUTPUT =
*              R15 = 0 - NAME FOUND
*                    4 - NAME NOT FOUND
*
*02*    EXIT-NORMAL = RETURN TO CALLER VIA BR14.
*
*03*      CONDITIONS = INPUT LIST SCANNED AS REQUESTED.
*
*04*        EXIT-REGISTERS =
*                  R0-R12 = AS ON INPUT
*                  R13    = SAVEAREA ADDRESS
*                  R14    = RETURN ADDRESS
*                  R15    = RETURN CODE.
*
*03*      RETURN-CODES =
*                       R15 = 0 - NAME FOUND
*                             4 - NAME NOT FOUND
*
*02*    EXIT-ERROR = NONE
*
*02*    WAIT-STATE-CODES = NONE
*
*01*  EXTERNAL-REFERENCES = NONE.
*
*02*    ROUTINES = NONE.
*
*02*    DATA-AREAS = NONE
*
*02*    CONTROL-BLOCKS =
*                       PEL       R
*                       RNLE      R
*
*01*  TABLES =
*             1. SYSTEMS (SCOPE) INCLUSION RNL (ISGGIRNL)
*             2. SYSTEM  (SCOPE) EXCLUSION RNL (ISGGERNL)
*             3. RESERVE CONVERSION RNL        (ISGGCRNL)
*
*01*  MACROS-EXECUTABLE =
*                          MODID
*
*02*    SERIALIZATION = NONE.
*
*01*  CHANGE-ACTIVITY = SUPPORTS THE FOLLOWING PRODUCTS:
*                       $L0=FUNCT,JBB1326,810323,PDYC: VERSION OF @L0A
*                                ISGGREX0 FOR SAMPLIB            @L0A
*                       HBB2102  NEW VERSION OF ISGGREX0 PLUS @G860PYC
*                                SUPPORT FOR TEMPORARY D.S.   @G860PYC
*
*                      SUPPORTS THE FOLLOWING PTMS:
*                       PBB0618 ADD RECOVERY SUPPORT          @ZMB0618
*                       PBB0854 REMOVE ISGGRNLV ENTRY PT.     @ZMB0854
*                       PCC2507 UPDATE COPYRIGHT FIELD AND    @ZMC2507
*                               ADD A STATUS FIELD            @ZMC2507
*
*01*  MESSAGES = NONE.
*
*01*  ABEND-CODES = NONE.
*
*01*  SYSGEN = INCLUDED FROM AOSC5 INTO IEANUC01 BY SGISG300
*
**** END OF SPECIFICATIONS **                                        */
         EJECT
*********************************************************************
*                                                                   *
*        REGISTER ASSIGNMENTS                                       *
*                                                                   *
*********************************************************************
```

```
         SPACE
RNLSTART EQU   0                     UPON ENTRY, START OF RNL
RTRNCODE EQU   0                     MATCH/NO-MATCH INDICATOR
PELPTR   EQU   1                     ADDRESS OF THE PEL
RNLEPTR  EQU   2                     POINTER TO AN RNL ENTRY (RNLE)
CHARPTR  EQU   3                     ADDR OF CHARACTER BEING TESTED
FLENRNLE EQU   3                     LENGTH OF FIXED PART OF AN RNLE
RNAMELEN EQU   3                     LENGTH OF RNAME
WORKREG  EQU   12                    LENGTH OF FIXED + VARIABLE
REG13    EQU   13                    SAVE AREA ADDRESS
REG14    EQU   14                    RETURN ADDRESS
REG15    EQU   15                    USED AS BASE REG UNTIL EXIT,
*                                    THEN CONTAINS THE RETURN CODE
R0       EQU   0                     START DI RNL
R1       EQU   1                     ADDRESS OF PEL
R2       EQU   2                     POINTER TO AN RNLE
R3       EQU   3                     LENGTH OF RNAME
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13                    SAVE AREA POINTER
R14      EQU   14                    RETURN ADDRESS
R15      EQU   15                    BASE REGISTER UNTIL EXIT
         SPACE 5
*********************************************************************
*                                                                   *
*        CONSTANTS                                                  *
*                                                                   *
*********************************************************************
         SPACE
FOUND    EQU   0                     RESOURCE NAME IS IN THE RNL
NOTFOUND EQU   4                     RESOURCE NAME IS NOT IN THE RNL
ZERO     EQU   0                     USED FOR LENGTH TEST
         EJECT
*
ISGGREXO AMODE 31                                          @G86OPYC
ISGGREXO RMODE ANY                                         @G86OPYC
         MODID BR=NO,MODLBL=ISGGREXM
MAINENT  DS    0H
         USING *,15
         ENTRY ISGGSIEX
         ENTRY ISGGSEEX
         ENTRY ISGGRCEX
         ENTRY ISGGREXE              END OF MODULE
         ENTRY ISGGREXM              MODULE INFORMATION
*-------------------------------------------------------------  *AAA
* CODE TO SUPPORT 'ALWAYS RESERVE' FOR VOLUMES SHARED WITH MVS  *AAA
* SYSTEMS OUTSIDE THE GRS COMPLEX.                              *AAA
*-------------------------------------------------------------  *AAA
* CONVERSION ENTRY                                             *AAA
*                                                              *AAA
* FUNCTION                                                     *AAA
*  CHECK REQUEST FOR RESERVE ===>SYSTEMS + H/W-RESERVE         *AAA
*  FIND VOLSER FROM UCB POINTED BY PEL                         *AAA
*  CHECK IF CONVERSION TABLE HAS A SPECIFIC ENTRY FOR          *AAA
*                                                              *AAA
*        QNAME(HWRESERV)                                       *AAA
*        RNAME(VOLSER)                                         *AAA
*                                                              *AAA
*  IF MATCH FOUND RETURN CODE = 4 ==> SYSTEMS + H/W-RESERVE    *AAA
*  IF NOT, RESTORE REGISTERS AND CONTINUE NORMALLY             *AAA
*-------------------------------------------------------------  *AAA
ISGGRCEX DS    0H               *AAA  SCAN RESERVE CONVERSION RNL
         SPACE
         STM   14,12,12(13)     *AAA  SAVE ENTRY REGS
         USING RNLE,RNLEPTR     *AAA  ADDRESSABILITY FOR RNL ENTRIES
         USING PEL,PELPTR       *AAA  PEL ENTRY ADDRESSABILITY
         LR    RNLEPTR,RNLSTART *AAA  SET RNLE PTR TO START OF RNL
         LA    RTRNCODE,NOTFOUND *AAA  R0 = 4   NO MATCH
*
         TM    PELFLAG,PELSCPE2 *AAA  SYSTEMS+UCB
         BZ    NOHARDW          *AAA
         L     R1,PELXUCBA      *AAA  UCB POINTER CLEAN,
*                                     HIGH BYTE NOT USED
         DROP  PELPTR           *AAA
*
```

```
* R1 = UCB POINTER USED FOR  RNAME (VOLSER)
*
         SPACE
COMPRNL1 EQU   *                *AAA  COMPARE PEL NAME TO RNL ENTRY
         TM    RNLEFLGS,RNLELAST *AAA  AT END OF THE RNL ?
         BO    NOHARDW           *AAA  YES => RNL SCAN DONE
         SPACE
* COMPARE QNAME HWRESERV   TO THE ONE IN THE RNLE
         CLC   HRDWNAME,RNLEQNME *AAA  CHECK IF QNAME HWRESERV
         BNE   NEXTRNL1          *AAA  NO => GET NEXT RNL ENTRY
         SPACE
* COMPARE THE VOLSER
         SLR   RNAMELEN,RNAMELEN *AAA  CLEAR WORK REG
         IC    RNAMELEN,RNLERNML *AAA  GET LENGTH OF RNAME IN THE RNLE
         BCTR  RNAMELEN,ZERO     *AAA  ADJUST LENGTH
         EX    RNAMELEN,COMPRNM1 *AAA  COMPARE THE RNAME (VOLSER)
         BNE   NEXTRNL1          *AAA  UNEQUAL => GET NEXT RNLE
         SPACE
* INDICATE RESOURCE NAME FOUND IN THE RNL
         LA    RTRNCODE,NOTFOUND *AAA  INDICATE NO CONVERSION
*                                      FOR THIS UCB
         B     MODEXIT           *AAA  EXIT PROCESSING COMPLETE
         SPACE
* GET THE ADDRESS OF THE NEXT RNL ENTRY
NEXTRNL1 EQU   *
         LA    FLENRNLE,RNLERNME-RNLE  LENGTH OF FIXED PART OF RNLE
         SLR   WORKREG,WORKREG   *AAA  CLEAR WORK REG
         IC    WORKREG,RNLERNML  *AAA  GET RNAME LENGTH (VARIABLE)
         ALR   WORKREG,FLENRNLE  *AAA  ADD FIXED + VARIABLE LENGTHS
         ALR   RNLEPTR,WORKREG   *AAA  GET ADDRESS OF NEXT RNL ENTRY
         B     COMPRNL1          *AAA  CHECK THE NEW RNL ENTRY
         EJECT
*--------------------------------------------------------------------
* R0, R1, DA RIPRISTINARE
* R13 SAVE AREA, R14, RETURN ADDRESS, R15 BASE
*--------------------------------------------------------------------
NOHARDW  EQU   *                *AAA
         DROP  RNLEPTR
         LM    R0,R12,20(R13)    *AAA  RIPRISTINO REGISTRI
         B     SECENTRY          *AAA  A CODIFICA NORMALE
*
*--FINE MODIFICA SUPPORTO DISCHI CONDIVISI FUORI SYSPLEX  *AAA-------
**********************************************************************
         EJECT
ISGGSIEX DS    OH                     SCAN INCLUSION RNL
ISGGSEEX EQU   *                      SCAN EXCLUSION RNL
*SGGRCEX EQU   *                      SCAN RESERVE CONVERSION RNL *AAA
         SPACE 3
**********************************************************************
*                                                                    *
*        LOGIC FLOW FOR ISGGREXO                                      *
*                                                                    *
**********************************************************************
/*                                                                   */
/*++ 'ISGGREXO':  ENTRY TO RNL EXIT PROCESSING                       */
/*++ ESTABLISH ADDRESSABILITY                                        */
/*++ SET RETURN CODE TO NOT-FOUND                                    */
/*++ DO WHILE MATCH NOT FOUND AND NOT LAST ENTRY IN THE RNL          */
/*++   IF THE PEL EXTENSION QNAME EQUALS THE QNAME IN THE RNL ENTRY */
/*++     IF THE RNL ENTRY IS GENERIC                                 */
/*++       IF THE LENGTH OF THE RNAME IN THE RNL ENTRY IS ZERO       */
/*++         SET RETURN CODE TO FOUND                                */
/*++       ELSE  (RNLE RNAME LENGTH IS NOT ZERO)                     */
/*++         IF THE PEL RNAME LENGTH IS GREATER THAN OR EQUAL TO
*               THE RNAME LENGTH IN THE RNL ENTRY                   */
/*++           IF THE RNAME IN THE RNLE MATCHES THE ONE IN THE PEL  */
/*++             SET RETURN CODE TO FOUND                          */
/*++           ENDIF    (END OF RNAME COMPARISON)                   */
/*++         ENDIF    (END OF COMPARISON OF RNAME LENGTHS)         */
/*++       ENDIF  (END OF CHECK FOR RNAME LENGTH OF ZERO)           */
/*++     ELSE  (RNL ENTRY IS NON-GENERIC)                           */
/*++       IF THE RNAME LENGTH IN THE RNL EQUALS THE RNAME LENGTH
*              IN THE PEL                                           */
/*++         IF THE RNAME IN THE RNLE MATCHES THE ONE IN THE PEL    */
/*++           SET RETURN CODE TO FOUND                             */
/*++         ENDIF  (END OF RNAME COMPARISON)                       */
/*++       ENDIF  (END OF COMPARISON OF RNAME LENGTHS)              */
/*++     ENDIF  (END OF GENERIC/NON-GENERIC CHECK)                  */
/*++   ENDIF  (END OF QNAME COMPARISON)                             */
/*++   GET NEXT RNL ENTRY                                           */
/*++ ENDDO  (REPEAT SEQUENCE UNTIL MATCH FOUND OR END OF RNL)       */
/*++ RETURN                                                         */
/*++ END 'ISGGREXO'                                                 */
```

