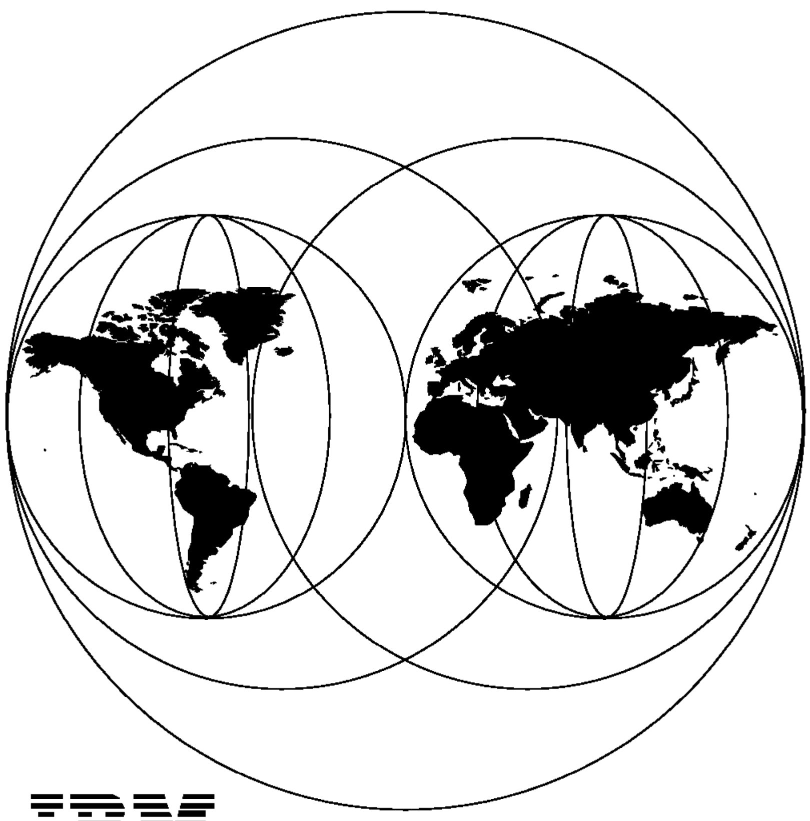
SG24-4776-00

JES3 in a Parallel Sysplex

October 1996



International Technical Support Organization
Poughkeepsie Center

International Technical Support Organization

SG24-4776-00

JES3 in a Parallel Sysplex

October 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix G, “Special Notices” on page 147.

First Edition (October 1996)

This edition applies to MVS/ESA JES3 Version 5 Release 1.1, Program Number 5655-069 and MVS/ESA JES3 Version 5 Release 2.1, Program Number 5655-069, for use with the MVS/ESA SP Version 5 Release 2.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. 541 Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

Contents

Figures

Tables

Preface

This redbook discusses the role of JES3 in a Parallel Sysplex. It focuses on those features of MVS/ESA and OS/390 that can be used by JES3 in a Parallel Sysplex, and provides information about the following topics:

JES3 CTC support has been removed from the JES3 code and is replaced with XCF signaling services and JESXCF.

JES3 console services and JES3-only consoles have been removed with JES3 5.2.1 and are replaced with MCS multisystem console support.

The JES3 MLOG is removed and can be replaced with either the MVS system logger or a new JES3 DLOG.

The following new MVS sysplex functions supported by JES3 in the sysplex environment are discussed:

- Automatic Restart Management (ARM)
- Shared tape support and JES3 tape management
- System symbolics

S/390 hardware and software running as a Parallel Sysplex offers a unique continuous availability platform, handling both fault tolerance and system management issues. The following issues are discussed in the context of continuous availability planning: DFSMS and dynamic I/O reconfiguration, and the advantages of JES3 main device scheduling.

Migration issues that need to be addressed when converting to the sysplex environment are described.

This redbook was written for systems programmers and IBM personnel involved in implementing a JES3 Parallel Sysplex.

How This Redbook Is Organized

The redbook contains 156 pages. It is organized as follows:

- Chapter 1, “Introduction to Sysplex”

  This chapter describes the base sysplex and the Parallel Sysplex environments and introduces how JES3 is implemented in this environment.

- Chapter 2, “JES3 Concepts and the Sysplex”

  This chapter describes the basic JES3 concepts and describes the JES3 functions that have now become part of the base MVS system.

- Chapter 3, “JES3 Support of MVS Sysplex Functions”

  This chapter describes the new functions of MVS that have been added in support of the sysplex environment. They include:

  – Automatic Restart Management (ARM)
  – Shared tape support versus JES3 tape management
  – System symbolics (cloning)

- Chapter 4, “JES3 Continuous Availability Issues”

This chapter discusses the availability and configuration issues with JES3 in a Parallel Sysplex.

- Chapter 5, “Parallel Sysplex Migration Considerations”

  There are many important migration decisions that need to be made during a conversion to the sysplex environment. This chapter discusses these issues.

- Appendix A, “Sysplex Terminology”

  This appendix describes the sysplex terminology and definitions.

- Appendix B, “Sysplex Evolution”

  This appendix summarizes all the MVS release changes and the MVS enhancements in chronological order.

- Appendix C, “JES3 Version 5 and JES Common Coupling Services”

  This appendix provides information on JES common coupling services (JESXCF), XCF services, XES services, and gives an overview of the JES3 and the JES common coupling services interactions. These components provide the basis for a Parallel Sysplex environment.

- Appendix E, “Sample Program Using JESXCF Services”

  This appendix describes a sample JES3 DSP that uses JESXCF services to monitor JES3 local systems.

- Appendix F, “MCS Sysplex Operations”

  This appendix discusses the many changes to MCS in support of the sysplex environment.

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.

Paul C. Rogers works at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of MVS and JES.

Juha Vainikainen IBM Finland

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. Introduction to Sysplex

A sysplex is a set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.

A sysplex implementation without a coupling facility is referred to as a *base sysplex*. When the implementation includes a coupling facility, it is called a *Parallel Sysplex*.

See Appendix A, "Sysplex Terminology" on page 87 for the definitions of the sysplex terminology and the sysplex modes of operation.

The base sysplex lays the groundwork for simplified multisystem management through the cross-system coupling facility (XCF). In a base sysplex, central processing complexes (CPC) are connected through channel-to-channel (CTC) adapters and a shared coupling data set. When the sysplex consists of multiple MVS systems running on two or more processors, MVS requires that the processors be connected to the same sysplex timer. MVS uses the sysplex timer to synchronize TOD clocks across systems. The base sysplex allows up to eight MVS systems in a sysplex.

MVS/ESA SP 5.1 extends the sysplex to provide high-performance data sharing through a coupling technology. MVS/ESA SP 5.1, together with JES3 5.1.1, increases the number of MVS systems in a sysplex from eight to 32. The capability of linking many systems and providing multisystem data sharing makes the sysplex ideal for parallel processing, particularly for online transaction processing.

This redbook describes how to implement JES3 in a Parallel Sysplex environment and how JES3 customers can exploit the common functions of MVS that have been implemented for the Parallel Sysplex environment. It applies to all releases of MVS/ESA Version 5 and OS/390 Release 1.

See Appendix B, "Sysplex Evolution" on page 91 for the changes implemented in MVS for the Parallel Sysplex environment.

1.1 Base Sysplex

The concept of a base MVS sysplex (**sys**tems com**plex**) has been available since 1990 as a platform for an evolving large system computing environment. A sysplex, as announced in MVS/ESA SP Version 4, consists of multiple MVS systems coupled together by hardware elements and MVS software services.

The base sysplex is similar to a JES3 loosely coupled configuration in that more than one CPC shares direct access devices (DASD) and is managed by more than one MVS image. A sysplex is different from a loosely coupled configuration because there is a single standard communication mechanism, the MVS cross-system coupling facility (XCF), that can shared by all MVS components, authorized applications, and subsystems. XCF allows up to 32 MVS systems to communicate in a sysplex environment. The communications and control of these systems is accomplished as if the individual systems were a single system. This single system is an MVS sysplex.

1.1.1 Base Sysplex Hardware Elements

The base sysplex hardware elements include a sysplex timer, channel-to channel I/O interfaces, and shared direct access devices:

- The sysplex timer is a unit that synchronizes the time-of-day (TOD) clocks between multiple CPCs in a sysplex. The time stamp from the sysplex timer is a way for XCF to monitor and sequence events within the sysplex. If a sysplex is implemented within a single CPC in multiple logical partitions (LPAR), the sysplex timer is not required.
- The XCF inter-processor communication, XCF signaling, is implemented through CTC connections (S/370 extended mode) and are one directional. That is, on each system, messages sent to other systems require an outbound path, and messages received from other systems require a separate inbound path. XCF signaling within an MVS image does not require CTCs.

  XCF signaling paths can be added and deleted dynamically with the SETXCF operator command.
- XCF requires a DASD resident sysplex couple data set (and an alternate data set is recommended for availability) to be shared by all systems in the sysplex. On the sysplex couple data set, XCF stores information related to the sysplex, systems in the sysplex, and the multisystem applications using XCF services.

  A sysplex couple data set is always required for a multisystem sysplex and for most single-system sysplexes. Note, however, that you can define a single system sysplex that does not require a sysplex couple data set.

1.2 Parallel Sysplex

Since the introduction of the sysplex in MVS/ESA SP Version 5, IBM has made available technologies that enhance sysplex capabilities. The sysplex supports a greater number of systems (32 instead of eight) and significantly improves communication and data sharing among the systems in the Parallel Sysplex through a coupling facility which otherwise could be technically difficult.

The coupling facility provides shared storage and shared storage management functions for the sysplex (for example, high speed caching, list processing, and locking functions). A coupling facility is a special logical partition on certain ES/9000 model processors and on CPCs in the S/390 microprocessor cluster. Resource managers, for example, IMS DB and DB/2, running on MVS images in the sysplex define the shared structures used in the coupling facility.

The MVS cross-system extended services (XES) provides formally defined programming services for data sharing with integrity and consistency to authorized applications through the coupling facility in a sysplex.

The capability of linking together many systems and providing multisystem data sharing extends the sysplex platform for parallel processing, particularly for online transaction processing (OLTP) and decision support.

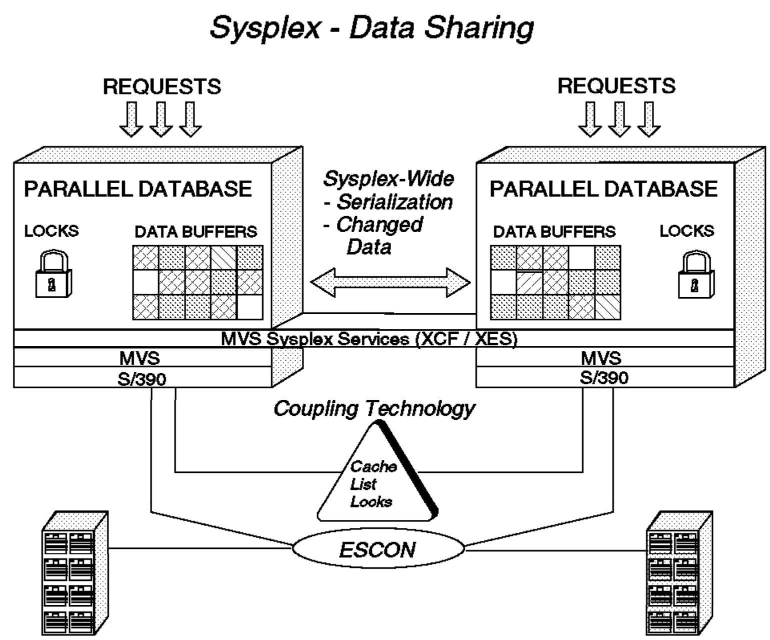


Figure 1. Parallel Sysplex Data Sharing

Authorized applications, such as subsystems and MVS components in the sysplex, can use the coupling facility services to cache data, share queues and status, and access sysplex lock structures in order to implement high performance data sharing and rapid recovery from failures. The subsystems and components transparently provide the data sharing and recovery benefits to their applications.

1.2.1 Parallel Sysplex Data Sharing

Some IBM data management systems that are using the coupling facility include database managers and a data access method.

- IMS DB is IBM′s strategic hierarchical database manager. It is used for numerous applications that depend on its high performance, availability, and reliability.

  IMS database managers on different MVS systems can access data at the same time. By using the coupling facility in a sysplex, IMS DB can efficiently provide data sharing for more than two MVS systems and thereby extends the benefits of IMS DB data sharing. IMS DB uses the coupling facility to centrally keep track of when shared data is changed. IMS/VS resource lock manager (IRLM) is still used to manage data locking, but does not notify each IMS DB of every change. IMS DB does not need to know about changed data until it is ready to use that data.

- DB2 is IBM′s strategic relational database manager.

  DB2 data sharing support allows multiple DB2 subsystems within a sysplex to concurrently access and update shared databases. DB2 data sharing uses the coupling facility to lock, to ensure consistency, and to buffer shared data. Similar to IMS, DB2 will serialize data access across the sysplex through IRLM locking. DB2 cache structures are also used to buffer shared data within a sysplex for improved sysplex efficiency.

- VSAM, a component of DFSMS/MVS, is an access method rather than a database manager. It is an access method that gives CICS and other application programs access to data stored in VSAM-managed data sets.

  DFSMS/MVS Version 1.3 supports a new VSAM data set accessing mode called record level sharing (RLS). RLS uses the coupling facility to provide sysplex-wide data sharing for CICS and the other applications that use the new accessing mode. By controlling access to data at the record level, VSAM enables CICS application programs running in different CICS regions, and in different MVS images, to share VSAM data with complete integrity. The coupling facility provides the high performance data sharing capability necessary to handle the requests from multiple CICS regions.