```
*/*                                                                   */
*/*********************************************************************/
         EJECT
*************************************************************************
*                                                                       *
*  1. SAVE ENTRY REGISTERS                                              *
*                                                                       *
*  2. ESTABLISH CONTROL BLOCK ADDRESSABILITY                            *
*                                                                       *
*  3. CHECK FOR END OF RNL.  IF END, GO TO STEP 15.                     *
*                                                                       *
*  4. COMPARE QNAME IN PEL TO RNLE.  IF NOT EQUAL, GO TO STEP 14.       *
*                                                                       *
*  5. CHECK FOR GENERIC ENTRY IN THE RNLE.  IF NOT, GO TO STEP 11.      *
*                                                                       *
*  6. CHECK FOR AN RNAME PRESENT IN THE RNLE.  IF YES, GO TO STEP 8.    *
*                                                                       *
*  7. QNAMES MATCH + GENERIC ENTRY + NO RNAME IN THE RNLE               *
*     => THE NAME IN THE PEL MATCHES.  GO TO STEP 24.                   *
*                                                                       *
*************************************************************************
         SPACE
SECENTRY EQU   *                                          *AAA
         BALR  R15,0                                      *AAA
         DROP  R15
         USING *,R15                                      *AAA
         STM   14,12,12(13)            SAVE ENTRY REGS
         USING RNLE,RNLEPTR            ADDRESSABILITY FOR RNL ENTRIES
         USING PEL,PELPTR              PEL ENTRY ADDRESSABILITY
         LR    RNLEPTR,RNLSTART        SET RNLE PTR TO START OF RNL
         LA    RTRNCODE,NOTFOUND       ASSUME NO MATCH WILL BE FOUND
         SPACE
COMPRNLE EQU   *                      COMPARE PEL NAME TO RNL ENTRY
         TM    RNLEFLGS,RNLELAST       AT END OF THE RNL ?
         BO    ENDOFRNL                YES => RNL SCAN DONE
         SPACE
* COMPARE QNAME IN THE PEL TO THE ONE IN THE RNLE
         CLC   PELXQNME,RNLEQNME       CHECK IF QNAMES MATCH
         BNE   NEXTRNLE                NO => GET NEXT RNL ENTRY
         SPACE
* CHECK FOR GENERIC ENTRY IN THE RNL
         TM    RNLEFLGS,RNLEGENR       GENERIC ENTRY IN THE RNL ?
         BNO   NONGENER                NO => HANDLE NON-GENERIC ENTRY
         SPACE
* CHECK FOR AN RNAME PRESENT IN THE RNLE
         CLI   RNLERNML,ZERO           IS THE RNLE RNAME LENGTH ZERO ?
         BNE   CHKLEN                  NO => CHECK RNAME LENGTHS
         SPACE
* INDICATE RESOURCE NAME IS IN THE RNL
         LA    RTRNCODE,FOUND          MATCH HAS BEEN FOUND
         B     MODEXIT                 EXIT PROCESSING COMPLETE
         EJECT
*************************************************************************
*                                                                       *
*  8. IS THE LENGTH OF THE RNAME IN THE PEL TOO SHORT FOR THE GENERIC   *
*     ENTRY IN THE RNLE?  IF SO, GO TO STEP 14.                         *
*                                                                       *
*  9. CHECK IF THE RNAMES MATCH.  IF NOT, GO TO STEP 14.                *
*                                                                       *
* 10. RNAMES MATCH.  NAME FOUND IN THE RNL.  GO TO STEP 24.             *
*                                                                       *
*************************************************************************
         SPACE
CHKLEN   EQU   *
* CHECK IF THE LENGTH OF THE RNAME IN THE PEL IS TOO SHORT
         CLC   PELMILEN,RNLERNML       PEL RNAME LEN >= RNLE RNAME LEN?
         BL    NEXTRNLE                NO => NO MATCH, GET NEXT RNLE
         SPACE
* CHECK IF THE RNAMES MATCH
         SLR   RNAMELEN,RNAMELEN       CLEAR WORK REG
         IC    RNAMELEN,RNLERNML       GET LENGTH OF RNAME IN THE RNLE
         BCTR  RNAMELEN,ZERO           ADJUST LENGTH
         EX    RNAMELEN,COMPRNME       COMPARE THE RNAMES
         BNE   NEXTRNLE                UNEQUAL => GET NEXT RNLE
         SPACE
* INDICATE RESOURCE NAME FOUND IN THE RNL
         LA    RTRNCODE,FOUND          MATCH HAS BEEN FOUND
         B     MODEXIT                 EXIT PROCESSING COMPLETE
         EJECT
*************************************************************************
*                                                                       *
*        NON-GENERIC ENTRY PROCESSING                                   *
*                                                                       *
```

```
* 11. CHECK IF RNAME LENGTHS ARE EQUAL.  IF NOT, GO TO STEP 14.       *
*                                                                     *
* 12. COMPARE RNAMES.  IF UNEQUAL, GO TO STEP 14.                     *
*                                                                     *
* 13. RNAMES MATCH.  NAME FOUND IN THE RNL.  GO TO STEP 24.           *
*                                                                     *
*        GET NEXT RNL ENTRY                                           *
*                                                                     *
* 14. SKIP OVER THE RNL ENTRY JUST TESTED.  GO TO STEP 3.             *
*                                                                     *
***********************************************************************
         SPACE
NONGENER EQU   *                       HANDLE NON-GENERIC RNL ENTRY
* CHECK IF RNAME LENGTHS ARE EQUAL
         CLC   PELMILEN,RNLERNML       PEL RNAME LEN = RNLE RNAME LEN ?
         BNE   NEXTRNLE                NO => NO MATCH, GET NEXT RNLE
         SPACE
* COMPARE THE RNAMES
         SLR   RNAMELEN,RNAMELEN       CLEAR WORK REG
         IC    RNAMELEN,RNLERNML       GET LENGTH OF RNAME IN THE RNLE
         BCTR  RNAMELEN,ZERO           ADJUST LENGTH
         EX    RNAMELEN,COMPRNME       COMPARE THE RNAMES
         BNE   NEXTRNLE                UNEQUAL => GET NEXT RNLE
         SPACE
* INDICATE RESOURCE NAME FOUND IN THE RNL
         LA    RTRNCODE,FOUND          MATCH HAS BEEN FOUND
         B     MODEXIT                 EXIT PROCESSING COMPLETE
         SPACE
* GET THE ADDRESS OF THE NEXT RNL ENTRY
NEXTRNLE EQU   *
         LA    FLENRNLE,RNLERNME-RNLE  LENGTH OF FIXED PART OF RNLE
         SLR   WORKREG,WORKREG         CLEAR WORK REG
         IC    WORKREG,RNLERNML        GET RNAME LENGTH (VARIABLE)
         ALR   WORKREG,FLENRNLE        ADD FIXED + VARIABLE LENGTHS
         ALR   RNLEPTR,WORKREG         GET ADDRESS OF NEXT RNL ENTRY
         B     COMPRNLE                CHECK THE NEW RNL ENTRY
         EJECT
***********************************************************************
*                                                                     *
*  CONTROL ARRIVES AT ENDOFRNL ONLY WHEN THE ENTIRE RNL HAS BEEN      *
*  SCANNED AND NO ENTRY WAS FOUND THAT MATCHED THE RESOURCE NAME      *
*  IN THE PEL.                                                        *
*                                                                     *
         SPACE
ENDOFRNL EQU   *
***********************************************************************
*                                                                     *
* 24. RETURN TO CALLER.        M O D E X I T                          *
*                                                                     *
***********************************************************************
         SPACE
MODEXIT  EQU   *
         LR    REG15,RTRNCODE          GET RETURN CODE INTO REG 15
         L     14,12(,REG13)           RECOVER THE RETURN ADDRESS
         LM    0,12,20(13)             RECOVER OTHERS EXCEPT REG 115
         BR    14                      RETURN TO THE CALLER
ISGGREXE EQU   *                       END OF INSTRUCTIONS
         EJECT
***********************************************************************
*                                                                     *
* DATA USED                                                           *
*                                                                     *
***********************************************************************
         DS    0H
COMPRNME CLC   RNLERNME(ZERO),PELXRNME COMPARE RNAMES
*
COMPRNM1 CLC   RNLERNME(ZERO),UCBVOLI-UCBOB(R1) COMPARE VOLSER /*AAA
*OMPRNM1 CLC   RNLERNME(ZERO),X'1C'(R1)         COMPARE VOLSER /*AAA
*
EXCLNAME DC    CL8'RNLESEEX'           NAME IN END OF LIST ENTRY
*
HRDWNAME DC    CL8'HWRESERV'           MAJOR NAME PER ESCLUDE /*AAA
         SPACE 3
         PRINT NOGEN
         ISGPEL
         EJECT
         ISGRNLE
         IEFUCBOB DEVCLAS=DA
*
         END
```

Appendix D. SMP/E BUILDMCS Output Elements

This appendix shows each of the output elements associated with the BUILDMCS process. These elements are the:

1. Function Summary Report
2. Entry Summary Report
3. Function Sysmod

D.1 BUILDMCS Function Summary Report

```
BUILDMCS FUNCTION SUMMARY REPORT

NOTE: '*' INDICATES THE ASSOCIATED SYSMOD IS IN ERROR

FUNCTION  STATUS        FMID      ASSOCIATED SYSMODS

HMQ4120   SELECTED      HMQ4120    UN76836   UN79366   UN80416   UN80776   UN84724
                                   UN86449   UN87286   UN87827
                                   UN89088   UN90955   UN91167
JMQ412A   SELECTED      HMQ4120    UN90800
```

D.2 BUILDMCS Entry Summary Report

```
BUILDMCS ENTRY SUMMARY REPORT FOR FMID HMQ4120

ENTRY    ENTRY     ENTRY         CURRENT
TYPE     NAME      STATUS        RMID       COMMENTS

DDDEF    AASMMAC1  REQUIRED                 REQUIRED IN TARGET AND DISTRIBUTION ZONE
DDDEF    AASMMOD1  REQUIRED                 REQUIRED IN TARGET AND DISTRIBUTION ZONE
DDDEF    AASMSAM1  REQUIRED                 REQUIRED IN TARGET AND DISTRIBUTION ZONE
DDDEF    SASMMAC1  REQUIRED                 REQUIRED IN TARGET ZONE
DDDEF    SASMMOD1  REQUIRED                 REQUIRED IN TARGET ZONE
DDDEF    SASMSAM1  REQUIRED                 REQUIRED IN TARGET ZONE
LMOD     ASMADOPT  SELECTED
....
LMOD     ASMA93    SELECTED
MAC      ASMADATA  SELECTED       UN90955
....
MAC      ASMAXITP  SELECTED       UN84724
MOD      ASMACPR   SELECTED       HMQ4120
....
PROC     ASMAC     SELECTED       HMQ4120
....
PROC     ASMAPROC  SELECTED       HMQ4120
SAMP     ASMAALOC  SELECTED       HMQ4120
....
SAMP     ASMAXPRT  SELECTED       HMQ4120
SRC      ASMADOPT  SELECTED       UN76836