In short, the Parallel Sysplex builds on the base sysplex capability, and allows an increase in the number of CPCs and MVS images that can directly share work through data sharing. The coupling facility enables high performance, multisystem data sharing across all the systems. Through the exploitation of the coupling facility services, many IBM resource managers enhance single system image for their users. In addition, in a JES3 complex, work can be scheduled and balanced across systems without consideration as to whether the resources controlled by other resource managers are required by the work's MVS execution and if they are available on the selected JES3 main processor.

For MVS release to release sysplex evolution see Appendix B, "Sysplex Evolution" on page 91 and for details on sysplex definitions see Chapter 1, "Introduction to Sysplex" on page 1.

1.3 JES3 and the Parallel Sysplex

This redbook focuses on those features of MVS/ESA Version 5 and OS/390 Release 1 that can be used by JES3 in a Parallel Sysplex. It discusses the following changes to MVS and JES3 for the sysplex environment.

Two release changes for Version 5 are available for JES3 with functional changes for the sysplex environment.

1.3.1 JES3 Version 5.1.1 Enhancements for Sysplex

MVS/ESA SP 5.1 enhancements for sysplex are described on page 97. MVS/ESA JES3 Version 5 Release 1.1, available only for MVS/ESA SP 5.1 and later MVS and OS/390 releases, responded to the MVS enhancements by providing:

- 32-way sysplex enablement changes are required to allow more than eight JES3 main processors in a sysplex. The expanded JES3 complex can include a variety of processors, high-end to low-end, as long as processors of sufficient processing power are available to assume the role of the global processor and in the event of a dynamic system interchange (DSI) of the new global. Figure 2 shows a very large sysplex containing a 32-way JES3 5.1.1 global/local configuration.

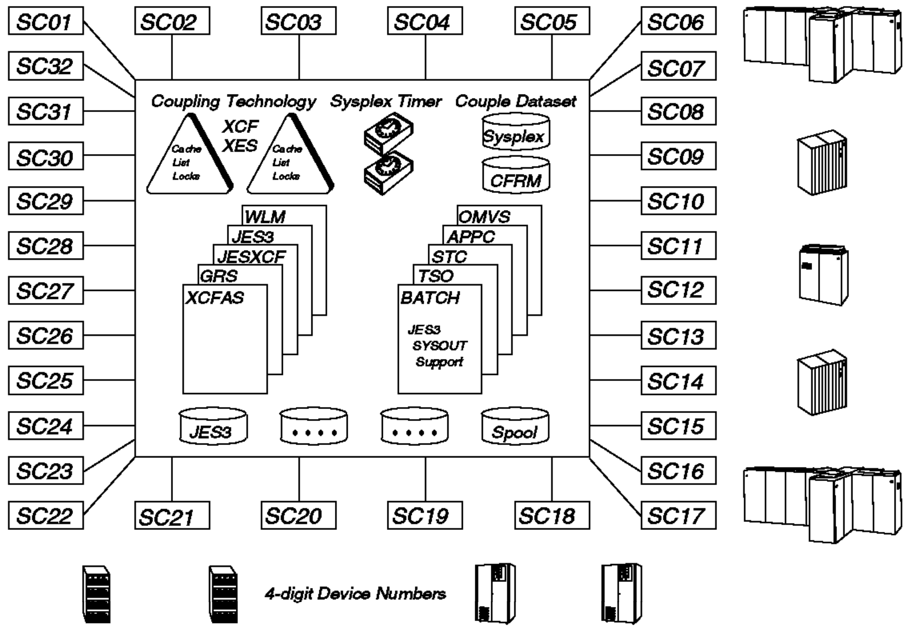


Figure 2. JES3 32-Way Sysplex

Note: DFSMS/MVS Version 1.3 is also required to support a greater than eight main processor JES3 complex when an installation is running with SMS-managed resources. Before DFSMS/MVS Version 1.3, an SMS complex could be specified to have only up to eight system names.

- JES3 5.1.1 allows 4-digit device number specifications. This allows installations to have a single I/O configuration for all their MVS systems because unique device numbers can be specified for each physical device.

  The support for 4-digit device numbers is compatible with the specification of 3-digit device numbers. All device numbers displayed will appear as four digits. Commands, initialization statements, and write-to-operator responses will accept both 3-digit and 4-digit device numbers. On commands and statements where device numbers and device types might be confused, a preceding slash (/) will be used to indicate device numbers.

  Messages and fields in control blocks that define a device number now accommodate 4-digit device numbers.

- Systems management enhancements (usability improvements to several JES3 operator commands and migration aids) assist in the definition of I/O devices in the JES3 initialization stream.

  Two new ISPF macros are provided in SYS1.SAMPLIB to alleviate the work involved in modifying the DEVICE initialization statement. The IATDEVA macro allows you to add new DEVICE statements (a single device, a list of devices, or a range of devices) by specifying a “model” DEVICE statement. The IATDEVC macro allows you to add or delete a main or group of mains automatically to a group of DEVICE statements.

  JES3 5.1.1 no longer requires CPUID numbers be identified to JES3 initialization. The CPUID= and MODEL= parameters are removed from the MAINPROC statement. The JES3 system name now instead must match the

MVS system name as specified on the SYSNAME= parameter in the IEASYSxx parmlib member.

The CPUID, which was specified on the MAINPROC initialization statement, identified to JES3 what system was being IPLed. Each time the CPUID was changed on a processor, the MAINPROC statement had to be updated and a JES3 warmstart done, which requires each system to be IPLed. This was particularly difficult for installations running PR/SM. All potential CPUID numbers had to be included in the initialization stream to avoid having to warmstart JES3 with each CPUID change.

- An availability enhancement is improved JES3 restart for Main Device scheduler (MDS), which is achieved by adding multiple secondary SETUP (MDS) FCTs.

1.3.2 JES3 5.2.1 Enhancements for Sysplex

JES3 5.2.1 provides enhancements for the following:

- Support for MVS sysplex operations - JES3 5.2.1 improves systems management and automated operations by enabling the sysplex operations features of MVS/ESA. System and subsystem code can build on these features without the need to limit function or provide alternative implementations in the JES3 environment. Sysplex-wide control is available from consoles on any system rather than a subset of the console on the global processor. JES3 command processors are changed to preserve the command and response token (CART) and appropriately identify responses, enabling improved automated operations. For details on MVS sysplex operation, see Appendix F, "MCS Sysplex Operations" on page 137.

  The MVS/ESA SP 5.2 system logger facility may be used to provide a sysplex-wide log containing considerably more information than the current JES3 DLOG. The JES3-managed operator and MLOG consoles are no longer supported.

  Prior to JES3 5.2.1, MVS initialized with the MCS sysplex environment disabled, preventing JES3 installations from making full use of the MVS sysplex functions. As a result, only consoles attached to the JES3 global processor could direct commands to, and receive messages from, all systems in the JES3 complex.

- Support for MVS/ESA SP 5.2 automatic restart management (ARM) is provided. The automatic restart manager allows automatic restart of batch jobs and started tasks. If a job registers with the automatic restart manager, the automatic restart manager restarts it according the installation-defined policy when either:

  - The executing job unexpectedly ends.
  - The system on which the job is executing unexpectedly fails or leaves the sysplex.

  The JES3 ARM support is in addition to the existing job failure option processing, which allows a job to be restarted if the FAILURE=RESTART parameter is specified on the //*MAIN JECL statement or CLASS initialization statement or STANDARDS initialization statement and the system was IPLed.

- Support for a UCB above 16 Mb in common virtual storage is provided. A unit control block (UCB) contains device information, and the system creates a UCB for each device definition. JES3 5.2.1 JES3 supports the placement of UCBs above 16 Mb in common virtual storage for the following devices:

- Spool
- Checkpoint
- JES3-managed execution devices (for example, tape and DASD used to satisfy execution job requirements)
- JES3 global devices (for example, tapes used by the dump job (DJ) facility).

During JES3 initialization, a data extent block (DEB) is built for each JES3-managed global device. Prior to JES3 5.2.1, DEBSUCBA was initialized with a UCB address of the corresponding device. Beginning with this release, DEBSUCBA is initialized only during JESOPEN processing.

- Support for system symbols in JES3 commands and source JCL for demand select jobs. JES3 5.2.1 allows systems in the JES3 complex to share commands and demand select jobs, while retaining unique values where required. When all systems in a JES3 complex share JES3 commands and demand select jobs, the environment can be viewed as a single image with one point of control.

  The ability to share JES3 commands and demand select jobs that specify unique values:

  - Reduces the number of commands entered and the number of source JCL data sets
  - Ensures that installations specify unique values on each system for resources that are specified in JES3 commands and demand select jobs
  - Helps to establish meaningful and consistent naming conventions for system resources.

1.4 Parallel Sysplex Implementation with JES3

The remaining chapters and the appendix discuss the following sysplex environment for a JES3 complex.

1.4.1 JES3 Functions Now Part of MVS

The following functions are now part of MVS and have been removed from JES3:

- JES3 multisystem communication

  JES3 CTC support has been removed from the JES3 code and is replaced with:

  - XCF signaling services (CTC definitions) and JESXCF. See 2.2.2, "JES3 Sysplex Multisystem Communication" on page 17.

- Consoles using MCS multisystem support

  JES3 console services and JES3-only consoles have been removed with JES3 5.2.1 and are replaced with MCS multisystem console support. See 2.4, "Consoles Using MCS Multisystem Support" on page 31.

- JES3 MLOG and the MVS system logger

  The JES3 MLOG is removed and can be replaced with:

  - The MVS system logger. See 2.5, "MVS System Logger" on page 37.
  - A new JES3 DLOG. See 2.5.2, "JES3 5.2.1 DLOG" on page 38.

1.4.2 JES3 and MVS Sysplex Functions

The following MVS sysplex functions are supported by JES3 in the sysplex environment and are described in Chapter 3, "JES3 Support of MVS Sysplex Functions" on page 41:

- Automatic Restart Management (ARM)
- Shared tape support versus JES3 tape management
- System symbolics (cloning)

1.4.3 JES3 Continuous Availability Issues

A major design goal for the System/390 Parallel Sysplex is to provide a continuously available environment with scalable parallel servers, building on the robustness of MVS and System/390.

Consider the following issues if you have major concerns with continuous availability planning:

- DFSMS, DEVICE statements, and Dynamic I/O for JES3-managed devices. See 4.1, "Availability and JES3-Managed Devices" on page 51.
- Dynamic I/O reconfiguration is not supported for JES3-managed devices. See 4.1.1, "Dynamic I/O Reconfiguration" on page 52.
- GRS in a JES3 sysplex environment. See 4.1.2, "Global Resource Serialization (GRS)" on page 52.
- DSI in a sysplex and flush code. See 4.5, "JES3 Local Main Failures" on page 62 and 4.5.1, "JES3 DSI Processing" on page 62.

1.4.4 Parallel Sysplex Migration Considerations

The following are migration considerations for a JES3 sysplex environment:

- Parallel Sysplex application environment

  See 5.1, "Parallel Sysplex Application Environment" on page 63.

- Sysplex Naming Conventions

  See 5.2, "Sysplex Naming Conventions" on page 66.

- JES3 Parallel Sysplex Migration

  There are many migration issues that need to be addressed when converting to the sysplex environment. See 5.3, "JES3 Parallel Sysplex Migration" on page 67. Consider the following functions when converting JES3 to the sysplex environment:

  – Consoles using MCS multisystem support

    - Command processing

      See 5.6, "JES3 5.2.1 Command Processing" on page 74.

    - Message processing

      See 5.8, "JES3 5.2.1 Message Processing" on page 77.

- Running multiple globals in same sysplex

  See 5.4, "Multiple JES3 Complexes in the Same Sysplex" on page 68.

- JES3 initialization changes for sysplex

  See 5.5, "JES3 Initialization Changes for Sysplex" on page 70.

- Console processing user considerations

  See 5.9, “JES3 5.2.1 Enhanced Console Processing User Considerations” on page 81.

- Defining the sysplex environment couple data sets

  See Appendix D, “Coupling Facility for XCF Communication” on page 125.

Chapter 2. JES3 Concepts and the Sysplex

A major goal of the MVS/ESA operating systems is to process work while making the best use of system resources. Today, installations process different types of work with different completion and resource requirements. Every installation wants to make the best use of its resources, maintain the highest possible throughput, and achieve the best possible system responsiveness. Thus, one way of viewing the MVS operating system is as a resource manager.

With the MVS system, resource management and workflow management are shared between MVS, the job entry subsystem (JES3), and other resource managers such as the Information Management System/Enterprise Systems Architecture Database Manager (IMS DB), the IBM DATABASE 2 (DB2), and the IBM virtual storage access method (VSAM). Generally speaking, JES3 does resource management and workflow management before JES3-registered work enters MVS processing and after the work completes processing, while MVS and the other resource managers do resource and workflow management while the work is in MVS processing.

Figure 3 summarizes the scope of JES3 resource management. Note that JES3 resource management is applied only for work that enters the system through JES3 or through the address spaces that are known to JES3.