BUILDMCS ENTRY SUMMARY REPORT FOR FMID JMQ412A

ENTRY    ENTRY     ENTRY         CURRENT
TYPE     NAME      STATUS        RMID       COMMENTS
```

```
BOOK     ASMAGMNT  SELECTED      JMQ412A
....
BOOK     ASMTUG00  SELECTED      JMQ412A
BSIND    ASMTSH00  SELECTED      JMQ412A
DDDEF    AASMBOK2  REQUIRED                REQUIRED IN TARGET AND DISTRIBUTION ZONE
DDDEF    AASMMAC2  REQUIRED                REQUIRED IN TARGET AND DISTRIBUTION ZONE
DDDEF    AASMMOD2  REQUIRED                REQUIRED IN TARGET AND DISTRIBUTION ZONE
DDDEF    AASMSAM2  REQUIRED                REQUIRED IN TARGET AND DISTRIBUTION ZONE
DDDEF    AASMSHF2  REQUIRED                REQUIRED IN TARGET AND DISTRIBUTION ZONE
DDDEF    SASMBOK2  REQUIRED                REQUIRED IN TARGET ZONE
DDDEF    SASMMAC2  REQUIRED                REQUIRED IN TARGET ZONE
DDDEF    SASMMOD2  REQUIRED                REQUIRED IN TARGET ZONE
DDDEF    SASMSAM2  REQUIRED                REQUIRED IN TARGET ZONE
DDDEF    SASMSHF2  REQUIRED                REQUIRED IN TARGET ZONE
LMOD     ASMDASM   SELECTED
....
LMOD     ASMLANGX  SELECTED
MAC      ASMMDOPR  SELECTED      JMQ412A
....
MAC      ASMMSTKI  SELECTED      JMQ412A
MOD      ASMARG    SELECTED      JMQ412A
....
MOD      ASMMUII   SELECTED      UN90800
....
MOD      ASMTSOSD  SELECTED      UN90800
....
MOD      ASMTSOST  SELECTED      UN90800
....
MOD      ASMXWRTX  SELECTED      JMQ412A
PROC     ASMTPROC  SELECTED      JMQ412A
PUBLB    ASMTSH00  SELECTED      JMQ412A
SAMP     ASMIDF    SELECTED      JMQ412A
....
SAMP     ASMTSAMP  SELECTED      JMQ412A
```

D.3 BUILDMCS Generated Function Sysmod

```
++FUNCTION(HMQ4120) REWORK(1996235)
/*****************************************************************/
/*****************************************************************/
/*** THIS MCS FOR FUNCTION HMQ4120 WAS CREATED BY THE BUILDMCS  ***/
/*** COMMAND ON 1996235                                         ***/
/*****************************************************************/
/*****************************************************************/ .
++VER(Z038) DELETE(HMQ4100) SUP(AN69921,AN70952,AN70957,AN71301,AN71587,
  AN72000,AN72037,AN73097,AN73114,AN73139,AN73202,AN73205,AN73207,
  AN73294,AN73387,AN73919,AN74072,AN74370,AN74463,AN75023,AN75047,
  AN75745,AN76107,AN76128,AN76147,AN76379,AN76836,AN77424,AN77616,
  AN77770,AN78037,AN78183,AN78442,AN78468,AN78543,AN78630,AN78849,
  AN79308,AN79423,AN79790,AN80666,AN80697,AN81008,AN81022,AN81176,
  AN81262,AN81268,AN81432,AN81691,AN82415,AN82488,AN82659,AN82931,
  AN83223,AN83484,AN83652,AN83824,AN84217,AN84348,AN84597,HMQ4100,
  UN76836,UN79366,UN79517,UN80416,UN80776,UN84724,UN86449,UN87286,
  UN87827,UN89088,UN90526,UN90955,UN91167).
++MAC(ASMADATA) DISTLIB(AASMMAC1) FROMDS(DSN(ASMA.V1R2M0.AASMMAC1)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) SYSLIB(SASMMAC1).
++MAC(ASMADD) DISTLIB(AASMMAC1) FROMDS(DSN(ASMA.V1R2M0.AASMMAC1)
```

```
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC1).
++MAC(ASMADDV) DISTLIB(AASMMAC1) FROMDS(DSN(ASMA.V1R2M0.AASMMAC1)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC1).
++MAC(ASMAEFNP) DISTLIB(AASMMAC1) FROMDS(DSN(ASMA.V1R2M0.AASMMAC1)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC1).
++MAC(ASMAMODV) DISTLIB(AASMMAC1) FROMDS(DSN(ASMA.V1R2M0.AASMMAC1)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC1).
++MAC(ASMAOPT) DISTLIB(AASMMAC1) FROMDS(DSN(ASMA.V1R2M0.AASMMAC1)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) RMID(UN89088) SYSLIB(SASMMAC1).
++MAC(ASMAXFMB) DISTLIB(AASMMAC1) FROMDS(DSN(ASMA.V1R2M0.AASMMAC1)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC1).
++MAC(ASMAXITP) DISTLIB(AASMMAC1) FROMDS(DSN(ASMA.V1R2M0.AASMMAC1)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) RMID(UN84724) SYSLIB(SASMMAC1).
++MOD(ASMACPR) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMADOPT,ASMALTAS,ASMAMDE,
  ASMAMES,ASMAMJP,ASMAMUE,ASMAODOS,ASMAOESA,ASMAOXA,ASMAO370,ASMA90,
  ASMA93) .
++MOD(ASMADOPT) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN89088) LMOD(ASMADOPT) .
++MOD(ASMALTAS) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMALTAS) .
++MOD(ASMAOA) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMAOB) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMAOC) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN80776) LMOD(ASMA93) .
++MOD(ASMAOT) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMAOO) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMAO1) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN89088) LMOD(ASMA93) .
++MOD(ASMAO2) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMAO3) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN80416) LMOD(ASMA93) .
++MOD(ASMAO4) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN89088) LMOD(ASMA93) .
++MOD(ASMA1A) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA1B) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN80776) LMOD(ASMA93) .
++MOD(ASMA1F) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN91167) LMOD(ASMA93) .
++MOD(ASMA1G) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN84724) LMOD(ASMA93) .
++MOD(ASMA1H) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN80776) LMOD(ASMA93) .
++MOD(ASMA1I) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN86449) LMOD(ASMA93) .
++MOD(ASMA1J) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN84724) LMOD(ASMA93) .
++MOD(ASMA1K) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA1L) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN91167) LMOD(ASMA93) .
++MOD(ASMA1M) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN91167) LMOD(ASMA93) .
```

```
++MOD(ASMA1N) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN91167) LMOD(ASMA93) .
++MOD(ASMA1P) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA1Q) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN87286) LMOD(ASMA93) .
++MOD(ASMA1R) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA1S) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA1T) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN87827) LMOD(ASMA93) .
++MOD(ASMA1U) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA1V) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN80776) LMOD(ASMA93) .
++MOD(ASMA1W) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA1X) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN80776) LMOD(ASMA93) .
++MOD(ASMA1Y) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN80776) LMOD(ASMA93) .
++MOD(ASMA1Z) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN80776) LMOD(ASMA93) .
++MOD(ASMA10) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA11) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA12) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA13) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA15) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA16) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN91167) LMOD(ASMA93) .
++MOD(ASMA17) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA2@) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA2A) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN86449) LMOD(ASMA93) .
++MOD(ASMA2B) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA2D) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA2F) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA2G) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN91167) LMOD(ASMA93) .
++MOD(ASMA2H) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA2R) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA2S) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA2U) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA2V) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
```

```
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA2W) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA2Y) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA2Z) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA20) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA21) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA22) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA23) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN87827) LMOD(ASMA93) .
++MOD(ASMA24) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA25) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA26) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN91167) LMOD(ASMA93) .
++MOD(ASMA27) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN86449) LMOD(ASMA93) .
++MOD(ASMA28) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++MOD(ASMA29) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN86449) LMOD(ASMA93) .
++MOD(ASMA40) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA5DOS) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMAODOS) .
++MOD(ASMA5ESA) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMAOESA) .
++MOD(ASMA5UNI) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA5XA) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMAOXA) .
++MOD(ASMA5370) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMAO370) .
++MOD(ASMA7DE) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN86449) LMOD(ASMAMDE) .
++MOD(ASMA7EN) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN86449) LMOD(ASMA93) .
++MOD(ASMA7ES) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN86449) LMOD(ASMAMES) .
++MOD(ASMA7JP) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN80416) LMOD(ASMAMJP) .
++MOD(ASMA7UE) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN86449) LMOD(ASMAMUE) .
++MOD(ASMA70) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA71) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA8T) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
++MOD(ASMA80) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN89088) LMOD(ASMA93) .
++MOD(ASMA9D) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) LMOD(ASMA93) .
```

```
++MOD(ASMA9Z) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN91167) LMOD(ASMA93) .
++MOD(ASMA90) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA90) .
++MOD(ASMA91) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN80776) LMOD(ASMA93) .
++MOD(ASMA92) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN79366) LMOD(ASMA93) .
++MOD(ASMA93) DISTLIB(AASMMOD1) FROMDS(DSN(ASMA.V1R2M0.AASMMOD1)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) RMID(UN90955) LMOD(ASMA93) .
++PROC(ASMAC) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1) ALIAS(HLASMC).
++PROC(ASMACG) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1) ALIAS(HLASMCG).
++PROC(ASMACL) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1) ALIAS(HLASMCL).
++PROC(ASMACLG) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1) ALIAS(HLASMCLG).
++PROC(ASMAPROC) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAACPT) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAALOC) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAAPLY) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAASM) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMACOPY) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMADDDF) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMADEL) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMADELO) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAEDIT) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAFGMT) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAIEV) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAIVP) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAOPTS) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAPTST) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMARECV) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMASAMP) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMASMPE) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMASMPN) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMASTD) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAXADT) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
```

```
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAXFLU) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAXFMT) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAXFSK) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAXINV) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SAMP(ASMAXPRT) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM1).
++SRC(ASMADOPT) DISTLIB(AASMSAM1) FROMDS(DSN(ASMA.V1R2M0.AASMSAM1)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) RMID(UN76836) SYSLIB(SASMSAM1).
++JCLIN CALLLIBS.
//LINK0001 EXEC PGM=IEWBLINK,PARM=('OPTIONS(GENOPTS)')
//GENOPTS  DD *
                RENT,REUS,REFR,AMODE=ANY,RMODE=24,NCAL
//SYSLMOD  DD DSN=SASMMOD1
//SYSLIN   DD *
 ORDER ASMA90                                                          00000100
 ENTRY ASMA90                                                          00000200
  INCLUDE AASMMOD1(ASMACPR)           FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA90)            FMID=HMQ4120
  NAME ASMA90
//LINK0002 EXEC PGM=IEWBLINK,PARM=('OPTIONS(GENOPTS)')
//GENOPTS  DD *
                RENT,REUS,REFR,AMODE=31,RMODE=ANY,NCAL
//SYSLMOD  DD DSN=SASMMOD1
//SYSLIN   DD *
 ENTRY ASMADOPT                                                        00000100
  INCLUDE AASMMOD1(ASMACPR)           FMID=HMQ4120
  INCLUDE AASMMOD1(ASMADOPT)          FMID=HMQ4120
  NAME ASMADOPT
 ENTRY ASMALTAS                                                        00000100
  INCLUDE AASMMOD1(ASMACPR)           FMID=HMQ4120
  INCLUDE AASMMOD1(ASMALTAS)          FMID=HMQ4120
  NAME ASMALTAS
 ENTRY ASMAMDE                                                         00000100
  INCLUDE AASMMOD1(ASMACPR)           FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA7DE)           FMID=HMQ4120
  NAME ASMAMDE
 ENTRY ASMAMES                                                         00000100
  INCLUDE AASMMOD1(ASMACPR)           FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA7ES)           FMID=HMQ4120
  NAME ASMAMES
 ENTRY ASMAMJP                                                         00000100
  INCLUDE AASMMOD1(ASMACPR)           FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA7JP)           FMID=HMQ4120
  NAME ASMAMJP
 ENTRY ASMAMUE                                                         00000100
  INCLUDE AASMMOD1(ASMACPR)           FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA7UE)           FMID=HMQ4120
  NAME ASMAMUE
 ENTRY ASMAODOS                                                        00000100
  INCLUDE AASMMOD1(ASMACPR)           FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA5DOS)          FMID=HMQ4120
  NAME ASMAODOS
 ENTRY ASMAOESA                                                        00000100
  INCLUDE AASMMOD1(ASMACPR)           FMID=HMQ4120
```