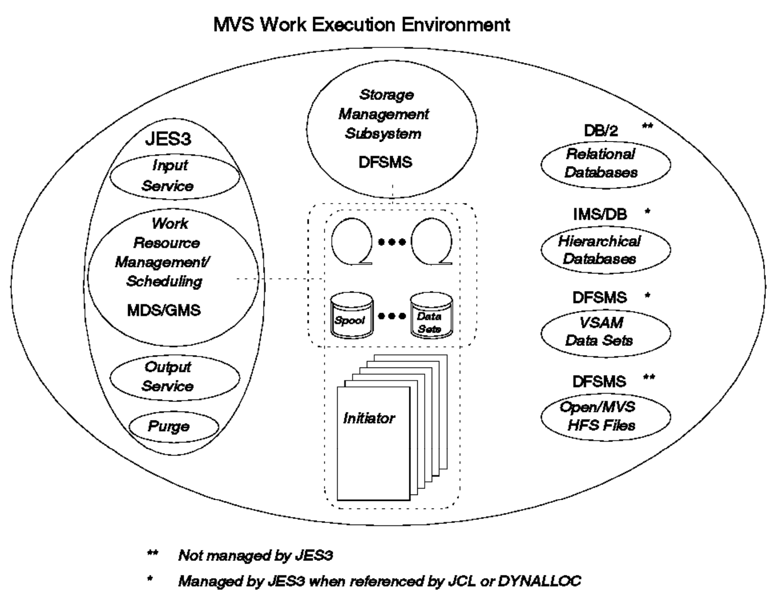


Figure 3. JES3 and Other Resource Managers

To get your MVS system to do work for you, you must describe to the system the work you want done and the resources you need through job control language (JCL). JES3 resource management for work is based on these descriptions.

After registering work and before passing it to MVS processing, JES3 makes sure that the JES3-managed resources required by the work are available. Typically these resources include devices, data sets on the devices requested through JCL, and resources required by the execution of the work, such as MVS initiators. During the work′s MVS execution, the MVS operating system and its workload manager (WLM) manages resources such as processor time, storage, and I/O for work. If JES3-managed resources are dynamically requested or released by work, JES3 updates the work′s resource requirements correspondingly.

In general, JES3 does not directly keep track of the resources requested by the work during the MVS execution from the other resource managers. For example, if an IMS batch message processing region requests access to a database, the IMS control region allocates the database dynamically if it is not already allocated. JES3 accounts the request for the IMS control region, rather than the work that caused the database to be allocated.

After the work completes processing, MVS and JES3 free all resources used by the completed work during MVS execution, making the resources available to other work. JES3 also processes the spooled output created by the work before purging the unit of work.

2.1 JES3 Job Flow

There are no changes to the JES3 flow of work through the system in a Parallel Sysplex. As JES3 has always provided a single system image through its global processor, the advantages of the Parallel Sysplex environment for work in execution is with data sharing applications as discussed in 1.2.1, "Parallel Sysplex Data Sharing" on page 3 and workload balancing with CICS/ESA 4.1 and CP/SM 1.1.1.

2.1.1 Input Service

JES3 input service accepts and queues all work entering the JES3 system. The global processor reads the work definitions into the system and creates JES3 job structures for them. Work is accepted from:

- A TSO SUBMIT command
- A local card reader
- A local tape reader
- A disk reader
- A remote work station
- Another node in a job entry network
- The internal reader

In addition, the request and return job ID subsystem interface calls allow an authorized address space, that is not known to JES3, to establish or return a job structure. Once the caller receives a job ID, the address space can use JES3 services, such as allocating a SYSOUT data set or submitting work through an internal reader.

2.1.2 Converter/Interpreter Processing

After input service processing, a job passes through the JES3 converter/interpreter processing (C/I). As a result, JES3 learns about the resources the job requires during execution. C/I routines provide input to the main device scheduling (MDS) routines by determining available devices, volumes, and data sets. These service routines process the job′s JCL to create control blocks for setup and also prevent jobs with JCL errors from continuing in the system. The C/I section of setup processing is further divided into three phases:

- MVS converter/interpreter processing
- Prescan processing
- Postscan processing

The first two phases can occur in either the JES3 address space on the global processor or in the C/I functional subsystem address space on either the local or the global processor.

2.1.3 Job Resource Management

The next phase of JES3 job processing is called job resource management. The job resource management function provides for the effective use of system resources. JES3 main device scheduler (MDS) functions, alias the setup, ensure the operative use of non-sharable mountable volumes, eliminates operator intervention during job execution, and performs data set serialization. It oversees specific types of pre-execution job setup and generally prepares all necessary resources to process the job. The main device scheduler routines use resource tables and allocation algorithms to satisfy a job′s requirements through the allocation of volumes and devices, and, if necessary, the serialization of data sets.

2.1.3.1 JES3 Setup

JES3 resource management, commonly referred to as “setup,” is a unique JES3 function and differentiates it from JES2. Setup is active on the global processor and manages the entire JES3 complex by matching work with available resources. The managed resources include processors, I/O devices, volumes, and data sets. When DFSMS/MVS is installed and active, JES3 and DFSMS cooperate to provide resource management for data sets on the volumes defined to DFSMS. The key JES3 resource management functions, main device scheduler and generalized main scheduling, ensure that all JES3-and DFSMS-managed resources are also available before scheduling work for MVS processing, to avoid MVS processing delays that may result when these resources are not readily available.

The resource management algorithms exercised by JES3 primarily consider the complex-wide resource use attributes (shareable/serially reusable) and the intended use (shared/exclusive) by the work units (a job with one or more steps) when allocating resources to work. The JES3 global maintains the resource tables, defined through initialization parameters, and the status of the resources for the entire JES3 complex. It also interfaces with all other processors through the channel-to-channel communication and provides centralized JES3 services and resource management for dynamic allocation requests for work in MVS execution. Thus, the JES3 architecture of a global processor, centralized resource and workflow management, and centralized operator control provide a single-system image for workflow and operations control.

2.1.3.2 MDS Data Set Management

DASD volume data sets are managed for integrity by JES3 when the DASD volumes are defined in the initialization stream using the DEVICE statement. There are two types of data sets for which JES3 cannot provide integrity:

- New generations of Generation Data Groups

A job creates a new generation and JES3 only knows about the relative name (for example BASENAME (+1)). Another job later allocates the same data set by specifying its fully-qualified data set name (for example BASENAME.GxxxxVyy). To maintain integrity, it would be necessary to ensure that the two jobs do not access the data set concurrently.

- New non-specific non-private DASD requests for non-SMS managed data sets

  JES3 does not provide data set integrity for new non-specific non-private requests for non-SMS managed DASD data sets. These requests either omit the VOLUME parameter, or have a VOLUME parameter which does not contain the SER, REF, or PRIVATE subparameter. See 4.4, "JES3 SMS Interactions" on page 57.

Since global resource serialization (GRS) is required in a sysplex environment, GRS could be used to provide data set integrity for these two types of data sets. GRS can provide data set integrity across multiple MVS systems in a JES3 complex through managing the SYSDSN enqueue, or an equivalent product can be used. Managing data set integrity by GRS or an equivalent product does not necessarily mean that you do not want JES3 to manage the data set resources of batch jobs. Benefits of JES3 data set management are still potentially required to prevent scheduling of workload that would interlock while on the MVS initiators.

GRS could be used to provide data set protection for all data sets by specifying:

- SETUP=NONE

on the STANDARDS JES3 initialization statement

2.1.3.3 MDS Processing Options

JES3 setup processing is defined by JES3 initialization statement parameters, JCL control statements, and JES3 operator commands. JES3 setup is available through the following options:

Job Setup: Job setup allocates enough devices to pre-mount all the required volumes prior to MVS execution of a job. This saves time required for mounting volumes after the job starts executing, but might cause a large number of devices to be allocated to a single job. The advantage is that little or no operator intervention is necessary between job steps.

If job setup is not implemented, volumes are mounted as the job enters its execution cycle, except in the case of high-watermark setup, where the first volume is mounted before execution.

High-watermark Setup (HWS): HWS uses fewer devices than job setup when allocating devices to a single job. Where possible, volumes are pre-mounted.

Generally speaking, for each device type JES3 first finds the job step that will use the most devices. JES3 then allocates to the entire job the number of devices needed to service that step. High-watermark setup is available for tape, DASD, or all JES3-managed devices.

Explicit Setup: Explicit setup allows the work originator to indicate to JES3 the DDNAMES that are or are not to be set up. This type of setup combines some advantages of job setup and high-watermark setup. Only the device pre-mount characteristic is affected by specifying explicit setup, which is mutually exclusive with any high-watermark setup. The device allocation for job setup is the default when explicit setup is specified.

2.1.3.4 MDS Breakdown

Main device breakdown or the deallocation routines process job step completion and job termination. Breakdown returns resources used during execution which were not freed during step or job execution. For dynamic allocation, breakdown takes place when dynamic deallocation is requested or when the job ends.

2.1.4 Generalized Main Scheduling

Once a job is set up, it enters JES3 job scheduling. JES3 job scheduling is the group of services that govern where and when MVS execution of a JES3 job occurs. Job scheduling controls the order and execution of jobs running within the JES3 complex.

Job Scheduling routines are a vital part of JES3 processing. Main service selects a job for execution using the job selection algorithms established at JES3 initialization through the installation′s initialization statement definitions:

- Generalized main scheduling (GMS) routines will schedule a job after it is placed on the selection queue, control the workload, and maximize system throughput.

  GMS allows an installation many controls over job scheduling for work flow management without requiring operator action or intervention. Some of these controls are:

  - Job priority
  - Job interaction within a group
  - Explicit and implicit processor dependency
  - Logical storage size
  - Class mix
  - I/O rate differentiation
  - Initiator availability
  - Related job sequencing or dependent job control (DJC)

2.1.4.1 Deadline Scheduling

Deadline scheduling schedules jobs based upon time intervals, the time of the day, and associated job priority changes.

2.1.4.2 Dependent Job Control

Dependent job control (DJC) is a means of relating the jobs that make up an application. The term DJC network denotes a set of related jobs that must run in a predetermined order. Often data set dependencies determine the ordering of jobs, though there may be other reasons as well.

2.1.5 Job Execution

Jobs are scheduled to the waiting initiators on the JES3 main processors. For the sysplex environment, MVS/ESA Version 5 allows the use of workload management (WLM) to allow MVS through the SRM to dynamically optimize resources to address spaces according to goals defined for the work in a service policy. See workload management implementation on page 98.

2.1.6 Output Processing

The final part of JES3 job processing is called job output and termination. Output service routines operate in various phases to process sysout data sets destined for print or punch devices (local, RJP, or NJE), TSO users, internal readers, external writers, and writer functional subsystems.

2.1.7 Purge Processing

Purge processing represents the last JES3 processing step for any job. It releases the resources used during the job and uses the system management facility (SMF) to record statistics.

2.2 JES3 Multisystem Communication

After migrating to JES3 Version 5 and with one or more MVS images in the complex, a sysplex environment is required. The MVS multisystem mode sysplex is required if the JES3 complex has more than one JES3 main.

This redbook concentrates on the multisystem environment because of the nature of the Parallel Sysplex and its data sharing capabilities.

Prior to MVS/ESA JES3 Version 5 Release 1.1, a JES3 complex could have at most eight interconnected MVS images in a loosely coupled configuration. A loosely coupled configuration has more than one central processing complex (CPC) running MVS, sharing DASD, and other I/O devices. The CPCs are also interconnected by JES3-managed channel-to-channel (CTC) communication. JES3 distributes work from a shared job queue to each MVS image.

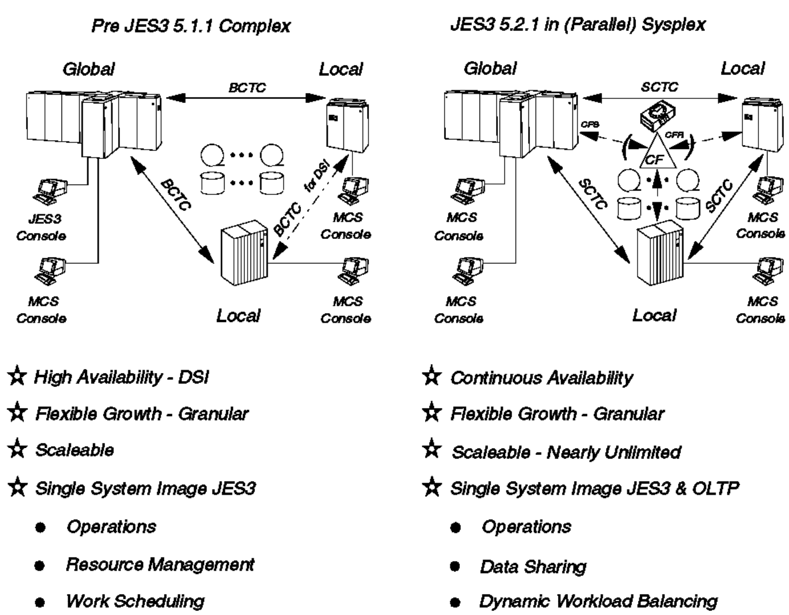


Figure 4. JES3 and Sysplex Topology

The left side of Figure 4 summarizes a pre-JES3 5.1.1 configuration.

One MVS image in the JES3 complex acts as the focal point for the entry and distribution of work and for the control of resources needed by the work. This MVS image, called the *global*, distributes work to the other processors, called *locals*.

The right side of Figure 4 on page 16 shows a JES3 5.2.1 configuration implemented as a sysplex. Note that a base sysplex does not have a coupling facility.

2.2.1 Pre-JES3 5.1.1 Communication

A JES3 complex that contains more than one MVS image uses channel-to-channel (CTC) adapters to connect the images in order to communicate information between them.

Prior to JES3 5.1.1, the JES3 CTC communication is JES3-exclusive and cannot be used by other products and applications in the complex. Products and applications that need to communicate and are running on separate systems have to create their own communication mechanism. These varied communication mechanisms added to the difficulty of managing a loosely coupled configuration.