```
 INCLUDE AASMMOD1(ASMA5ESA)          FMID=HMQ4120
 NAME ASMAOESA
ENTRY ASMAOXA                                                    00000100
 INCLUDE AASMMOD1(ASMACPR)           FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA5XA)           FMID=HMQ4120
 NAME ASMAOXA
ENTRY ASMAO370                                                   00000100
 INCLUDE AASMMOD1(ASMACPR)           FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA5370)          FMID=HMQ4120
 NAME ASMAO370
ORDER ASMA93                                                     00000100
ORDER ASMA91                                                     00000200
ORDER ASMA92                                                     00000300
ORDER ASMA70                                                     00000400
ENTRY ASMA93                                                     00000500
 INCLUDE AASMMOD1(ASMACPR)           FMID=HMQ4120
 INCLUDE AASMMOD1(ASMAOA)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMAOB)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMAOC)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMAOT)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMAOO)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMAO1)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMAO2)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMAO3)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMAO4)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1A)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1B)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1F)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1G)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1H)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1I)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1J)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1K)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1L)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1M)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1N)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1P)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1Q)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1R)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1S)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1T)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1U)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1V)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1W)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1X)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1Y)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA1Z)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA10)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA11)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA12)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA13)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA15)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA16)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA17)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA2@)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA2A)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA2B)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA2D)            FMID=HMQ4120
 INCLUDE AASMMOD1(ASMA2F)            FMID=HMQ4120
```

```
  INCLUDE AASMMOD1(ASMA2G)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA2H)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA2R)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA2S)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA2U)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA2V)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA2W)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA2Y)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA2Z)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA20)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA21)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA22)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA23)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA24)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA25)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA26)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA27)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA28)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA29)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA40)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA5UNI)          FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA7EN)           FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA70)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA71)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA8T)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA80)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA9D)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA9Z)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA91)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA92)            FMID=HMQ4120
  INCLUDE AASMMOD1(ASMA93)            FMID=HMQ4120
  NAME ASMA93
++FUNCTION(JMQ412A) REWORK(1996235)
/*****************************************************************/
/*****************************************************************/
/*** THIS MCS FOR FUNCTION JMQ412A WAS CREATED BY THE BUILDMCS  ***/
/*** COMMAND ON 1996235                                         ***/
/*****************************************************************/
/*****************************************************************/ .
++VER(Z038) FMID(HMQ4120) SUP(AN84253,UN90800).
++BOOK(ASMAGMNT) DISTLIB(AASMBOK2) FROMDS(DSN(ASMT.V1R2M0.AASMBOK2)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMBOK2).
++BOOK(ASMLP002) DISTLIB(AASMBOK2) FROMDS(DSN(ASMT.V1R2M0.AASMBOK2)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMBOK2).
++BOOK(ASMTIC00) DISTLIB(AASMBOK2) FROMDS(DSN(ASMT.V1R2M0.AASMBOK2)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMBOK2).
++BOOK(ASMTIS00) DISTLIB(AASMBOK2) FROMDS(DSN(ASMT.V1R2M0.AASMBOK2)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMBOK2).
++BOOK(ASMTIU00) DISTLIB(AASMBOK2) FROMDS(DSN(ASMT.V1R2M0.AASMBOK2)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMBOK2).
++BOOK(ASMTUG00) DISTLIB(AASMBOK2) FROMDS(DSN(ASMT.V1R2M0.AASMBOK2)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMBOK2).
++BSIND(ASMTSH00) DISTLIB(AASMBOK2) FROMDS(DSN(ASMT.V1R2M0.AASMBOK2)
  NUMBER(1) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMBOK2).
++MAC(ASMMDOPR) DISTLIB(AASMMAC2) FROMDS(DSN(ASMT.V1R2M0.AASMMAC2)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC2).
++MAC(ASMMGBLV) DISTLIB(AASMMAC2) FROMDS(DSN(ASMT.V1R2M0.AASMMAC2)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC2).
++MAC(ASMMGETC) DISTLIB(AASMMAC2) FROMDS(DSN(ASMT.V1R2M0.AASMMAC2)
```

```
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC2).
++MAC(ASMMIFPR) DISTLIB(AASMMAC2) FROMDS(DSN(ASMT.V1R2M0.AASMMAC2)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC2).
++MAC(ASMMPOPI) DISTLIB(AASMMAC2) FROMDS(DSN(ASMT.V1R2M0.AASMMAC2)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC2).
++MAC(ASMMPOPN) DISTLIB(AASMMAC2) FROMDS(DSN(ASMT.V1R2M0.AASMMAC2)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC2).
++MAC(ASMMPSHI) DISTLIB(AASMMAC2) FROMDS(DSN(ASMT.V1R2M0.AASMMAC2)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC2).
++MAC(ASMMPSHL) DISTLIB(AASMMAC2) FROMDS(DSN(ASMT.V1R2M0.AASMMAC2)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC2).
++MAC(ASMMPSHN) DISTLIB(AASMMAC2) FROMDS(DSN(ASMT.V1R2M0.AASMMAC2)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC2).
++MAC(ASMMSP) DISTLIB(AASMMAC2) FROMDS(DSN(ASMT.V1R2M0.AASMMAC2)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC2).
++MAC(ASMMSTKI) DISTLIB(AASMMAC2) FROMDS(DSN(ASMT.V1R2M0.AASMMAC2)
  NUMBER(2) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMMAC2).
++MOD(ASMARG) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMDASM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMDASM) .
++MOD(ASMDTIME) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMD01) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMDASM) .
++MOD(ASMD02) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMDASM) .
++MOD(ASMGTERM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMLCLAC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLCLCT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLCLVF) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLCXSC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLDACP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLDATA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLDFRD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLDFSE) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLDLMD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLDLPG) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLEXIT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLGSAC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLGSDA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLGSDR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLGSFI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
```

```
++MOD(ASMLGSRD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLGSWR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLGTSA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLGTSC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLGTSP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLGTSR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLGTST) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLINIT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLLGOP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLLIT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLLITE) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLLNRS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLLNRT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLMDAD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLMDDR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLMEIA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLMERD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLMEUA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLMEWR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLMSG) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLMSGE) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLMTSO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLPSBT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLPSFR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLPSLS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLPSTP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLPTAC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLPTAE) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLPTDE) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLPTFE) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
```

```
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLPTGL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLPTGP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLPTGT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLRGRD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLRGWR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLRXAB) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLRXGS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLRXSC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLRXVD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLRXVL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLRXVV) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSCAB) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSCCM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSCTK) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHCL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHEN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHGN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHGS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHLN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHLS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHMM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHMS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHNM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHOP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHPA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHPC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHPI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHPL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHPP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
```

```
++MOD(ASMLSHPS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHPT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHPV) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHSM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHSV) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSHTZ) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSIMA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSIMD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSISA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSISS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSMTS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSQTS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSRCS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSRSA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSRSC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSRSI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSRSR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSRSS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSRST) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSRVN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSRVP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSRVS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSSFI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSSFS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSSLO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSSRS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLSVIN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPBG) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPCL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPCT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
```

```
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPEN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPEX) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPFL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPIN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPMW) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPOP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPRA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPRW) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPSC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTPSM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLTZTS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVIAT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVIFM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVILC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVISB) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVISS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVISZ) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVLAT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVLSS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVLWR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVRCI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVRDS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVREN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVRFY) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVRID) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVRLC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVRLK) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVRLV) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVROI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
```

```
++MOD(ASMLVRPR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVRSB) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVRSE) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVRWR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLVRXX) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWABL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWACI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWACL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWAOR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWATI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWNAL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWNAR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWNBL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWNCI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWNFR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWNMR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWNOR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWNSC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWNSE) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLWNUD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXAEN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXALI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXAVI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXBAO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXBET) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXBEX) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXBFP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXBID) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXBLD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXBPS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
```

```
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXBSS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXCEN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXCHK) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDAO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDAR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDBO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDCA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDCD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDEN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDFN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDID) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDIX) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDLC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDLO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDLP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDOP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDPA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDPO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDPP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDSS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDUO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXDVL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXEAT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXEDM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXEID) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXELX) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXEOP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXEOZ) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXFEN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
```

```
++MOD(ASMLXFIN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXIDC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXIDS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXIDX) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXILD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXILL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXILS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXIPR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXIXP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXNEN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXPEF) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXPEI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXPEN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXPLI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXPRT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXPVI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXRBL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXRBO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXRCL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXRDL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXRFN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXRID) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXRLO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXRLP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXRUO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXRXL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXTML) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXTMS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXTMX) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXVEN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
```

```
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMLXX) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMAIN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMCAP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMCDB) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMCMN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMCMX) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMCOM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMCON) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMCTR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMSG) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMSTRT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMSUB1) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMSUB2) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMSUB3) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMSUB4) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMSUB5) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMSUB6) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMTOM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMTSEX) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMTSIO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMTSMA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMTSSV) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMTSS1) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMTSU1) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMTSU2) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMTTBL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUAO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUCI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUDI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
```

```
++MOD(ASMMUGS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUID) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUII) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) RMID(UN90800) LMOD(ASMIDF) .
++MOD(ASMMUIO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUI1) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUI2) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUI3) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUI4) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUI5) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUI6) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMULL) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMULS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMULT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUPR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUSV) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUTL2) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUTL3) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUTL5) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUTL6) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUTL7) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMUTR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMMWACH) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMPTERM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMRTENV) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMTSOFI) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMTSOIO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMTSOMM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMTSORT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMTSORX) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMTSOSD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
```

```
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) RMID(UN90800) LMOD(ASMIDF,ASMLANGX)
  .
++MOD(ASMTSOST) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) RMID(UN90800) LMOD(ASMIDF,ASMLANGX)
  .
++MOD(ASMWAFCE) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMWCDST) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMWCPR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMDASM,ASMIDF,ASMLANGX) .
++MOD(ASMWDBIO) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMWDBMA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMWDBMB) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMWDBPK) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMWDBPM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF,ASMLANGX) .
++MOD(ASMWDBSF) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMWLSCR) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMWSERV) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMWSYST) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMIDF) .
++MOD(ASMXADAT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXDACP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXDATA) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHASM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHDSM) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHESD) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHFLT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHICT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHPP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHSRC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHUSE) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHVAS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHVCW) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHVDC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHVDS) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
```

```
++MOD(ASMXHVEQ) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXHVLB) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXLIT) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXLITE) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXMAIN) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXMSG) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXMSGE) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXPP) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXSUBC) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXSUB1) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXSUB2) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXSUB3) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXSUB4) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXSUB5) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXSUB6) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXSUB7) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXSUB8) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXSUB9) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++MOD(ASMXWRTX) DISTLIB(AASMMOD2) FROMDS(DSN(ASMT.V1R2M0.AASMMOD2)
  NUMBER(3) VOL(OS3DL1) UNIT(3390)) LMOD(ASMLANGX) .
++PROC(ASMTPROC) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++PUBLB(ASMTSH00) DISTLIB(AASMSHF2) FROMDS(DSN(ASMT.V1R2M0.AASMSHF2)
  NUMBER(5) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSHF2).
++SAMP(ASMIDF) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++SAMP(ASMTACPT) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++SAMP(ASMTALOC) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++SAMP(ASMTAPLY) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++SAMP(ASMTCOPY) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++SAMP(ASMTDDDF) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++SAMP(ASMTDELO) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++SAMP(ASMTEDIT) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++SAMP(ASMTIVP) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
```