2.2.2 JES3 Sysplex Multisystem Communication

In JES3 5.1.1, the JES3-managed CTCs for global/local communications are eliminated and JES3 communication between MVS images is changed to use the more flexible MVS XCF signaling services through the JES common coupling services (JESXCF). JESXCF, a MVS BCP component, provides services for both JES3 and JES2.

Because the JES3-managed CTCs are replaced with XCF signaling services, all systems in a JES3 complex must be at JES3 5.1.1 or higher level, and the JES3 complex must be fully contained within the same sysplex.

Installations migrating to JES3 Version 5 may gain CTC hardware savings due to the removal of the dedicated JES3-managed CTCs for communication between MVS images or, at least, they may observe a better CTC hardware availability and utilization when the JES3 CTCs are reassigned to the XCF signaling use where they can be shared by all applications requiring communication between images. Also, the system programmers′ task is simplified as they do not have to maintain multiple separate sets of CTC definitions.

XCF provides the MVS services that allow authorized programs, in a multisystem environment, to communicate (send and receive data) with programs on the same MVS system or other MVS systems. The XCF communication paths can be either CTCs or the coupling facility.

For a JES3 multisystem sysplex environment, this XCF support:

- Reduces the hardware investment to run JES3
- Allows through ESCON CTCs for improved monitoring of the hardware and a greater distance between processors in a sysplex
- Allows for multiple communication paths between MVS images concurrently, which reduces the potential for single points of failure
- Communication paths between MVS images can be added dynamically

2.2.3 JES3 and JESXCF Services

Migration to JES3 Version 5 requires a sysplex environment to be active. In a global-only JES3 complex, this requirement is fulfilled automatically. When MVS/ESA SP 5.1 is IPLed, the set of XCF services required by JESXCF is available by definition. In a global/local JES3 complex, the MVS system on which JES3 mains are running must be in the same multisystem sysplex.

2.2.3.1 XCF Services

The sysplex software services, provided by XCF, allow authorized applications or subsystems, running on the same system or on different systems in a sysplex, to communicate with each other and share status information. Each instance of an application can use XCF services to communicate with each other. They can:

- Send messages to and receive messages from each other
- Inform others of their status, for example active or failed
- Obtain information about the status of the other instances of the application

Applications that use XCF services can be viewed as a single entity and can provide a single system image for their users. For example, MVS console services (of the communication task component) allow the installation to operate all MVS systems in a sysplex from any console, similar to JES3 console support that allowed a JES3-managed console to communicate with any system in the JES3 complex.

For additional information on XCF, see C.1, “XCF Concepts” on page 107 and C.1.1, “XCF Services” on page 108.

2.2.3.2 JESXCF Services

JESXCF is an XCF application and provides, based on XCF coupling services, common inter-processor and intra-processor communication services for both JES3 and JES2 subsystems. The JESXCF address space, shown in Figure 5, is created in each system during the first JES initialization beginning with MVS/ESA Version 5.1. With MVS/ESA Version 5.2, the JESXCF address space is created before JES is started. As each JES is initialized, JESXCF joins it as a member into a JES complex specific XCF group.

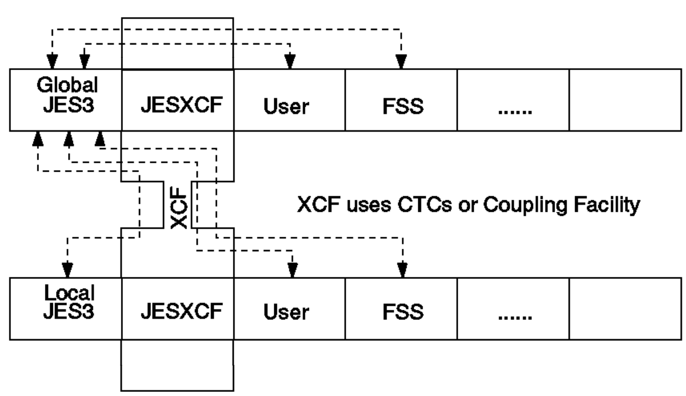


Figure 5. JES3 Configuration in a Sysplex

The JESXCF services are tailored to meet JES specific requirements. The services are available to either JES3 or JES2 and can be used only in JES environments. JES3 Version 5 uses JESXCF services for:

- Joining and leaving the sysplex.
- Staging area transport between the JES3 global and other address spaces (JES3 local, user, FSS) in the JES3 complex.

For a more detailed description, see Appendix C, "JES3 Version 5 and JES Common Coupling Services" on page 107.

2.2.3.3 JESXCF Groups

A JESXCF group is a set of JES3 subsystems that communicate using the JES common coupling services component (JESXCF). JESXCF uses XCF as its communication vehicle. In a JES3 environment, the JESXCF group is a JES3 complex. Within the MVS sysplex, you can define one or more JESXCF groups.

This new programming interface called JESXCF provides the means for JES3 software components to communicate with each other, even though they are on different systems in the sysplex. JES3 uses this interface as a replacement for its JSERV/SSISERV data transport layer.

A JES3 Version 5 complex is required to be contained within one sysplex. The following considerations are necessary for a migration to sysplex:

Sysplex name A name specified for the sysplex in the COUPLExx member of SYS1.PARMLIB.

Group name A group in XCF is the set of related members defined to XCF by a multisystem application. A member of an XCF group resides on one system in the sysplex, and can use XCF services to communicate with other members of the same group. Each group must have a unique XCF group name in the sysplex. With JES3 Version 5, a JES3 global address space, each JES3 local address space, each FSS address space, and the DLOG address space become members of the same XCF group.

member name Each member of the XCF group must have a unique member name. JES3 provides the member name upon attaching to JESXCF.

Note: For JES3 initialization changes for the XCF group name and sysname, see 5.5.1, "Defining the XCF Group Name" on page 70.

2.2.3.4 JES3 JESXCF Considerations

In a JES3 complex, the following members join the JES3 XCF group:

- Global
- Each local
- Each FSS, both CI and WTR
- JES3 DLOG address space

Displaying the JES3 Group and Member Name: The `D XCF,GROUP,WTSCPLX3` command, shown in Figure 6, lists the active members of the JES3 complex. SC50 and SC49 are the JES3 mains, CI5 and CI7 are the CI FSS names, WTRIAZF is the WTR FSS, and JES3DLOG is the JES3 DLOG address space. WTSCPLX3 is the JES3 XCF group name.

```
D XCF,SYSPLEX
IXC334I  17.32.04  DISPLAY XCF 202
   SYSPLEX WTSCPLX3:   SC50            SC49
D XCF,GROUP
IXC331I  15.45.46  DISPLAY XCF
   GROUPS(SIZE):   AOFSMGRP(2)      COFVLFNO(6)      DFHIR000(16)
                   EJESEJES(2)      SYSDAE(11)       SYSGRS(6)
                   SYSIGW00(7)      SYSIGW01(7)      SYSIKJBC(6)
                   SYSMCS(9)        SYSMCS2(5)       SYSRMF(6)
                   SYSWLM(6)        XCFCONS(6)       WTSCPLX3(4)
D XCF,GROUP,WTSCPLX3
IXC332I  15.46.08  DISPLAY XCF
   GROUP WTSCPLX3:  CI5             CI7             JES3DLOG
                    SC49            SC50            WTRIAZF
```

Figure 6. Sample DISPLAY XCF Command Output

JESXCF Address Space Region Size: The JESXCF address space requires storage for the tables associated with each member joining the group. If the configuration consists of one of each of the member types, 6 members as shown in Figure 6, the storage required is 6 X 6 X 64K or 2204K. Ten members would be (10X10X64) or 6400K of storage. For a very large number of FSS address spaces, this is a great deal of storage. So plan your region size very carefully. Be aware of any exit that limits region sizes.

Sysplex Couple Data Set: The sysplex couple data set defines the maximum number of groups and members allowed in the sysplex. Make sure that this number includes the possibility of many FSS address spaces.

2.2.4 Defining CTC Communication Paths

JES3 shares the CTCs with all other components that use XCF signaling for communication between members of the sysplex. This eliminates dedicating CTCs for JES3 global/local communication.

Default XCF signaling paths may be defined for CTCs and the coupling facility. When XCF signaling is through a coupling facility, all systems are connected to the same coupling facility where the signaling structure is defined. XCF signaling through a coupling facility alleviates the work involved in defining the XCF configuration. See A.2, “COUPLExx Parmlib Member” on page 90.

Full Connectivity: XCF signaling requires full connectivity between the systems in a multisystem sysplex. An ESCON CTC signaling path consists of at least two pairs of CTC devices, one inbound and one outbound. The inbound signaling CTC device is initially defined with the PATHIN DEVICE parameter on the COUPLExx parmlib member, and the outbound CTC device is defined with the PATHOUT DEVICE parameter. The number of XCF CTCs required for signaling can be computed using the following formula:

```
Number of CTC channels = (number of systems) * (number of systems - 1 )
```

MVS/ESA SP 5.1 supports 32 systems in a sysplex, and this maximum configuration would require 992 XCF CTCs without the use of an ESCON director and without backup paths.

2.2.4.1 CTC Connections

Figure 7 shows the minimum number of CTCs required for four MVS images to be 12 (6 CTC channels and 6 CNC channels).

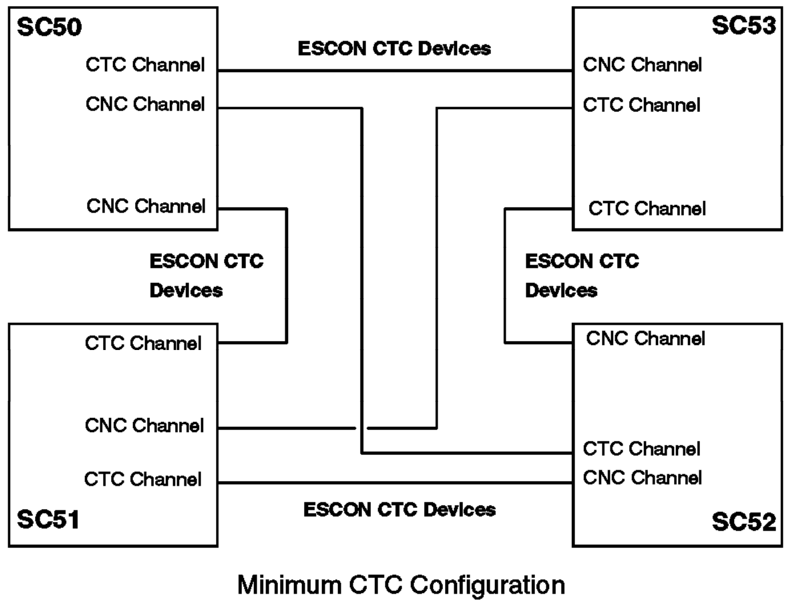


Figure 7. Minimum CTC Configuration for Four MVS Images

A CTC connection requires an ESCON CTC channel at one end of the connection and an ESCON CNC channel at the other end of the connection. A CTC connection supports bidirectional CTC communications. A CTC channel and a CNC channel in a CTC connection can be used to send information from one MVS image to another. The ESCON channels can be either CTC or CNC channels as follows:

CNC channel An ESCON CNC channel is an ESCON channel that can communicate with any ESCON control unit.

CTC channel An ESCON CTC channel is an ESCON channel, and under the covers is a control unit that can only communicate with a CNC channel. The ESCON CTC implements CTC control units and devices within the channel and supports up to 120 two-sided (one side on the CTC channel, the other side on the CNC channel) CTC control units and 512 two-sided CTC devices. A maximum of 256 ESCON CTC devices can be configured to a CTC control unit.

Both the ESCON CTC channel and ESCON CNC channel use the same ESCON channel hardware, but different Licensed Internal Code (LIC). The CTC channel acts like a control unit, not a channel on the ESCON I/O interface to the connected CNC channels.

Connection Rules: The following general rules apply to ESCON CTC connections:

- The CTC channel can be connected to a CNC channel point-to-point or by a static connection using an ESCON Director. It can be connected to multiple CNC channels by dynamic connections using an ESCON Director.

- A CTC channel cannot form a CTC connection with another CTC channel.
- A CNC channel cannot form a CTC connection with another CNC channel.
- The CTC channel is dedicated exclusively to CTC operations. It can be connected only to CNC channels.
- The CNC channel in a CTC connection can be configured to access other CTC channels and other ESCON I/O control units attached to the same ESCON Director.

Note: We recommend that you use a 4-digit device address method for defining the CTC signaling paths as described in *MVS/ESA SP Version 5 Sysplex Migration Guide* and in Chapter 5 of *HCD and Dynamic I/O Reconfiguration Primer*.

2.2.4.2 Full Redundancy

Full redundancy would require doubling the number of CTCs to 24. This option should be considered for not only backup considerations, but for defining a symmetric configuration. The symmetric configuration allows the configuration definitions to be shared across all systems in the sysplex.