```
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++SAMP(ASMTMOVE) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++SAMP(ASMTRECV) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++SAMP(ASMTSAM) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++SAMP(ASMTSAMP) DISTLIB(AASMSAM2) FROMDS(DSN(ASMT.V1R2M0.AASMSAM2)
  NUMBER(4) VOL(OS3DL1) UNIT(3390)) SYSLIB(SASMSAM2).
++JCLIN CALLLIBS.
//LINK0001 EXEC PGM=IEWBLINK,PARM=('OPTIONS(GENOPTS)')
//GENOPTS  DD *
                REUS,AMODE=24,RMODE=24,NCAL
//SYSLMOD  DD DSN=SASMMOD2
//SYSLIN   DD *
 ORDER ASMDASM                                                          00000100
 ENTRY ASMDASM                                                          00000200
  INCLUDE AASMMOD2(ASMDASM)           FMID=JMQ412A
  INCLUDE AASMMOD2(ASMD01)            FMID=JMQ412A
  INCLUDE AASMMOD2(ASMD02)            FMID=JMQ412A
  INCLUDE AASMMOD2(ASMWCPR)           FMID=JMQ412A
  NAME ASMDASM
 ORDER ASMMSTRT                                                         00000100
 ORDER ASMMCOM                                                          00000200
 ORDER ASMMTTBL                                                         00000300
 ORDER ASMMAIN                                                          00000400
 ORDER ASMMTSMA                                                         00000500
 ORDER ASMLMAIN                                                         00000600
 ORDER ASMLDATA                                                         00000700
 ORDER ASMLMSG                                                          00000800
 ORDER ASMLLIT                                                          00000900
 ORDER ASMGTRML                                                         00001000
 ORDER ASMPTRML                                                         00001100
 ENTRY ASMMAIN                                                          00001200
  INCLUDE AASMMOD2(ASMARG)            FMID=JMQ412A
  INCLUDE AASMMOD2(ASMDTIME)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMGTERM)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLCLAC)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLCLCT)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLCLVF)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLCXSC)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLDACP)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLDATA)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLDFRD)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLDFSE)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLDLMD)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLDLPG)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLEXIT)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLGSAC)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLGSDA)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLGSDR)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLGSFI)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLGSRD)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLGSWR)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLGTSA)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLGTSC)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLGTSP)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLGTSR)          FMID=JMQ412A
  INCLUDE AASMMOD2(ASMLGTST)          FMID=JMQ412A
```

```
INCLUDE AASMMOD2(ASMLINIT)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLLGOP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLLIT)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMLLITE)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLLNRS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLLNRT)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLMDAD)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLMDDR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLMEIA)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLMERD)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLMEUA)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLMEWR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLMSG)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMLMSGE)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLMTSO)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLPSBT)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLPSFR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLPSLS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLPSTP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLPTAC)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLPTAE)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLPTDE)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLPTFE)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLPTGL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLPTGP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLPTGT)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLRGRD)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLRGWR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLRXAB)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLRXGS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLRXSC)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLRXVD)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLRXVL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLRXVV)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSCAB)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSCCM)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSCTK)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHCL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHEN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHGN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHGS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHLN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHLS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHMM)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHMS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHNM)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHOP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHPA)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHPC)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHPI)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHPL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHPP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHPS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHPT)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHPV)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHSM)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHSV)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSHTZ)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSIMA)          FMID=JMQ412A
```

```
INCLUDE AASMMOD2(ASMLSIMD)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSISA)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSISS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSMTS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSQTS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSRCS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSRSA)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSRSC)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSRSI)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSRSR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSRSS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSRST)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSRVN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSRVP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSRVS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSSFI)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSSFS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSSLO)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSSRS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLSVIN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPBG)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPCL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPCT)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPEN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPEX)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPFL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPIN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPMW)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPOP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPRA)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPRW)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPSC)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTPSM)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLTZTS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVIAT)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVIFM)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVILC)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVISB)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVISS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVISZ)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVLAT)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVLSS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVLWR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVRCI)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVRDS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVREN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVRFY)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVRID)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVRLC)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVRLK)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVRLV)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVROI)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVRPR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVRSB)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVRSE)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVRWR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLVRXX)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWABL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWACI)          FMID=JMQ412A
```

```
INCLUDE AASMMOD2(ASMLWACL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWAOR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWATI)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWNAL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWNAR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWNBL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWNCI)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWNFR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWNMR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWNOR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWNSC)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWNSE)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLWNUD)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXAEN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXALI)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXAVI)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXBAO)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXBET)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXBEX)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXBFP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXBID)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXBLD)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXBPS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXBSS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXCEN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXCHK)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDAO)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDAR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDBO)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDCA)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDCD)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDEN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDFN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDID)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDIX)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDLC)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDLO)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDLP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDOP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDPA)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDPO)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDPP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDSS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDUO)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXDVL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXEAT)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXEDM)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXEID)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXELX)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXEOP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXEOZ)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXFEN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXFIN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXIDC)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXIDS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXIDX)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXILD)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXILL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXILS)          FMID=JMQ412A
```

```
INCLUDE AASMMOD2(ASMLXIPR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXIXP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXNEN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXPEF)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXPEI)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXPEN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXPLI)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXPRT)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXPVI)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXRBL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXRBO)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXRCL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXRDL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXRFN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXRID)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXRLO)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXRLP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXRUO)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXRXL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXTML)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXTMS)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXTMX)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXVEN)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMLXX)            FMID=JMQ412A
INCLUDE AASMMOD2(ASMMAIN)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMCAP)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMCDB)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMCMN)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMCMX)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMCOM)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMCON)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMCTR)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMSG)            FMID=JMQ412A
INCLUDE AASMMOD2(ASMMSTRT)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMSUB1)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMSUB2)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMSUB3)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMSUB4)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMSUB5)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMSUB6)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMTOM)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMTSEX)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMTSIO)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMTSMA)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMTSSV)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMTSS1)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMTSU1)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMTSU2)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMTTBL)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMMUAO)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMUCI)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMUDI)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMUGS)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMUID)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMUII)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMUIO)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMUI1)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMUI2)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMMUI3)           FMID=JMQ412A
```

```
 INCLUDE AASMMOD2(ASMMUI4)          FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMUI5)          FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMUI6)          FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMULL)          FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMULS)          FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMULT)          FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMUPR)          FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMUSV)          FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMUTL2)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMUTL3)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMUTL5)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMUTL6)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMUTL7)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMUTR)          FMID=JMQ412A
 INCLUDE AASMMOD2(ASMMWACH)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMPTERM)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMRTENV)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSOFI)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSOIO)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSOMM)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSORX)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSOSD)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSOST)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMWAFCE)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMWCDST)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMWCPR)          FMID=JMQ412A
 INCLUDE AASMMOD2(ASMWDBIO)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMWDBMA)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMWDBMB)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMWDBPK)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMWDBPM)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMWLSCR)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMWSERV)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMWSYST)         FMID=JMQ412A
 NAME ASMIDF
ORDER ASMXDACP                                                  00000100
ORDER ASMXMAIN                                                  00000200
ORDER ASMXDATA                                                  00000300
ORDER ASMXMSG                                                   00000400
ORDER ASMXMSGE                                                  00000500
ORDER ASMXLIT                                                   00000600
ORDER ASMXLITE                                                  00000700
ORDER ASMPTRML                                                  00000800
ORDER ASMGTRML                                                  00000900
ORDER ASMXPP                                                    00001000
ORDER ASMXWRTX                                                  00001100
ENTRY ASMINITP                                                  00001200
 INCLUDE AASMMOD2(ASMARG)           FMID=JMQ412A
 INCLUDE AASMMOD2(ASMDTIME)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMGTERM)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMPTERM)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMRTENV)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSOFI)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSOIO)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSOMM)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSORT)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSORX)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSOSD)         FMID=JMQ412A
 INCLUDE AASMMOD2(ASMTSOST)         FMID=JMQ412A
```

```
INCLUDE AASMMOD2(ASMWCPR)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMWDBIO)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMWDBMA)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMWDBMB)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMWDBPK)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMWDBPM)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMWDBSF)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXADAT)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXDACP)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXDATA)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHASM)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHDSM)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHESD)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHFLT)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHICT)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHPP)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHSRC)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHUSE)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHVAS)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHVCW)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHVDC)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHVDS)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHVEQ)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXHVLB)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXLIT)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMXLITE)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXMAIN)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXMSG)          FMID=JMQ412A
INCLUDE AASMMOD2(ASMXMSGE)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXPP)           FMID=JMQ412A
INCLUDE AASMMOD2(ASMXSUBC)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXSUB1)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXSUB2)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXSUB3)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXSUB4)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXSUB5)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXSUB6)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXSUB7)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXSUB8)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXSUB9)         FMID=JMQ412A
INCLUDE AASMMOD2(ASMXWRTX)         FMID=JMQ412A
NAME ASMLANGX
```

Appendix E. Special Notices

This publication is intended to help system programmers understand the new functions of OS/390 Release 2 and to use the ServerPac for installation. The information in this publication is not intended as the specification of any programming interfaces that are provided by OS/390 Release 2 and ServerPac. See the PUBLICATIONS section of the IBM Programming Announcement for OS/390 Release 2 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AnyNet	AS/400
BookManager	BookMaster
C/MVS	C/370
CBIPO	CBPDO
CICS	CICS OS/2
CICS/ESA	DB2
DFSMS	DFSMS/MVS
DFSMSdfp	DFSMSdss

DFSMShsm
DFSMSrmm
DFSORT
ES/9000
Extended Services
FFST/MVS
GDDM
Hardware Configuration Definition
Hiperbatch
IBM
IMS
IMS/ESA
MVS
MVS/DFP
MVS/ESA
MVS/SP
NetView
OPC
OpenEdition
OS/2
OS/390
Parallel Sysplex
Print Services Facility
Processor Resource/Systems Manager
PROFS
PSF
RACF
RAMAC
RISC System/6000
RMF
S/390
ServicePac
Sysplex Timer
System/390
SystemView
VisualLift
VM/ESA
VTAM
YEAR 2000
3090

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo
are trademarks or registered trademarks of Microsoft Corporation.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Other trademarks are trademarks of their respective companies.

Appendix F. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

F.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How To Get ITSO Redbooks” on page 255.

Short Title	Title	Order Number
Version 5 Implementation Guide	*MVS/ESA Version 5 Implementation Guide*	SG24-4584
MVS 5.1 Presentation Guide	*MVS/ESA SP 5.1.0 Technical Presentation Guide*	GG24-4137
Parallel Sysplex Perf.	*S/390 Parallel Sysplex Performance*	SG24-4356
HCD and Dynamic I/O Reconfiguration Primer	*MVS/ESA HCD and Dynamic I/O Reconfiguration Primer*	SG24-4037
Sysplex Migration Guide	*MVS/ESA Version 5 Sysplex Migration Guide*	SG24-4581

F.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

F.3 Other Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook. A publication whose order number begins with the prefix **LY** is available to IBM-licensed customers only.

These publications are also relevant as further information sources:

- ServerPac documents

Short Title	Title	Order Number
ServerPac Guide and Worksheet	*ServerPac Guide and Worksheet*	SC28-1244

Short Title	Title	Order Number
SMP/E R8.1 CBIPO Dialog Users' Guide	*SMP/E R8.1 CBIPO Dialog Users' Guide*	SC23-0538
MVS Customized Offerings	*MVS Custom-Built Offerings Planning*	SC23-0352

- OS/390 BDT

Short Title	Title	Order Number
OS/390 BDT Overview	*OS/390 V1R2.0 BDT Overview*	GC28-1741
OS/390 BDT Installation	*OS/390 V1R2.0 BDT Installation*	SC28-1742
OS/390 BDT File-to-File Transaction Guide	*OS/390 V1R2.0 BDT File-to-File Transaction Guide*	SC28-1743
OS/390 BDT Commands	*OS/390 V1R2.0 BDT Commands*	SC28-1744
OS/390 BDT Messages and Codes	*OS/390 V1R2.0 BDT Messages and Codes*	SC28-1745

- OS/390 OpenEdition

Short Title	Title	Order Number
OS/390 OpenEdition MVS Introduction	*OS/390 OpenEdition MVS Introduction*	GC28-1889
OS/390 OpenEdition MVS POSIX.1 Conformance Document	*OS/390 OpenEdition MVS POSIX.1 Conformance Document*	GC28-1895
OS/390 OpenEdition MVS POSIX.2 Conformance Document	*OS/390 OpenEdition MVS POSIX.2 Conformance Document*	GC28-1896
OS/390 OpenEdition MVS User's Guide	*OS/390 OpenEdition MVS User's Guide*	SC28-1891
OS/390 OpenEdition MVS Command Reference	*OS/390 OpenEdition MVS Command Reference*	SC28-1892
OS/390 OpenEdition MVS Programming Tools	*OS/390 OpenEdition MVS Programming Tools*	SC28-1904
OS/390 OpenEdition MVS Messages and Codes	*OS/390 OpenEdition MVS Messages and Codes*	SC28-1908
OS/390 OpenEdition MVS Programming: Assembler Callable Services Reference	*OS/390 OpenEdition MVS Programming: Assembler Callable Services Reference*	SC28-1899
OS/390 OpenEdition MVS Planning	*OS/390 OpenEdition MVS Planning*	SC28-1890
OS/390 OpenEdition MVS File System Interface Reference	*OS/390 OpenEdition MVS File System Interface Reference*	SC28-1909
OS/390 OpenEdition MVS Using REXX and OpenEdition MVS	*OS/390 OpenEdition MVS Using REXX and OpenEdition MVS*	SC28-1905
C/MVS Library Reference: OpenEdition MVS Curses	*C/MVS Library Reference: OpenEdition MVS Curses*	SC23-3876
OS/390 OpenEdition MVS XPG4 Conformance Document	*OS/390 OpenEdition MVS XPG4 Conformance Document*	GC28-1897
C/C++ for MVS Library Reference	*C/C++ for MVS Library Reference*	SC23-3881
OS/390 OpenEdition MVS Communications Server Guide	*OS/390 OpenEdition MVS Communications Server Guide*	SC28-1906

- I/O Configuration Management

Short Title	Title	Order Number
OS/390 HCD Planning	*OS/390 Hardware Configuration Definition Planning*	GC28-1750
OS/390 HCD User's Guide	*OS/390 HCD User's Guide*	SC28-1848
HCD Messages	*OS/390 HCD Messages*	GC28-1849
HCD Scenarios	*OS/390 HCD Scenarios*	SC28-1850
HCD Reference Summary	*OS/390 HCD Reference Summary*	SX22-0043

- RMF

Short Title	Title	Order Number
RMF Messages and Codes	*OS/390 RMF Messages and Codes*	GC28-1948
RMF Performance Management Guide	*OS/390 RMF Performance Management Guide*	SC28-1951
RMF User's Guide	*OS/390 RMF User's Guide*	SC28-1949
RMF Diagnosis Guide	*OS/390 RMF Diagnosis Guide*	LY28-1132
RMF Report Analysis	*OS/390 RMF Report Analysis*	SC28-1950
RMF Reference Summary	*OS/390 RMF Reference Summary*	SX22-0044
RMF Programmer's Guide	*OS/390 RMF Programmer's Guide*	SC28-1952

- OpenEdition DCE

Short Title	Title	Order Number
OE DCE Application Development Guide	*OS/390 OpenEdition DCE Application Development Guide: Introduction and Style*	SC28-1587
OpenEdition DCE Application Development Guide: Core Components	*OS/390 OpenEdition DCE Application Development Guide: Core Components*	SC28-1588
OpenEdition DCE Application Development Guide: Directory Services	*OS/390 OpenEdition DCE Application Development Guide: Directory Services*	SC28-1589
OpenEdition DCE Command Reference	*OS/390 OpenEdition DCE Command Reference*	SC28-1585
OpenEdition DCE Administration Guide	*OS/390 OpenEdition DCE Administration Guide*	SC28-1584
OpenEdition DCE Application Development Reference	*OS/390 OpenEdition DCE Application Development Reference*	SC28-1590
Understanding DCE Concepts	*Understanding Distributed Computing Environment Concepts*	GC09-1478
OpenEdition DCE Configuring and Getting Started	*OS/390 OpenEdition DCE Configuring and Getting Started*	SC28-1583
OpenEdition DCE Introduction	*OS/390 OpenEdition DCE Introduction*	GC28-1581
OpenEdition DCE Messages and Codes	*OS/390 OpenEdition DCE Messages and Codes*	SC28-1591
OpenEdition DCE User's Guide	*OS/390 OpenEdition DCE User's Guide*	SC28-1586
OpenEdition DCE Security Server Overview	*OS/390 OpenEdition DCE Security Server Overview*	GC28-1938
OS/390 DCE AS Supplementary Information	*OS/390 DCE Application Support Supplementary Information*	LY28-1100

Short Title	Title	Order Number
OpenEdition DCE Application Support Programming Guide	*OpenEdition DCE Application Support for MVS/ESA Programming Guide*	SC09-1530
OpenEdition DCE Application Support Licensed Program Specifications	*OpenEdition DCE Application Support for MVS/ESA Licensed Program Specifications*	GC09-1701
OpenEdition DCE Application Support Configuration and Administration Guide	*OpenEdition DCE Application Support for MVS/ESA Configuration and Administration Guide*	SC09-1659

- Multi-System Configuration Management

Short Title	Title	Order Number
OS/390 Parallel Sysplex Overview	*OS/390 Parallel Sysplex Overview*	GC28-1860
OS/390 Parallel Sysplex Systems Management	*OS/390 Parallel Sysplex Systems Management*	GC28-1861
OS/390 Parallel Sysplex Hardware and Software Migration	*OS/390 Parallel Sysplex Hardware and Software Migration*	GC28-1862
OS/390 Parallel Sysplex Application Migration	*OS/390 Parallel Sysplex Application Migration*	GC28-1863
OS/390 V1R2.0 MVS Setting Up a Sysplex	*OS/390 MVS Setting Up a Sysplex*	GC28-1779
OS/390 V1R2.0 MVS Sysplex Services Guide	*OS/390 MVS Programming: Sysplex Services Guide*	GC28-1771
OS/390 V1R2.0 MVS Sysplex Services Reference	*OS/390 MVS Programming: Sysplex Services Reference*	GC28-1772

- OS/390 Operating System

Short Title	Title	Order Number
OS/390 V1R2.0 MVS Auth Assembler Services Reference ALE-DYN	*OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 1, ALE-DYN*	GC28-1764
OS/390 V1R2.0 MVS Auth Assembler Services Reference ENF-ITT	*OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 2, ENF-ITT*	GC28-1765
OS/390 V1R2.0 MVS Auth Assembler Services Reference LLA-SDU	*OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 3, LLA-SDU*	GC28-1766
OS/390 V1R2.0 MVS Auth Assembler Services Reference SET-WTO	*OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 4, SET-WTO*	GC28-1767
OS/390 V1R2.0 MVS Extended Addressability Guide	*OS/390 MVS Programming: Extended Addressability Guide*	GC28-1769