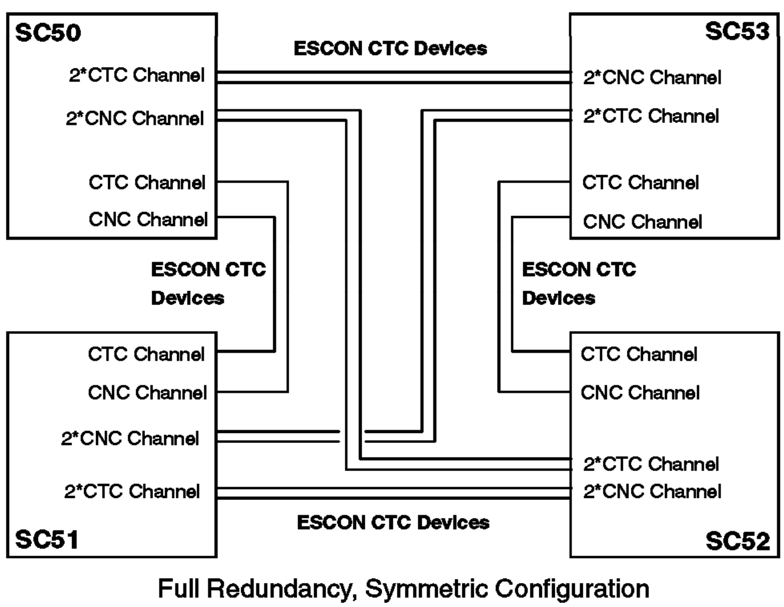


Figure 8. Full Redundancy and Symmetric CTC Configuration

2.2.4.3 Using ESCON Directors

An ESCON Director provides dynamic connectivity between MVS images. The ESCON Director is a switching device that attaches to processors with ESCON channels and to ESCON control units. Figure 9 shows the use of the ESCON Director and reduces the number of CTCs for the configurations shown in Figure 7 on page 21 from 12 to 6, and in Figure 8 from 24 to 8.

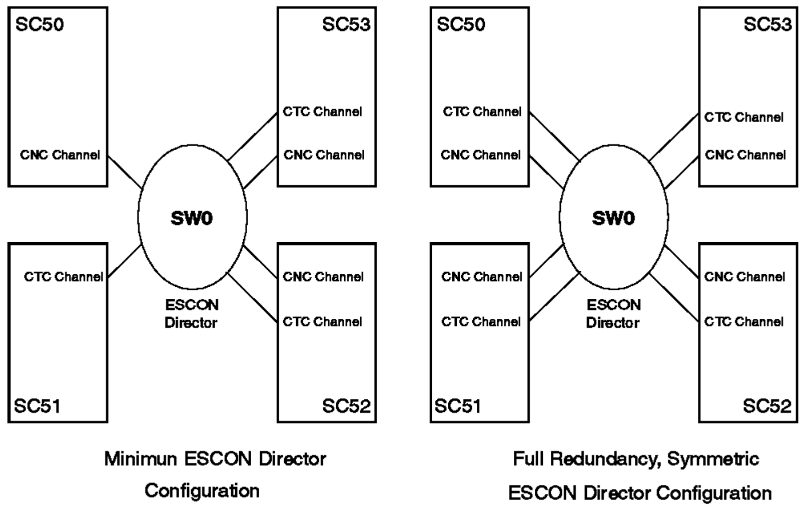


Figure 9. Using ESCON Directors to Minimize CTCs

The connection rule that a CNC channel in a CTC connection can be configured to access other CTC channels and other ESCON I/O control units attached to the same ESCON Director allows the *same* CNC channel at SC50 in Figure 9 to access the CTC channel at SC53, SC52, and SC51. This reduces the total number of channels required.

2.2.4.4 ESCON Director Redundancy

With the use of a second ESCON Director to avoid a single point of failure, Figure 10 and Figure 11 on page 24 indicate the number of CTCs required.

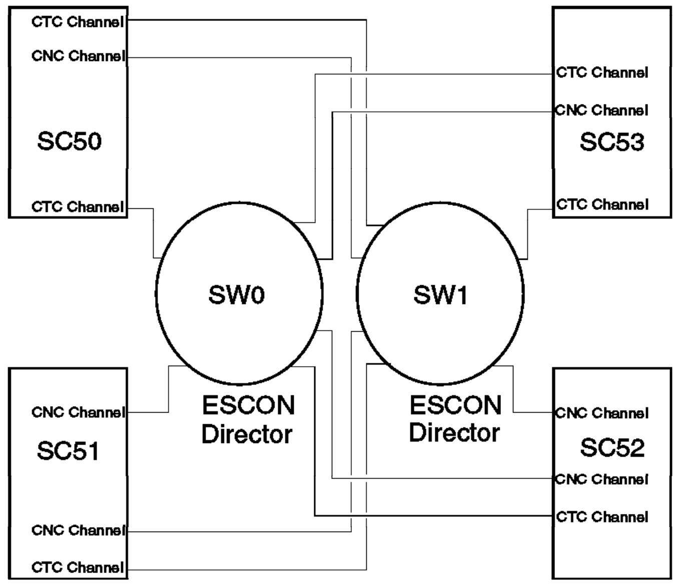


Figure 10. Minimum Redundant Configuration Using Two ESCON Directors

These figures show the differences in the number of CTCs needed for redundant and symmetric configurations using ESCON Directors.

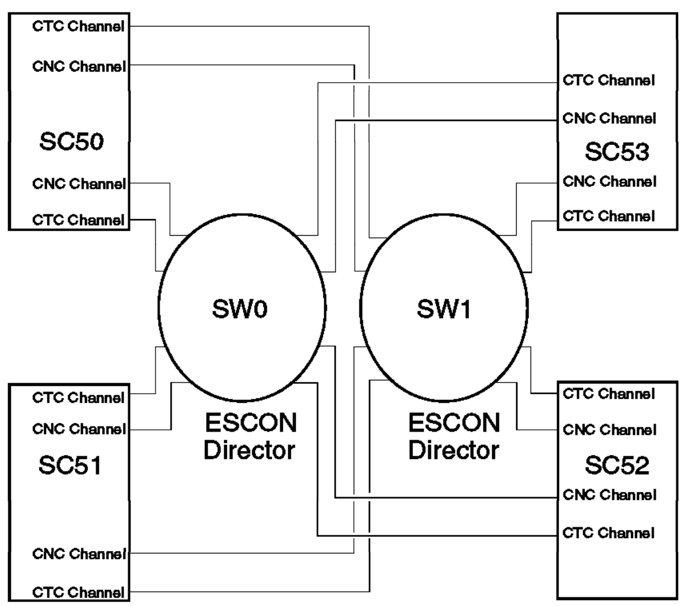


Figure 11. Redundant Symmetric Configuration Using Two ESCON Directors

2.2.4.5 JES3 Initialization CTC Addresses

JES3 CTC addresses are no longer defined in the initialization stream. The changes to define the DEVICE statements for the main processors are discussed in 5.5.2, "CTC Replacement Definitions" on page 71.

2.2.5 Defining Coupling Facility Communication Paths

To use the coupling facility in place of XCF CTCs for JES3 global to local communication, see Appendix D, "Coupling Facility for XCF Communication" on page 125.

2.3 JES3 Sysplex Configurations

You must specify the type of sysplex configuration using the PLEXCFG parameter in the IEASYSxx member, as shown in Figure 12, Figure 13 on page 26, and Figure 18 on page 31. See A.1, "XCF Modes of Operation" on page 88 for a description of the sysplex modes of operation.

You must also specify a COUPLExx parmlib member to identify the sysplex couple data sets and the signaling paths between systems using the PATHIN and PATHOUT statements, as shown in A.2, "COUPLExx Parmlib Member" on page 90, when running in PLEXCFG=MULTISYSTEM mode.

2.3.1 Global-Only JES3 Configuration

With a single MVS/JES3 complex, you can define the XCF configuration as PLEXCFG=XCFLOCAL or PLEXCFG=MONOPLEX, as shown in Figure 12.

Note: This configuration is not the primary focus of this redbook.

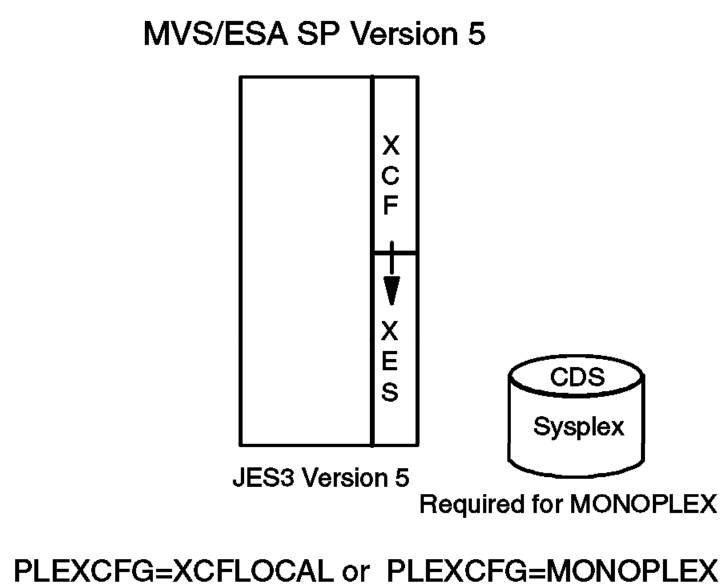


Figure 12. JES3 Global Only Configuration

Note: A Sysplex Timer is not required for any of the XCF modes specified in this configuration.

Consider using PLEXCFG=MONOPLEX in a single system environment because MVS/ESA Version 5 functions you may want to use require sysplex couple data sets. The advantage of specifying MONOPLEX mode is that it allows the MVS/ESA SP Version 5 workload manager (WLM) to be used in goal mode. WLM goal mode requires a function couple data set, and therefore, MONOPLEX mode is the minimum XCF mode of operation. See A.1, "XCF Modes of Operation" on page 88. In MVS/ESA Version 5.2, the automatic restart management function also requires a couple data set. See 3.1, "Automatic Restart Management" on page 41, for JES3 support of this function.

Note: It is possible to have a coupling facility in the single system environment, which also requires couple data sets making it necessary to use PLEXCFG=MONOPLEX mode.

2.3.2 Global-Local JES3 PR/SM Configuration

With multiple MVS images in a single CPC environment using PR/SM, PLEXCFG=MULTISYSTEM is the XCF mode required. A Sysplex Timer is not required as PR/SM provides a common time reference, as shown in Figure 13. The JES3 systems communicate using XCF services, and the signaling services can be through either CTCs or the coupling facility.

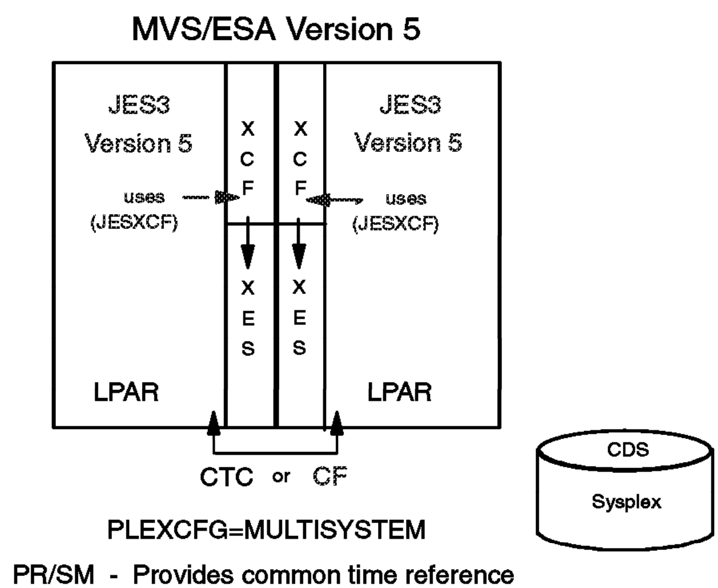


Figure 13. Global-Local JES3 PR/SM Configuration

2.3.2.1 Defining ESCON CTCs between LPARs

Figure 14 shows an example of a two-partition CTC configuration. A processor is defined with two partitions and one ESCON CTC connection between the partitions.

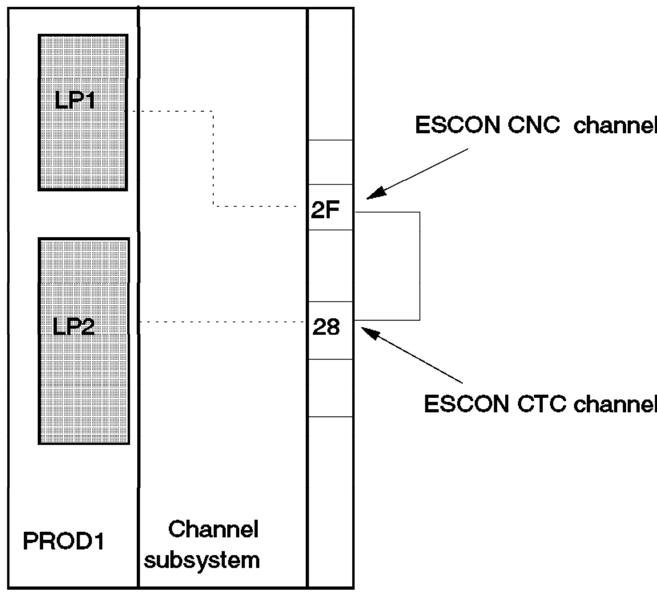


Figure 14. ESCON CTC Connection between LPARs

The IOCP statements to define this ESCON CTC configuration are:

1. Host PROD1

```
CHPID PATH=28,TYPE=CTC,PART=(LP2)
CHPID PATH=2F,TYPE=CNC,PART=(LP1)
```

2. This definition allows LP1 to communicate with LP2:

```
CNTLUNIT CUNUMBR=6020,PATH=(2F),UNIT=SCTC,UNITADD=(00,8)
IODEVICE ADDRESS=(6020,8),CUNUMBR=(6020),UNIT=SCTC,UNITADD=00
```

3. This definition allows LP2 to communicate with LP1:

```
CNTLUNIT CUNUMBR=6010,PATH=(28),UNIT=SCTC,UNITADD=(00,8)
IODEVICE ADDRESS=(6010,8),CUNUMBR=(6010),UNIT=SCTC,UNITADD=00
```

These definitions provide eight SCTC devices. The UNITADD field in the IODEVICE statement provides the linkage between the devices defined on CHPID 28 and those defined on CHPID 2F. That is, device number 6020 with UNITADD 00 on CHPID 2F is used by LP1 to communicate with device number 6010 with UNITADD 00 on CHPID 28 (used by LP2). So, 6010 is connected with 6020, 6011 is connected with 6021, and so on.

LP1 LP1 uses the 602x device numbers to communicate with LP2. You should think of the 602x devices as being the destination in LP2.

LP2 LP2 uses the 601x device numbers to communicate with LP1. You should think of the 601x devices as being the destination in LP1.

Note: The addresses on the two ends of each SCTC device do not match. This is intentional. Matching device numbers mean duplicate device numbers, and we try to avoid having duplicate device numbers.

2.3.2.2 Defining an ESCON CTC in an EMIF Environment

The configuration in Figure 15 shows a processor in LPAR mode with three partitions named LP1, LP2 and LP3. EMIF is used to share channels 29 and 2C among the three partitions. Each partition connects to the other two using this single CTC/CNC pair.

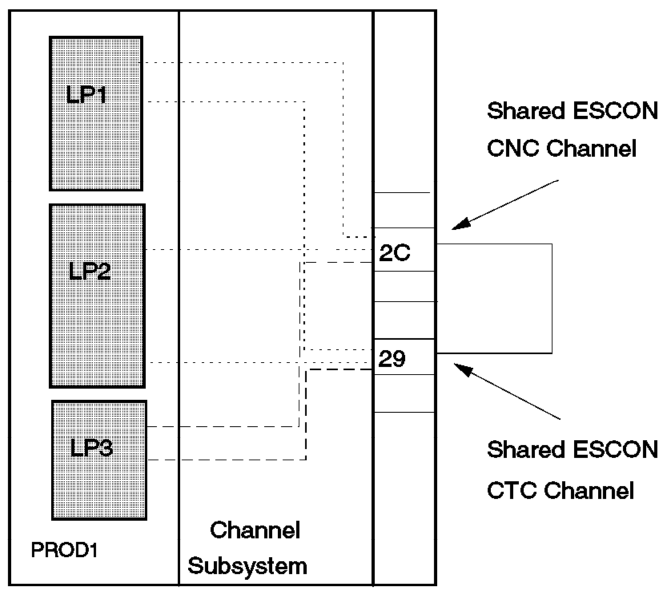


Figure 15. ESCON CTC Connection between LPARs Using EMIF

Figure 16 shows the IOCP statements that correspond to this configuration.

LP1 is in hardware partition number 1, LP2 is in hardware partition number 2, and LP3 is in hardware partition number 3. This is defined in the RESOURCE statement.

We now construct a numbering scheme for control units and devices. We begin by assigning an image ID to each MVS image. This image ID is from X′00′ through X′FF′. In this example:

- Image ID 01 represents partition LP1.
- Image ID 02 represents partition LP2.
- Image ID 03 represents partition LP3.

The control unit numbers are of the form 6xxy, where xx is the image ID, and y is a zero for a first control unit, and a one for a second control unit. Control unit 6010 (01 is the image ID) is used by all other images for communication with LP1. Control unit 6020 (02 is the image ID) is used by all other images for communication with LP2. Control units 6030 and 6031 (03 is the image ID) are used by other images for communication with LP3.

LP1 LP1 has a single control unit (6010) on the CTC channel path. This is connected to control units on the CNC channel path in LP2 and LP3.

LP2 LP2 has a single control unit (6020) on the CNC channel path. This is connected to control units on the CTC channel path in LP1 and LP3.

LP3 LP3 (and additional LPARs if they existed) requires two control units: one on the CTC channel path (6030), and one on the CNC channel path (6031).

The IOCP statements for defining this shared ESCON CTC configuration are:

Host PROD1:

```
RESOURCE PART=((LP1,1),(LP2,2),(LP3,3))
CHPID PATH=29,TYPE=CTC,SHARED,PART=((LP1,LP2,LP3),(=))
CHPID PATH=2C,TYPE=CNC,SHARED,PART=((LP1,LP2,LP3),(=))
```

- The following definition allows all other partitions to communicate with LP1:

```
CNTLUNIT CUNUMBR=6010,PATH=(29),UNIT=SCTC,CUADD=1,UNITADD=((00,8))
IODEVICE ADDRESS=(6010,8),CUNUMBR=(6010),UNIT=SCTC,UNITADD=00,          X
              PART=(LP2,LP3)
```

- The following definition allows all other partitions to communicate with LP2:

```
CNTLUNIT CUNUMBR=6020,PATH=(2C),UNIT=SCTC,CUADD=2,UNITADD=((00,8))
IODEVICE ADDRESS=(6020,8),CUNUMBR=(6020),UNIT=SCTC,UNITADD=00,          X
              PART=(LP1,LP3)
```

- The following definitions allow all other partitions to communicate with LP3:

```
CNTLUNIT CUNUMBR=6030,PATH=(2C),UNIT=SCTC,CUADD=3,UNITADD=((00,8))
IODEVICE ADDRESS=(6030,8),CUNUMBR=(6030),UNIT=SCTC,UNITADD=00,          X
              PART=(LP1)

CNTLUNIT CUNUMBR=6031,PATH=(29),UNIT=SCTC,CUADD=3,UNITADD=((00,8))
IODEVICE ADDRESS=(6030,8),CUNUMBR=(6031),UNIT=SCTC,UNITADD=00,          X
              PART=(LP2)
```

Figure 16. Shared ESCON CTC IOCP Equivalent Statements

Figure 17 is a logical view of the ESCON CTC definitions. A CTC-CNC connection is required between all three partitions, and at least four control units are required.

LP1 Partitions LP2 and LP3 communicate with LP1 using the CTC channel path (CHPID 29) and control unit 6010. The devices attached to control unit 6010 have LP2 and LP3 in the explicit device candidate list (PART=). This prevents LP1 from communicating with itself. The logical address (CUADD=1) defined for control unit 6010, corresponds to LP1′s partition number (in the RESOURCE statement). This defines LP1 as the destination LPAR.

LP2 Partitions LP1 and LP3 communicate with LP2 using the CNC channel path (CHPID 2C) and control unit 6020. The devices attached to control unit 6020 have LP1 and LP3 in the explicit device candidate list. The logical address (CUADD=2) defined for control unit 6020, corresponds to LP2′s partition number.

LP3 Communication with LP3 is slightly different. LP1 communicates with LP3 using the CNC channel path and control unit 6030. The devices attached to control unit 6030 have only LP1 in the explicit device candidate list. LP2 communicates with LP3 using the CTC channel path and control unit 6031.

Devices attached to control unit 6031 have only LP2 in the explicit device candidate list.

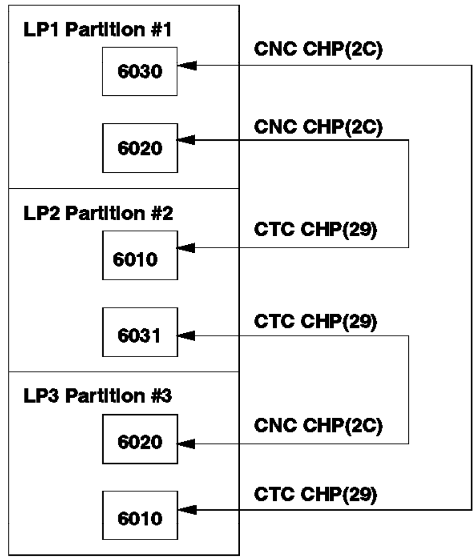


Figure 17. Shared CTC Connection - Logical View

In summary, one control unit with devices is required for each of the first two partitions; one attached to the CTC channel path and one attached to the CNC channel path. Each additional partition requires two control units with devices; one attached to the CTC channel path and the other attached to the CNC channel path.

2.3.3 Global-Local JES3 Multiple CPC Configuration

With multiple MVS images in multiple CPCs, PLEXCFG=MULTISYSTEM is the XCF mode required. A Sysplex Timer is required, as shown in Figure 18. The two JES3 systems communicate using XCF, and the signaling services can be through either CTCs or the coupling facility.

The couple data sets, shown in Figure 18, are briefly described in Appendix A, "Sysplex Terminology" on page 87. The only required one is the sysplex couple data set. The other are only necessary if the function they support is being used. No JES3 functions currently support or need couple data sets. Any or all of these couple data sets can be used in any combination for the JES3 complexes shown in this section of this book.

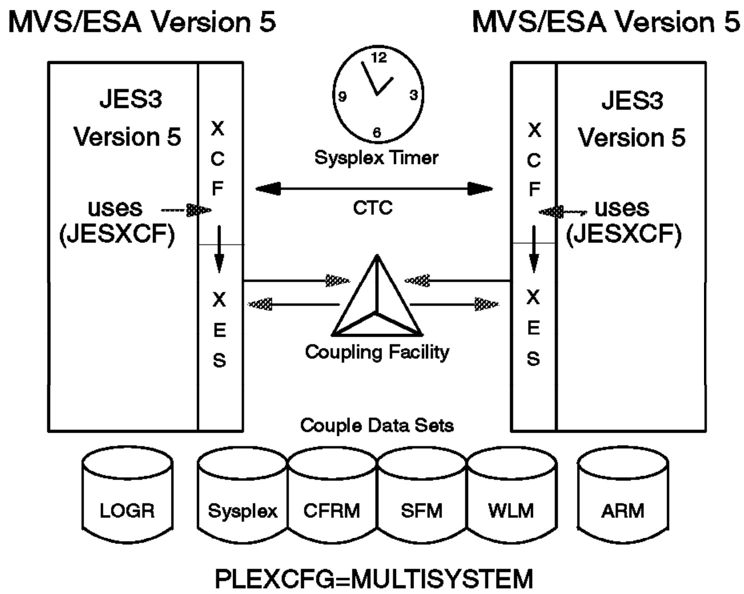


Figure 18. Global-Local JES3 Multiple CPC Configuration

Note: A Sysplex Timer is required in this configuration.

2.4 Consoles Using MCS Multisystem Support

JES3 5.2.1 operations support is significantly enhanced by exploiting the MCS multisystem operations capabilities. The JES3-managed locally attached consoles cannot be used any longer to manage a JES3 complex and therefore their support has been removed. Note that the JES3 unique operations features, such as functional message routing, are preserved in cases where there is no equivalent MCS support.

JES3 continues to provide functional message routing through the following existing interfaces:

- MSGROUTE initialization statement
- Device-related message routing through JES3 destination class specification on the DEVICE initialization statement
- Writer FSS message routing through destination class specification on the FSSDEF initialization statement

2.4.1 MCS Sysplex Operations

Enhancements to MCS support for the sysplex environment began in MVS/ESA Version 4. Figure 19 describes the many enhancements for a sysplex that JES3 Version 5.2.1 utilizes.

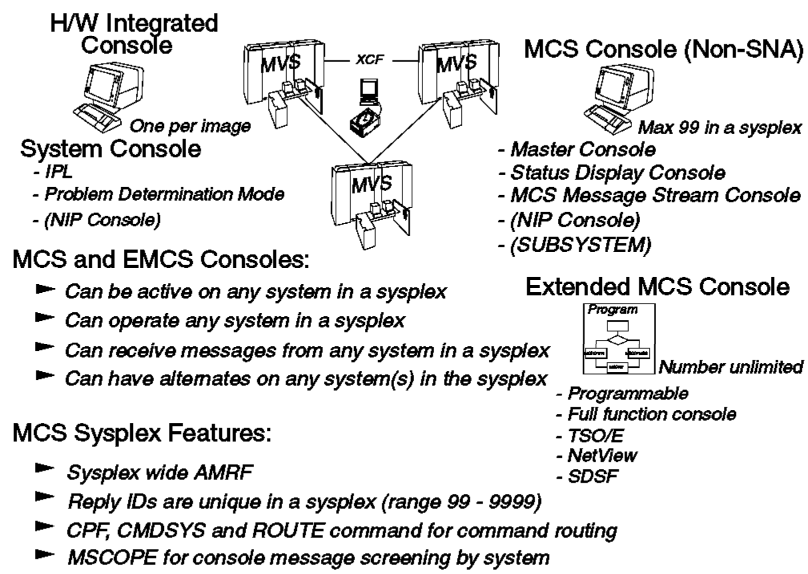


Figure 19. MCS Sysplex Environment

2.4.1.1 Sysplex Consoles

Three types of MCS consoles are shown in Figure 19:

MCS MCS consoles (DIDOCS and SUBSYSTEM); you can define up to 99 through CONSOLxx parmlib member.

EMCS Extended MCS (EMCS) consoles; these are programmable consoles defined and activated through an intended programming interface (the MCSOPER macro with REQUEST=ACTIVATE).

System System console in problem determination mode; this console is automatically defined by MVS during system initialization.

MCS consoles can be active on any system in a sysplex and provide sysplex-wide control. XCF services are used for command and message transportation between systems to provide a single system image for the console environment. MCS multisystem support provides:

- Sysplex-wide action message retention facility (ARMF)
- Sysplex-wide unique reply IDs

For a description of the MCS consoles, see F.1, “Consoles in a Sysplex” on page 137.

2.4.1.2 Command Routing

Command routing is done through:

- ROUTE operator command
- Command prefix facility (CPF)
- CMDSYS setting from the CONSOLE statement in the CONSOLxx parmlib member or the K V operator command

Message presentation is controlled by a console′s ROUTCDE and MSCOPE settings, and the UD (undeliverable message) attribute. SYNCHDEST parameter on the DEFAULT statement in CONSOLxx member defines the alternate console group whose members can receive a synchronous message.

- DIDOCS consoles display messages in either ROLL mode or WRAP mode.
- HOLDMODE specification on the DEFAULT statement of CONSOLxx member controls the temporary suspension or holding screens when in roll, roll-deletable, or wrap mode.