OS/390 V1R2.0 MVS Assembler Services Guide	*OS/390 MVS Programming: Assembler Services Guide*	GC28-1762
OS/390 V1R2.0 MVS Assembler Services Reference	*OS/390 MVS Programming: Assembler Services Reference*	GC28-1910
OS/390 V1R2.0 MVS Auth Assembler Services Guide	*OS/390 MVS Programming: Authorized Assembler Services Guide*	GC28-1763

Short Title	Title	Order Number
Introducing OS/390	*OS/390 Introduction and Release Guide*	GC28-1725
OS/390 V1R2.0 MVS JCL User's Guide	*OS/390 MVS JCL User's Guide*	GC28-1758
OS/390 V1R2.0 MVS JCL Reference	*OS/390 MVS JCL Reference*	GC28-1757
MVS OLTEP	*MVS OLTEP*	GC28-1448
MVS Remote Workstation Generation	*MVS Remote Workstation Generation*	GC28-1433
OS/390 V1R2.0 MVS Callable Services for HLL	*OS/390 MVS Programming: Callable Services for High-Level Languages*	GC28-1768
OS/390 V1R2.0 MVS Writing TPs for APPC/MVS	*OS/390 MVS: Writing Transaction Programs for APPC/MVS*	GC28-1775
OS/390 V1R2.0 MVS Planning: APPC/MVS Management	*OS/390 MVS Planning: APPC/MVS Management*	GC28-1807
OS/390 V1R2.0 MVS IPCS Commands	*OS/390 MVS Interactive Problem Control System (IPCS) Commands*	GC28-1754
OS/390 V1R2.0 MVS IPCS User's Guide	*OS/390 MVS Interactive Problem Control System (IPCS) User's Guide*	GC28-1756
OS/390 V1R2.0 MVS IPCS Customization	*OS/390 MVS Interactive Problem Control System (IPCS) Customization*	GC28-1755
OS/390 V1R2.0 MVS Initialization and Tuning Guide	*OS/390 MVS Initialization and Tuning Guide*	SC28-1751
OS/390 V1R2.0 MVS Initialization and Tuning Reference	*OS/390 MVS Initialization and Tuning Reference*	SC28-1752
OS/390 V1R2.0 MVS Installation Exits	*OS/390 MVS Installation Exits*	SC28-1753
OS/390 V1R2.0 MVS Using the Functional Subsystem Interface	*OS/390 MVS Using the Functional Subsystem Interface*	SC28-1911
OS/390 V1R2.0 MVS Conversion Notebook	*OS/390 MVS Conversion Notebook*	GC28-1747
OS/390 V1R2.0 MVS Using the Subsystem Interface	*OS/390 MVS Using the Subsystem Interface*	SC28-1789
OS/390 V1R2.0 MVS System Commands Summary	*OS/390 MVS System Commands Summary*	GX22-0040
MVS Planning: Security	*MVS Planning: Security*	GC28-1439
OS/390 Up and Running!	*OS/390 Up and Running!*	GC28-1726
OS/390 V1R2.0 MVS System Commands	*OS/390 MVS System Commands*	GC28-1781
OS/390 V1R2.0 MVS System Management Facilities (SMF)	*OS/390 MVS System Management Facilities (SMF)*	GC28-1783
OS/390 V1R2.0 MVS Planning: Operations	*OS/390 MVS Planning: Operations*	GC28-1760
OS/390 V1R2.0 MVS Planning: Global Resource Serialization	*OS/390 MVS Planning: Global Resource Serialization*	GC28-1759
OS/390 V1R2.0 MVS System Data Set Definition	*OS/390 MVS System Data Set Definition*	GC28-1782
OS/390 V1R2.0 MVS Device Validation Support	*OS/390 MVS Device Validation Support*	GC28-1748
OS/390 V1R2.0 MVS System Messages, Vol 1 (ABA-ASA)	*OS/390 MVS System Messages, Volume 1 (ABA-ASA)*	GC28-1784
OS/390 V1R2.0 MVS System Messages, Vol 2 (ASB-ERB)	*OS/390 MVS System Messages, Volume 2 (ASB-ERB)*	GC28-1785

Short Title	Title	Order Number
OS/390 V1R2.0 MVS System Messages, Vol 3 (GDE-IEB)	*OS/390 MVS System Messages, Volume 3 (GDE-IEB)*	GC28-1786
OS/390 V1R2.0 MVS System Messages, Vol 4 (IEC-IFD)	*OS/390 MVS System Messages, Volume 4 (IEC-IFD)*	GC28-1787
OS/390 V1R2.0 MVS System Messages, Vol 5 (IGD-IZP)	*OS/390 MVS System Messages, Volume 5 (IGD-IZP)*	GC28-1788
OS/390 V1R2.0 MVS Dump Output Messages	*OS/390 MVS Dump Output Messages*	GC28-1749
OS/390 V1R2.0 MVS System Codes	*OS/390 MVS System Codes*	GC28-1780
OS/390 V1R2.0 MVS Routing and Descriptor Codes	*OS/390 MVS Routing and Descriptor Codes*	GC28-1778
OS/390 V1R2.0 MVS Recovery and Reconfiguration Guide	*OS/390 MVS Recovery and Reconfiguration Guide*	GC28-1777
OS/390 V1R2.0 MVS JES Common Coupling Services	*OS/390 MVS Programming: JES Common Coupling Services*	GC28-1770
OS/390 V1R2.0 MVS IPCS Commands Summary	*OS/390 MVS Interactive Problem Control System (IPCS) Commands Summary*	GX22-0039
MVS Batch Local Shared Resources	*MVS Programming: Batch Local Shared Resources Subsystem Guide*	GC28-1469
MVS Hiperbatch Guide	*MVS Programming: Hiperbatch Guide*	GC28-1470
OS/390 V1R2.0 MVS: Writing Servers for APPC/MVS	*OS/390 MVS: Writing Servers for APPC/MVS*	GC28-1774
OS/390 V1R2.0: MVS Writing Transaction Schedulers for APPC/MVS	*OS/390 MVS: Writing Transaction Schedulers for APPC/MVS*	GC28-1776
OS/390 MVS APPC/MVS Handbook for OS/2	*OS/390 MVS APPC/MVS Handbook for OS/2*	GC28-1746
NJE Formats and Protocols	*Network Job Entry Formats and Protocols*	SC23-0070
OS/390 V1R2.0 MVS Product Registration	*OS/390 MVS Programming: Product Registration*	GC28-1729
OS/390 V1R2.0 MVS Product Management	*OS/390 MVS Product Management*	GC28-1730
OS/390 V1R2.0 MVS Diagnosis: Procedures	*OS/390 MVS Diagnosis: Procedures*	LY28-1082
OS/390 V1R2.0 MVS Diagnosis: Tools and Service Aids	*OS/390 MVS Diagnosis: Tools and Service Aids*	LY28-1085
OS/390 V1R2.0 MVS Diagnosis: Reference	*OS/390 MVS Diagnosis: Reference*	LY28-1084
OS/390 V1R2.0 MVS Planning: Workload Management	*OS/390 MVS Planning: Workload Management*	GC28-1761
OS/390 V1R2.0 MVS Workload Management Services	*OS/390 MVS Programming: Workload Management Services*	GC28-1773
OS/390 Information Roadmap	*OS/390 Information Roadmap*	GC28-1727
OS/390 V1R2.0 MVS Diagnosis: Tools and Service Aids	*OS/390 MVS Diagnosis: Tools and Service Aids*	LY28-1845
OS/390 V1R2.0 MVS Diagnosis: Procedures	*OS/390 MVS Diagnosis: Procedures*	LY28-1844
OS/390 V1R2.0 MVS Diagnosis: Reference	*OS/390 MVS Diagnosis: Reference*	LY28-1872
OS/390 Introduction and Release Guide	*OS/390 Introduction and Release Guide*	GC28-1725
SystemView for MVS Up and Running!	*SystemView for MVS Up and Running!*	GC28-1241
OS/390 Up and Running!	*OS/390 Up and Running!*	GC28-1726

Short Title	Title	Order Number
OS/390 Information Roadmap	*OS/390 Information Roadmap*	GC28-1727

- SMP/E and Installation Manuals

Short Title	Title	Order Number
ServerPac Guide and Worksheet	*ServerPac Guide and Worksheet*	SC28-1244
MVS Packaging Rules	*Standard Packaging Rules for MVS-Based Products*	SC23-3695
OS/390 SMP/E Diagnosis Guide	*OS/390 System Modification Program Extended Diagnosis Guide*	SC28-1737
OS/390 SMP/E Messages and Codes	*OS/390 System Modification Program Extended Messages and Codes*	SC28-1738
OS/390 SMP/E User's Guide	*OS/390 System Modification Program Extended User's Guide*	SC28-1740
OS/390 SMP/E Command Reference	*OS/390 System Modification Program Extended Command Reference*	SC28-1805
OS/390 SMP/E Reference	*OS/390 System Modification Program Extended Reference*	SC28-1806

- JES2 Subsystem

Short Title	Title	Order Number
OS/390 JES2 Messages	*OS/390 JES2 Messages*	GC28-1796
OS/390 JES2 Commands	*OS/390 JES2 Commands*	GC28-1790
OS/390 JES2 Initialization and Tuning Guide	*OS/390 JES2 Initialization and Tuning Guide*	SC28-1791
OS/390 JES2 Initialization and Tuning Reference	*OS/390 JES2 Initialization and Tuning Reference*	SC28-1792
OS/390 JES2 Diagnosis	*OS/390 JES2 Diagnosis*	LY28-1086
OS/390 JES2 Introduction	*OS/390 JES2 Introduction*	GC28-1794
OS/390 JES2 Installation Exits	*OS/390 JES2 Installation Exits*	SC28-1793
OS/390 JES2 Macros	*OS/390 JES2 Macros*	SC28-1795
OS/390 JES2 Migration Notebook	*OS/390 JES2 Migration Notebook*	GC28-1797

- JES3 Subsystem

Short Title	Title	Order Number
OS/390 JES3 Conversion Notebook	*OS/390 JES3 Conversion Notebook*	GC28-1799
OS/390 JES3 Diagnosis	*OS/390 JES3 Diagnosis*	LY28-1090
OS/390 JES3 Initialization and Tuning Guide	*OS/390 JES3 Initialization and Tuning Guide*	SC28-1802
OS/390 JES3 Introduction	*OS/390 JES3 Introduction*	GC28-1808

Short Title	Title	Order Number
OS/390 JES3 Initialization and Tuning Reference	*OS/390 JES3 Initialization and Tuning Reference*	SC28-1803
OS/390 JES3 Customization	*OS/390 JES3 Customization*	LY28-1089
OS/390 JES3 Messages	*OS/390 JES3 Messages*	GC28-1804
OS/390 JES3 Commands	*OS/390 JES3 Commands*	GC28-1798
OS/390 JES3 Diagnosis Reference	*OS/390 JES3 Diagnosis Reference*	LY28-1092

- ICKDSF

Short Title	Title	Order Number
ICKDSF R16 Refresh User's Guide	*ICKDSF R16 Refresh User's Guide*	GC35-0033

- Storage Environment

Short Title	Title	Order Number
DFSMS/MVS V1R3 General Information	*DFSMS/MVS Version 1 Release 3 General Information*	GC26-4900
DFSMS/MVS V1R3 Licensed Program Specifications	*DFSMS/MVS Version 1 Release 3 Licensed Program Specifications*	GC26-4903
DFSMS/MVS V1R3 Master Index	*DFSMS/MVS Version 1 Release 3 Master Index*	GC26-4904
DFSMS/MVS V1R3 Planning for Installation	*DFSMS/MVS Version 1 Release 3 Planning for Installation*	SC26-4919
DFSMS/MVS V1R3 Installation Exits	*DFSMS/MVS Version 1 Release 3 Installation Exits*	SC26-4908
DFSMS/MVS V1R3 Managing Catalogs	*DFSMS/MVS Version 1 Release 3 Managing Catalogs*	SC26-4914
DFSMS/MVS V1R1 Checkpoint/Restart	*DFSMS/MVS Version 1 Release 1 Checkpoint/Restart*	SC26-4907
DFSMS/MVS V1R3 Using Data Sets	*DFSMS/MVS Version 1 Release 3 Using Data Sets*	SC26-4922
DFSMS/MVS V1R3 Macro Instructions for Data Sets	*DFSMS/MVS Version 1 Release 3 Macro Instructions for Data Sets*	SC26-4913
DFSMS/MVS V1R3 DFSMSdfp Advanced Services	*DFSMS/MVS Version 1 Release 3 DFSMSdfp Advanced Services*	SC26-4921
DFSMS/MVS V1R3 Access Method Services for ICF	*DFSMS/MVS Version 1 Release 3 Access Method Services for the Integrated Catalog Facility*	SC26-4906
DFSMS/MVS V1R3 Access Method Services for VSAM	*DFSMS/MVS Version 1 Release 3 Access Method Services for VSAM Catalogs*	SC26-4905
DFSMS/MVS V1R3 DFSMSdfp Storage Administration Reference	*DFSMS/MVS Version 1 Release 3 DFSMSdfp Storage Administration Reference*	SC26-4920
DFSMS/MVS V1R3 Using ISMF	*DFSMS/MVS Version 1 Release 3 Using the Interactive Storage Management Facility*	SC26-4911
DFSMS/MVS V1R3 Utilities	*DFSMS/MVS Version 1 Release 3 Utilities*	SC26-4926

Short Title	Title	Order Number
DFSMS/MVS V1R3 DFM/MVS Guide and Reference	*DFSMS/MVS Version 1 Release 3 Distributed FileManager/MVS Guide and Reference*	SC26-4915
DFSMS/MVS V1R3 Program Management	*DFSMS/MVS Version 1 Release 3 Program Management*	SC26-4916
DFSMS/MVS V1R3 DFSMSdfp Diagnosis Guide	*DFSMS/MVS Version 1 Release 3 DFSMSdfp Diagnosis Guide*	LY27-9605
DFSMS/MVS V1R3 DFSMSdfp Diagnosis Reference	*DFSMS/MVS Version 1 Release 3 DFSMSdfp Diagnosis Reference*	LY27-9606
DFSMS/MVS V1R2 Using Magnetic Tapes	*DFSMS/MVS Version 1 Release 2 Using Magnetic Tapes*	SC26-4923
DFSMS/MVS V1R2 Using the Volume Mount Analyzer	*DFSMS/MVS Version 1 Release 2 Using the Volume Mount Analyzer*	SC26-4925
DFSMS/MVS V1R3 Implementing System-Managed Storage	*DFSMS/MVS Version 1 Release 3 Implementing System-Managed Storage*	SC26-3123
MVS/ESA SML: Leading a Storage Administration Group	*MVS/ESA Storage Management Library: Leading a Storage Administration Group*	SC26-3126
MVS/ESA SML: Managing Storage Groups	*MVS/ESA Storage Management Library: Managing Storage Groups*	SC26-3125
MVS/ESA SML: Managing Data	*MVS/ESA Storage Management Library: Managing Data*	SC26-3124
DFSMS/MVS V1R2 OAM Application Programmer's Reference	*DFSMS/MVS Version 1 Release 2 Object Access Method Application Programmer's Reference*	SC26-4917