For console switching, consoles may be associated with an alternate console group or an alternate console. When a failing console is switched to an alternate (or when an operator switches a console as a result of the SWITCH command), MCS merges console attributes with those of the alternate. Note that the attributes are added to those of the alternate console and do not replace the existing attributes. Thus, the command authority, message scope, and UD status of the alternate console are not permanently affected by the addition of the failing console′s attributes.

2.4.2 JES3 5.2.1 Sysplex Operations

Installations migrating to JES3 5.2.1 may gain console hardware savings due to the removal of the dedicated JES3-managed consoles. At a minimum, they may observe better console hardware availability and utilization through the sysplex-wide control from any MCS managed console, including the system consoles. Also, the system programmers′ task is simplified as they do not have to maintain multiple separate console environments for each system in the sysplex.

Figure 20 describes the new JES3 5.2.1 environment with MCS multisystem support as follows:

2.4.2.1 MCS and EMCS Consoles

MCS sysplex is enabled and provides sysplex-wide control from any console on any system in a sysplex. Command and message transportation between systems is controlled by MCS. As shown in Figure 20 on page 34, all console types (system console, master console, MCS and EMCS consoles) can send MVS and JES3 commands to any system and receive messages from any system in a sysplex.

2.4.2.2 JES3 Supported MCS Sysplex Features

The JES3 message retention facility (JMRF) is removed. MCS maintains a sysplex-wide view of retained messages through its action message retention facility (AMRF).

The MCS command prefixing facility (CPF) is enabled sysplex-wide in a JES3 5.2.1 complex. By using the appropriate prefix, commands can be routed to all subsystems that exploit this facility. JES3 registers command synonyms with CPF such that JES3 global commands can be entered from anywhere in the sysplex and they are routed to the global for processing. CPF is also used to register local prefixes for JES3 commands that must execute on a local processor, such as RETURN.

Subsystems and applications can exploit without restrictions all MCS facilities (for example, CART and 4-byte console ID). The JES3 command processors support 4-byte console IDs and the MCS command-and-response token (CART).

The *INQUIRY and *MODIFY command processors identify command responses by using the WTO command response descriptor code. Some JES3 inquiry command responses are changed to include a summary message, so that the end of the command response can be easily identified by automation applications. These commands are:*I A, *I G, *I P, and *I Q.

2.4.2.3 Extended MCS Consoles

JES3 exploits MCS facilities. Extended MCS consoles are used to implement JES3 RJP and NJE console support and the JES3 DLOG. MCS becomes responsible for queuing messages to all remote consoles.

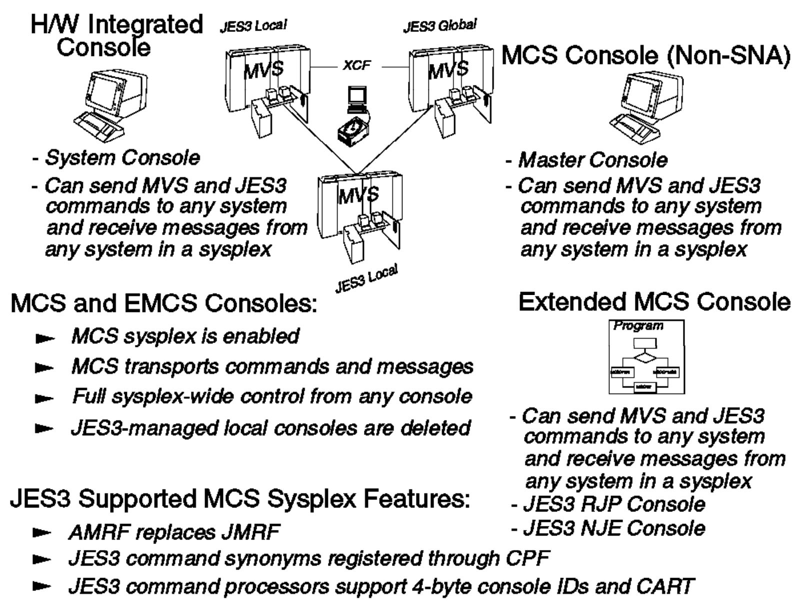


Figure 20. JES3 5.2.1 and MCS Multisystem Support

2.4.3 JES3 5.2.1 Operational Changes

The JES3 5.2.1 sysplex operations enhancements allow customers to realize the benefits of other MVS functions and products that depend on an MCS sysplex environment. For example, these functions and products include:

- MVS operations log
- Display RESERVE message
- ROUTE command
- DB2 use of sysplex command prefixes
- RACF use of sysplex command prefixes
- HCD sysplex ACTIVATE

The following is a list of the major functional changes introduced in the JES3 operations enhancements:

- JES3-only locally attached operator consoles are no longer supported.
- JES3 no longer requires the suppression of the MCS multisystem processing, but instead, relies on the MCS to transport command and message traffic between systems in a sysplex. Operators can control the sysplex operations

from any console regardless of where it is attached. This includes extended MCS consoles such as TSO consoles, or Netview consoles.

Because MCS takes over the command and message transportation, JES3 is no longer involved with the transportation of messages for display and does not reissue WTOs from local processors on the global to display the messages on MCS consoles attached to the global.

- JES3 continues to log messages in a job′s JESMSGLG data set.
- JES3 5.2.1 IATXMLWO and MESSAGE macro services allow JES3 functions to issue true multi-line WTOs. The IATXMLWO macro creates one line of a multi-line message. Each line of a multi-line message is stored in its own copy (IATYMLWO token). These lines (tokens) are chained together and sent to the MESSAGE macro to issue the multi-line WTO message. A multi-line WTO message is limited to a maximum of 999 lines.
- The JES3 SVC 34 SSI interface module has been changed to allow JES3 commands to be up to 126 characters (instead of 80 characters). In JES3 5.2.1, output service command processing (*I U and *F U) and the *CALL JMF command accept 126 character input. Other command processors are still restricted to the maximum of 80 character command length. In addition, commands may be stacked (multiple commands on a single line) up to 126 characters, provided each command in the stack adheres to its length restriction.

Note: All installation programs that are using the JSERV or SSISERV macros services to enter JES3 commands should be changed to use extended MCS consoles and MGCRE macro service.

2.4.4 JES3 5.2.1 and Command Routing

The MVS command prefix facility (CPF) allows a subsystem (like JES3 or JES2) to register system- or sysplex-wide command prefixes. An operator command that is entered with a registered CPF prefix is recognized by MCS and routed to the system where the prefix is defined.

In general, commands routed explicitly with the MVS ROUTE command and with sysplex-wide prefixes are rerouted by MCS as required according to the sysplex-wide prefix definition after the ROUTE command routing completes. If a command is routed using a sysplex-wide prefix, MCS routes it only *once*, even if the command has multiple sysplex-wide prefixes.

Note: Double prefixes are allowed for commands. MCS uses the first prefix to transport the command. Once the command is transported, MCS then presents the command with the remaining prefix to the command processors on that system.

The MCS CPF macro service allows you to:

- Define a command prefix by specifying the command prefix. It may specify a name that identifies the subsystem owning the command prefix, the scope (SYSPLEX or SYSTEM), the failure disposition, and whether the command prefix is to be removed from the command text prior to the command being executed on the receiving system.
- Delete an existing command prefix.
- Redefine an existing command prefix for a system or owner name.

CPF processing is enabled sysplex-wide in a JES3 5.2.1 complex. The CPF scope is no longer limited to a single system as it is in a sysplex running JES3 releases prior to JES3 5.2.1. See F.3, "Command Routing in a Sysplex" on page 141 for command routing considerations.

JES3 5.2.1 registers the SYSPLEX (PLEXSYN=) and SYSTEM (SYN=) scope command prefixes that are defined on the JES3 CONSTD initialization statement, as shown in Figure 21.

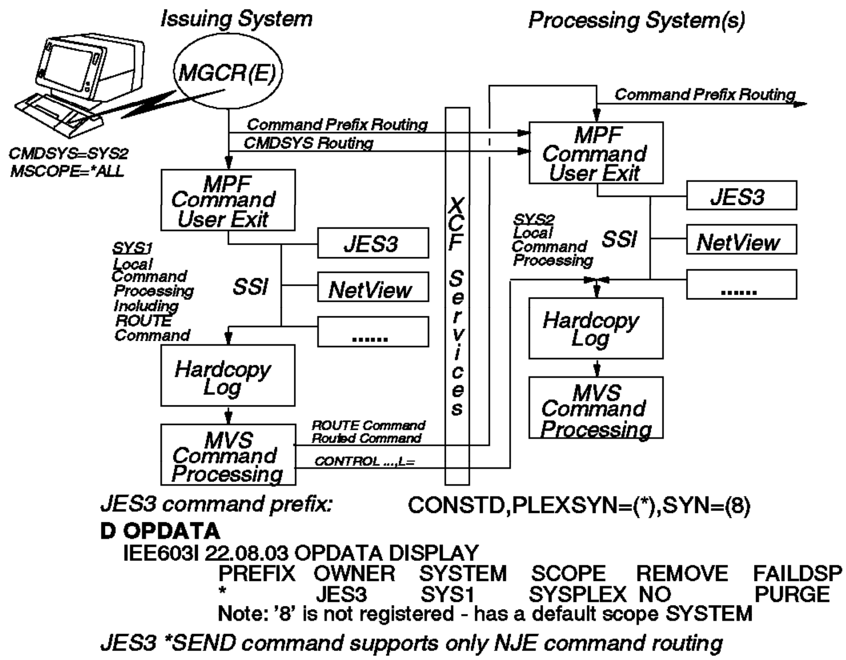


Figure 21. Command Prefix Definition and Command Routing

JES3 5.2.1 registers with CPF sysplex-wide command prefixes that are defined with the PLEXSYN= parameter on the JES3 CONSTD initialization statement. Up to six sysplex-wide JES3 prefixes (also known as synonyms) can be specified. The sysplex-wide prefixes are always registered with CPF on the global processor and are re-registered during the dynamic system interchange.

Note: To avoid conflicts with short-form WTOR replies, the JES3 default SYSTEM scope prefix (8) is not registered with CPF and therefore is not displayed by the operator command *D OPDATA*.

If JES3 fails to register the "*" prefix, there will *not* be any JES3 sysplex-wide prefixes, and all JES3 commands from locals must be explicitly routed to the global using either the MVS ROUTE command or double command prefixes. The first of the double prefixes must be defined as "REMOVE=YES," to route the command to the global. The second prefix should be the JES3 "*" or any of the SYSTEM-scope prefixes registered by the global.

Note: Starting with JES3 5.2.1, the JES3 *SEND command can be used only for sending operator commands to other NJE nodes. Within a sysplex, the MVS ROUTE command or command prefex routing must be used to send command between systems in the sysplex.

For additional JES3 5.2.1 console considerations, see 5.9, “JES3 5.2.1 Enhanced Console Processing User Considerations” on page 81.

2.5 MVS System Logger

The MVS system logger is a set of services that allow an application to write, browse, and delete log data. The system logger is introduced with MVS/ESA SP 5.2. You can use system logger services to merge data from multiple instances of an application, including merging data from different systems across a sysplex. The log data is in a log stream that resides on a coupling facility; as the coupling facility fills, older data might be offloaded to archival log stream data sets on DASD, as shown in Figure 22.

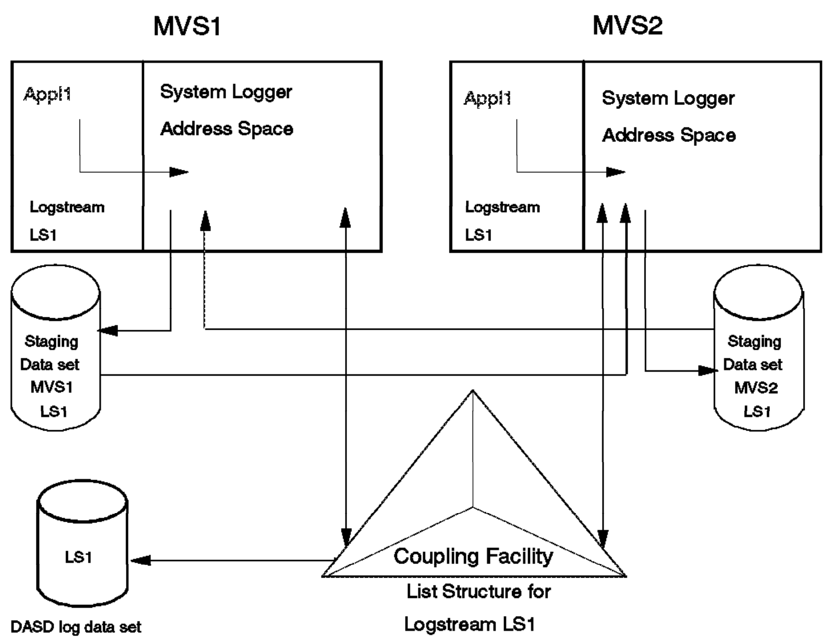


Figure 22. MVS System Logger Overview

A log stream is a collection of one or more log records (also referred to as log blocks) written by an application using services provided by the MVS system logger. The application using MVS system logger services may or may not have multiple instances of itself executing in a sysplex. In the case of an application where each instance of the application writes log blocks to the same log stream, the result is a sysplex-wide merged log stream.