DFSMS/MVS V1R3 OAM Planning, Installation, and Administration Guide for Object Support	*DFSMS/MVS Version 1 Release 3 Object Access Method Planning, Installation, and Storage Administration Guide for Object Support*	SC26-4918
DFSMS/MVS V1R3 OAM Planning, Installation, and Storage Administration Guide for Tape Libraries	*DFSMS/MVS Version 1 Release 3 Object Access Method Planning, Installation, and Storage Administration for Tape Libraries*	SC26-3051
DFSMS/MVS V1R3 DFSMSrmm Guide and Reference	*DFSMS/MVS Version 1 Release 3 DFSMSrmm Guide and Reference*	SC26-4931
DFSMS/MVS V1R3 DFSMSrmm Implementing and Customizing	*DFSMS/MVS Version 1 Release 3 DFSMSrmm Implementing and Customizing Guide*	SC26-4932
DFSMS/MVS V1R3 DFSMSrmm Diagnosis Guide	*DFSMS/MVS Version 1 Release 3 DFSMSrmm Diagnosis Guide*	LY27-9615
DFSMS/MVS V1R3 DFSMShsm Managing Your Own Data	*DFSMS/MVS Version 1 Release 3 DFSMShsm Managing Your Own Data*	SH21-1077
DFSMS/MVS V1R3 DFSMShsm Storage Administration Guide	*DFSMS/MVS Version 1 Release 3 DFSMShsm Storage Administration Guide*	SH21-1076
DFSMS/MVS V1R3 DFSMShsm Storage Administration Reference Summary	*DFSMS/MVS Version 1 Release 3 DFSMShsm Storage Administration Reference Summary*	SX26-3808
DFSMS/MVS V1R3 Managing Data Availability	*DFSMS/MVS Version 1 Release 3 Managing Data Availability*	SC26-4928
DFSMS/MVS V1R3 DFSMShsm Implementing and Customizing	*DFSMS/MVS Version 1 Release 3 DFSMShsm Implementing and Customizing*	SH21-1078
DFSMS/MVS V1R3 DFSMShsm Diagnosis Guide	*DFSMS/MVS Version 1 Release 3 DFSMShsm Diagnosis Guide*	LY27-9607

Short Title	Title	Order Number
DFSMS/MVS V1R3 DFSMShsm Diagnosis Reference	*DFSMS/MVS Version 1 Release 3 DFSMShsm Diagnosis Reference*	LY27-9608
DFSMS/MVS V1R3 DFSMShsm Storage Administration Reference	*DFSMS/MVS Version 1 Release 3 DFSMShsm Storage Administration Reference*	SH21-1075
DFSMS/MVS V1R3 DFSMSdss Storage Administration Guide	*DFSMS/MVS Version 1 Release 3 DFSMSdss Storage Administration Guide*	SC26-4930
DFSMS/MVS V1R3 DFSMSdss Storage Administration Reference	*DFSMS/MVS Version 1 Release 3 DFSMSdss Storage Administration Reference*	SC26-4929
DFSMS/MVS V1R3 DFSMSdss Diagnosis Guide	*DFSMS/MVS Version 1 Release 3 DFSMSdss Diagnosis Guide*	LY27-9609
Stand-Alone Services Overview	*Data Facility Data Set Services Version 2 Release 5 and DFSMS/MVS Version 1 Stand-Alone Services Overview*	SC26-0185
Remote Copy Administrator's Guide	*Remote Copy Administrator's Guide*	SC35-0169
DFSMS/MVS Network File System User's Guide	*DFSMS/MVS Network File System User's Guide*	SC26-7028
DFSMS/MVS Network File System Customization and Operation	*DFSMS/MVS Network File System Customization and Operation*	SC26-7029
DFSMS/MVS V1R2 Network File System Performance Tuning Guide	*DFSMS/MVS Version 1 Release 2 Network File System Performance Tuning Guide*	SC26-7019

- Security Server

Short Title	Title	Order Number
OS/390 Security Server (RACF) General User's Guide	*OS/390 Security Server (RACF) General User's Guide*	SC28-1917
OS/390 Security Server (RACF) System Programmer's Guide	*OS/390 Security Server (RACF) System Programmer's Guide*	SC28-1913
OS/390 Security Server (RACF) Macros and Interfaces	*OS/390 Security Server (RACF) Macros and Interfaces*	SC28-1914
OS/390 Security Server (RACF) Command Language Reference	*OS/390 Security Server (RACF) Command Language Reference*	SC28-1919
OS/390 Security Server (RACF) Introduction	*OS/390 Security Server (RACF) Introduction*	GC28-1912
OS/390 Security Server (RACF) Messages and Codes	*OS/390 Security Server (RACF) Messages and Codes*	SC28-1918
OS/390 Security Server (RACF) Security Administrator's Guide	*OS/390 Security Server (RACF) Security Administrator's Guide*	SC28-1915
OS/390 Security Server (RACF) Auditor's Guide	*OS/390 Security Server (RACF) Auditor's Guide*	SC28-1916
OS/390 Security Server (RACF) Diagnosis Guide	*OS/390 Security Server (RACF) Diagnosis Guide*	LY27-2639
OS/390 Security Server External Security Interface (RACROUTE) Macro Reference	*OS/390 Security Server External Security Interface (RACROUTE) Macro Reference*	GC28-1922
OS/390 Security Server (RACF) Callable Services	*OS/390 Security Server (RACF) Callable Services*	GC28-1921

Short Title	Title	Order Number
OS/390 Security Server (RACF) Planning: Installation and Migration	*OS/390 Security Server (RACF) Planning: Installation and Migration*	GC28-1920
OS/390 Security Server (RACF) Support for MVS OpenEdition DCE, SOMobjects for MVS and SystemView	*OS/390 Security Server (RACF) Support for MVS OpenEdition DCE, SOMobjects for MVS and SystemView*	GC28-1924
OS/390 Security Server (OpenEdition DCE Security Server) Overview	*OS/390 Security Server (OpenEdition DCE Security Server) Overview*	GC28-1938

- EREP

Short Title	Title	Order Number
EREP V3R5 User's Guide	*Environmental Record Editing and Printing Program Version 3 Release 5 User's Guide*	GC35-0151
EREP V3R5 Reference	*Environmental Record Editing and Printing Program Version 3 Release 5 Reference*	GC35-0152

- TSO/E

Short Title	Title	Order Number
TSO/E Administration	*OS/390 TSO/E Administration*	SC28-1966
TSO/E REXX Reference	*OS/390 TSO/E REXX Reference*	SC28-1975
TSO/E General Information	*OS/390 TSO/E General Information*	GC28-1964
TSO/E Customization	*OS/390 TSO/E Customization*	SC28-1965
TSO/E Programming Guide	*OS/390 TSO/E Programming Guide*	SC28-1970
TSO/E Programming Services	*OS/390 TSO/E Programming Services*	SC28-1971
TSO/E CLISTs	*OS/390 TSO/E CLISTs*	SC28-1973
TSO/E Guide to SRPI	*OS/390 TSO/E Guide to the Server-Requester Programming Interface*	SC28-1976
TSO/E User's Guide	*OS/390 TSO/E User's Guide*	SC28-1968
TSO/E REXX User's Guide	*OS/390 TSO/E REXX User's Guide*	SC28-1974
TSO/E System Programming Command Reference	*OS/390 TSO/E System Programming Command Reference*	SC28-1972
TSO/E Command Reference	*OS/390 TSO/E Command Reference*	SC28-1969
TSO/E Messages	*OS/390 TSO/E Messages*	GC28-1978
TSO/E Primer	*OS/390 TSO/E Primer*	GC28-1967

- Open System Adapter

Short Title	Title	Order Number
OSA Planning	*Planning for the System/390 Open Systems Adapter Feature*	GC23-3870
OSA/SF User's Guide	*OS/390 Open Systems Adapter Support Facility User's Guide*	SC28-1855

- OpenEdition DCE DFS

Short Title	Title	Order Number
DFS Administration Guide and Reference	*OS/390 OpenEdition DCE DFS Administration Guide and Reference*	SC28-1720
DFS Configuring and Getting Started	*OS/390 OpenEdition DCE DFS Configuring and Getting Started*	SC28-1722
DFS Messages and Codes	*OS/390 OpenEdition DCE DFS Messages and Codes*	SC28-1724

- ISPF

Short Title	Title	Order Number
OS/390 V1R2.0 ISPF Planning and Customizing	*OS/390 V1R2.0 ISPF Planning and Customizing*	SC28-1298
OS/390 V1R2.0 ISPF Getting Started	*OS/390 V1R2.0 ISPF Getting Started*	SC28-1294
OS/390 V1R2.0 ISPF Dialog Tag Language Guide and Reference	*OS/390 V1R2.0 ISPF DTL Guide*	SC28-1219
OS/390 V1R2.0 ISPF User's Guide	*OS/390 V1R2.0 ISPF User's Guide*	SC28-1239
OS/390 V1R2.0 ISPF Dialog Developer's Guide	*OS/390 V1R2.0 ISPF Dialog Developer's Guide and Reference*	SC28-1273
OS/390 V1R2.0 ISPF Edit and Edit Macros	*OS/390 V1R2.0 ISPF Edit and Edit Macros*	SC28-1312
OS/390 V1R2.0 ISPF Examples	*OS/390 V1R2.0 ISPF Examples*	SC28-1282
OS/390 V1R2.0 ISPF Services Guide	*OS/390 V1R2.0 ISPF Services Guide*	SC28-1272
OS/390 V1R2.0 ISPF Library Management Facility	*OS/390 V1R2.0 ISPF Library Management Facility*	SC28-1317
OS/390 V1R2.0 ISPF SCLM Project Manager's Guide	*OS/390 V1R2.0 ISPF Software Configuration and Library Manager Project Manager's Guide*	SC28-1319
OS/390 V1R2.0 ISPF SCLM Developer's Guide	*OS/390 V1R2.0 ISPF Software Configuration and Library Manager Developer's Guide*	SC28-1318
OS/390 V1R2.0 ISPF SCLM Reference	*OS/390 V1R2.0 ISPF Software Configuration and Library Manager Reference*	SC28-1320
OS/390 V1R2.0 ISPF Reference Summary	*OS/390 V1R2.0 ISPF Reference Summary*	SC28-1308

- VTAM

Short Title	Title	Order Number
VTAM Installation and Migration Guide	*VTAM Installation and Migration Guide*	GC31-6547
VTAM Release Guide	*VTAM Release Guide*	GC31-6555
VTAM Network Implementation Guide	*VTAM Network Implementation Guide*	SC31-6548
VTAM Resource Definition Reference	*VTAM Resource Definition Reference*	SC31-6552
VTAM Resource Definition Samples	*VTAM Resource Definition Samples*	SC31-6554
VTAM Customization	*VTAM Customization*	LY43-0068
VTAM Operation	*VTAM Operation*	SC31-6549
VTAM Messages and Codes	*VTAM Messages and Codes*	SC31-6546

Short Title	Title	Order Number
VTAM Programming	*VTAM Programming*	SC31-6550
VTAM Programming for LU 6.2	*VTAM Programming for LU 6.2*	SC31-6551
VTAM CMIP Services and Topology Agent Programming Guide	*VTAM CMIP Services and Topology Agent Programming Guide*	SC31-6544
VTAM Diagnosis	*VTAM Diagnosis*	LY43-0069
VTAM Data Areas for MVS/ESA	*VTAM Data Areas for MVS/ESA*	LY43-0071
APPC Application Suite User's Guide	*APPC Application Suite User's Guide*	SC31-6532
APPC Application Suite Administration	*APPC Application Suite Administration*	SC31-6533
APPC Application Suite Programming	*APPC Application Suite Programming*	SC31-6534
VTAM AnyNet Feature: Guide to Sockets over SNA	*VTAM AnyNet Feature: Guide to Sockets over SNA*	SC31-6559
VTAM AnyNet Feature: Guide to SNA over TCP/IP	*VTAM AnyNet Feature: Guide to SNA over TCP/IP*	SC31-6560
VTAM Glossary	*VTAM Glossary*	GC31-6556
Planning for NetView, NCP, and VTAM	*Planning for NetView, NCP, and VTAM*	SC31-8063
Planning for Integrated Networks	*Planning for Integrated Networks*	SC31-8062
VTAM Licensed Program Specifications	*VTAM Licensed Program Specifications*	GC31-6553
VTAM Program Directory	*VTAM Program Directory*	GC31-6403

- Other Products

Short Title	Title	Order Number
CICS Family: General Information	*CICS Family: General Information*	GC33-0155
CICS/ESA 4.1 Migration Guide	*CICS/ESA 4.1 Migration Guide*	GC33-1162
CICS/ESA 4.1 Installation Guide	*CICS/ESA 4.1 Installation Guide*	SC33-1163
CICS/ESA 4.1 System Definition Guide	*CICS/ESA 4.1 System Definition Guide*	SC33-1164
CICS/ESA 4.1 Customization Guide	*CICS/ESA 4.1 Customization Guide*	SC33-1165
CICS/ESA 4.1 Resource Definition Guide	*CICS/ESA 4.1 Resource Definition Guide*	SC33-1166
DB2 Installation Guide	*DB2 for MVS/ESA Installation Guide*	SC26-3456
DB2 Admin Guide	*DB2 for MVS/ESA Administration Guide*	SC26-3265
IMS/ESA Installation Volume 1	*IMS/ESA Installation Volume 1 Installation and Verification*	SC26-8023
IMS/ESA Installation Volume 2	*IMS/ESA Installation Volume 2 System Definition and Tailoring*	SC26-8024
IMS/ESA Customization Guide	*IMS/ESA Customization Guide*	SC26-8020
IMS/ESA Data Base	*IMS/ESA System Administration Guide: Data Base*	SC26-8012
IMS/ESA Transaction Manager	*IMS/ESA System Administration Guide: Transaction Manager*	SC26-8014
IMS/ESA System	*IMS/ESA System Administration Guide: System*	SC26-8013
NCP, SSP, EP Generation and Loading Guide	*NCP, SSP, EP Generation and Loading Guide*	SC31-6221

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number
- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Index

H

I

L

M

P

R

S

T

U

V

Printed in U.S.A.