The MVS system logger executes in its own MVS address space and provides a set of MVS system services that allow an application to:

- Connect to a log stream
- Write data to a log stream
- Browse data from a log stream
- Delete data from a log stream
- Disconnect from a log stream
- Maintain an inventory of log streams and their associated characteristics

The coupling facility/DASD configuration log stream consists of a structure resident in a coupling facility and one or more DASD data sets. When a log block is written to a coupling facility/DASD configuration log stream, the log data is buffered in processor-related storage and written to the coupling facility list structure. Once the log data is written to a local buffer and a coupling facility list structure, the writer is informed that the write request is complete.

As log data ages in the coupling facility, it is eventually migrated to a DASD data set so that coupling facility storage can be reclaimed to support incoming write requests. The migration process does not interfere with incoming write requests. Once data is successfully written to DASD, the log blocks are eligible for deletion from MVS system logger local buffers and the coupling facility structure.

When a DASD data set eventually becomes full, another data set is allocated and subsequent writes go to the new data set.

2.5.1 JES3 5.2.1 and the MVS Operations Log

The MVS operations log (OPERLOG) function of the MVS console services provides a sysplex-wide merged and chronologically ordered message log by using the MVS system logger services. The messages are logged in the form of *message data blocks* (MDB). An MDB is a structured control block that contains a complete representation of a message, including both text and control information.

Note: JES3 MLOG (hardcopy printer) support is removed in JES3 5.2.1. Also, JES3 exit IATUX31 (used to examine and modify destinations or message text) is removed and thus no longer affects logging decisions. See 5.8.2, “User Exit 69” on page 78 and 5.8.3, “User Exit 70” on page 79.

The MVS hardcopy message set can be logged into the OPERLOG (in an MDB forma)t, SYSLOG (MVS record format), or both.

2.5.2 JES3 5.2.1 DLOG

JES3 5.2.1 provides an alternate for SYSLOG as a migration accommodation (DLOG) for customers who are unable to activate the MVS OPERLOG across the JES3 complex. The JES3 DLOG centrally records the hardcopy message set for systems in a JES3 complex in the JES3 format. The JES3 DLOG is written to SYSLOG on the global processor. See also 2.5.2, “JES3 5.2.1 DLOG.”

The JES3 DLOG message traffic is managed by MCS. On the global, JES3 activates an extended MCS console with the HARDCOPY=YES attribute to receive the hardcopy message set, as shown in Figure 23. The messages received by the DLOG EMCS console are in the MDB format and are converted to the JES3 DLOG format and then written using the WTL macro service to the global JES3 system′s SYSLOG (alias DLOG).

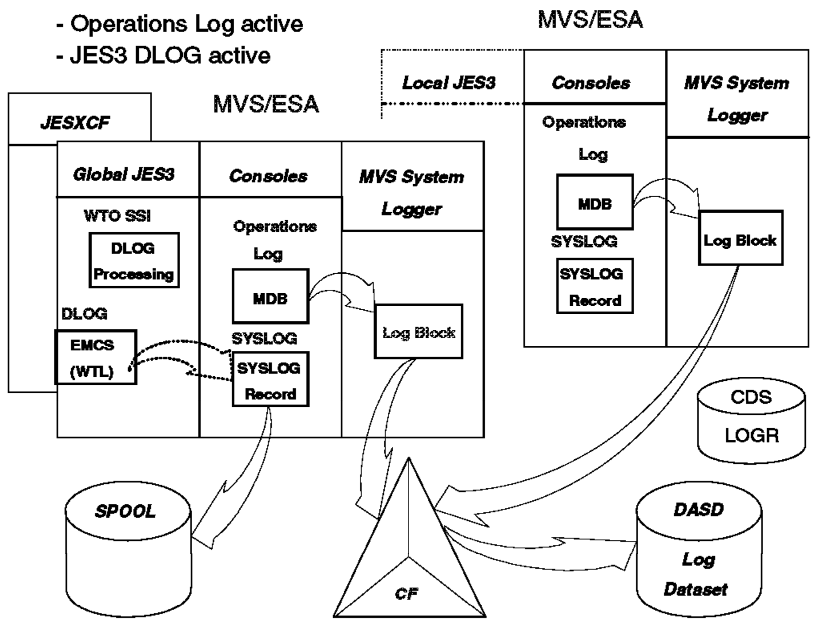


Figure 23. JES3 5.2.1 and MVS Operation Log

The JES3 WTO subsystem interface routines receive control as part of each WTO or WTL macro processing. When the DLOG is active, the JES3 WTO SSI processing suppresses the MCS logging of the hardcopy message set into the SYSLOG. Note that JES3 never suppresses the MCS logging into the OPERLOG log stream.

2.5.2.1 JES3 5.2.1 NJE Console

JES3 5.2.1 creates internally an extended MCS console for NJE console processing. NJE consoles provide a remote node with the capability to inquire on and control work that has arrived from the NJE network. There are no physical JES3 NJE consoles, rather the NJE console support provides a way for performing command association between a requestor on a remote node and console operations on the JES3 node. Internally JES3 uses CARTs to associate a command and its response with a given NJE node.

2.5.2.2 JES3 5.2.1 RJP Consoles

JES3 5.2.1 internally creates an extended MCS console for RJP console processing. This console is used to deliver messages to *all* RJP work stations. JES3 messages issued in response to commands entered from RJP consoles are routed to the RJP EMCS console. JES3 retrieves these messages and delivers them to the remote console. Internally JES3 uses CARTs to associate a command and its response with a given RJP work station.

Chapter 3. JES3 Support of MVS Sysplex Functions

The following new functions have been added to MVS with MVS/ESA Version 5:

- Automatic Restart Management (ARM)
- Shared tape support versus JES3 tape management
- System symbolics (cloning)

3.1 Automatic Restart Management

The MVS/ESA SP 5.2 automatic restart management (ARM) provides fast, efficient restarts for exploiting applications in case of a system or application failure. ARM automatically restarts registered batch jobs or started tasks (STC) when they unexpectedly terminate and reduces the time required for the restart. The application outages that may be restarted by ARM may be the result of an ABEND, system failure, an operator CANCEL command, or the removal of a system from the sysplex.

In a Parallel Sysplex, the typical exploiters of the ARM services are the functions (for example, CICS, DB2, and VTAM) that are key to the OLTP data sharing.

ARM uses *event driven* failure recognition, such as *end-of-memory* (EOM) and XCF *sysgone* processing, to trigger restart activities. It is part of the integrated sysplex recovery, which includes:

- Sysplex failure management (SFM), which provides automated sysplex recovery for signaling failures, missing status updates and reconfiguring systems in a PR/SM environment.
- Workload manager (WLM), which provides statistics on the remaining processing capacity in the sysplex.

ARM provides an ongoing continuity with automation (AOC/MVS) and production control (OPC/ESA) products by augmenting their function. The design is simple and is intended to reduce the need to manually adjust systems, balance workloads, or provide special recovery instructions relating to abnormal situations. It makes contingency planning far more flexible.

3.1.1 ARM Environment

ARM runs in the cross-system coupling facility (XCF) address space and maintains its own dataspaces. ARM requires a primary couple data set to contain policy information as well as status information for ARM-managed jobs. ARM can be used in JES2, JES3, and master subsystem environments. An ARM registered started task or batch job can be only restarted in the same job initiation environment where it was started.

The automatic restart management couple data set holds policy information that defines how MVS is to manage restarts for specific batch jobs and started tasks that are registered as elements of automatic restart management. An ARM policy identifies elements and restart groups with restart parameter values that differ from the system-provided policy default values. Your installation does not have to specify a policy if the default values are acceptable, or if the applications override the defaults.

Note: The ARM policy values are determined when an element is restarted, not when the batch job or started task registers with automatic restart management.

An ARM policy typically includes:

- Restarted groups which identify related elements that are to be restarted as a group.
- The order in which elements in the same restart group are to become ready after they are restarted.
- The level, name, and type associated with elements that must be restarted in a particular order. The elements are restarted from the lowest level to the highest level.
- Target systems on which elements can be restarted in a cross-system restart.
- Control values for restart pacing, restart attempts, restart timeout, ready timeout, and restart conditions.

3.1.2 Element Registration

Batch jobs or started tasks become eligible for ARM restarts when they register with ARM through the IXCARM authorized macro services. An ARM registration requires the specification of an restart “element” name. This element name must be unique in the sysplex as that is how ARM identifies jobs under its control for restart. The registration request can also override the restart policy definitions stored in the ARM couple data set. Optionally several ARM exits may also be used to customize automatic restart management processing and influence how an element will be restarted.

The three IXCARM ARM services, shown in Figure 24 on page 43, that are used to control ARM registration are:

REGISTER Early in the initialization processing for a job, a program that wants to use ARM for restart must issue the IXCARM REQUEST=REGISTER macro to register.

READY When the job that has issued the register request has completed initialization and is ready to process, it must issue the IXCARM REQUEST=READY request.

DEREGISTER Before the job completes its normal termination processing, the program issues an IXCARM REQUEST=DEREGISTER request to remove ARM restart request. If the job completes normally without deregistering itself, ARM deregisters the job.

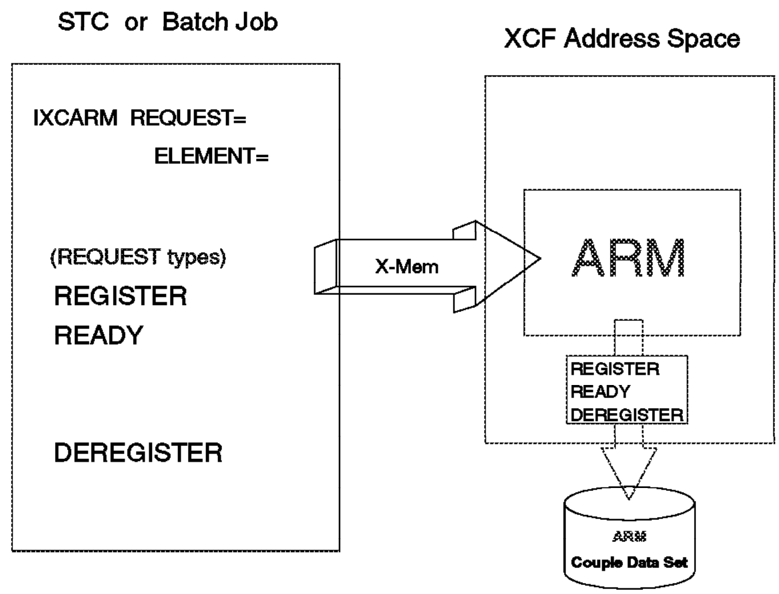


Figure 24. Registering with ARM

3.1.3 ARM Element States

ARM puts each element into one of five states to identify the last interaction that the element had with it. These states, shown in Figure 25 on page 44, are:

Starting From the initial registration (IXCARM-Register) of a program as an ARM element to it subsequently becoming ready (IXCARM-Ready).

Available From the time the element becomes ready (IXCARM-Ready) until the element either deregisters from ARM or terminates.

Failed From when ARM detects the termination of an element, or termination of the system on which the element had been running, until ARM initiates a restart of the element.

Restarting From ARM′s initiation of a restart until the subsequent reregistration of the element.

Recovering From the element′s registration after an ARM restart to its subsequent issuing of IXCARM-Ready.

The state of a given element will be part of the information provided when ARMSTATUS is requested through the DISPLAY XCF command for one or more elements.

When ARM detects that the element has unexpectedly terminated, that is, the program has terminated without invoking the deregister service, or that the system on which the element had been running has left the sysplex, ARM sets the element′s state to *failed* as part of its processing to restart the element. An element will be in the failed state for a very brief interval.

When ARM restarts the element, it sets the state to *restarting*. When a restarting element subsequently invokes the register service, ARM sets the state to *recovering*. Once the element is prepared to accept new work, it notifies ARM by invoking the ready service, which causes ARM to set the element's state to *available*.

Figure 25 summarizes the ARM components and ARM processing for a job.

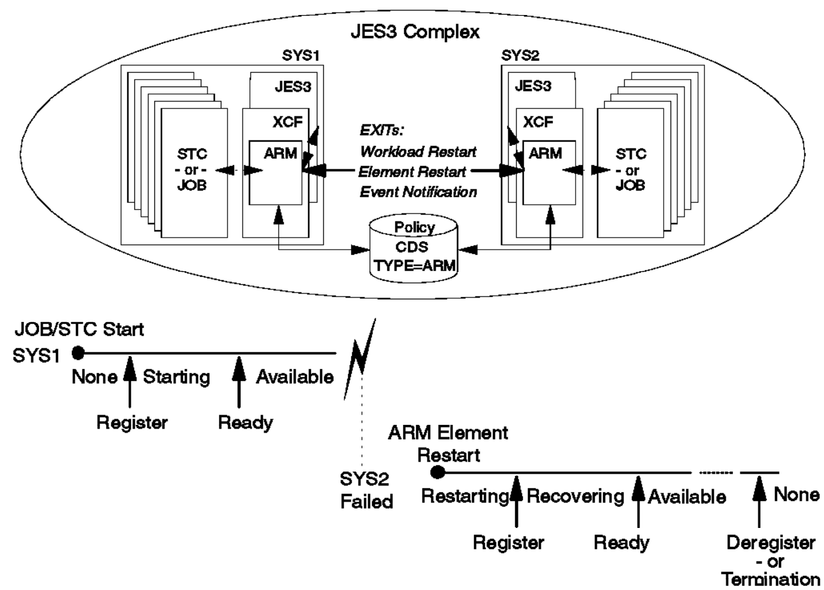


Figure 25. Automatic Restart Management

3.1.4 ARM Interactions with JES3

When a started task or batch job requests to be registered as an ARM element, ARM processing includes the registering of that element with JES3. JES3 returns a job token related to the registering job and the name of the JES XCF group in which that job is running. JES3 also indicates the job as ARM registered.

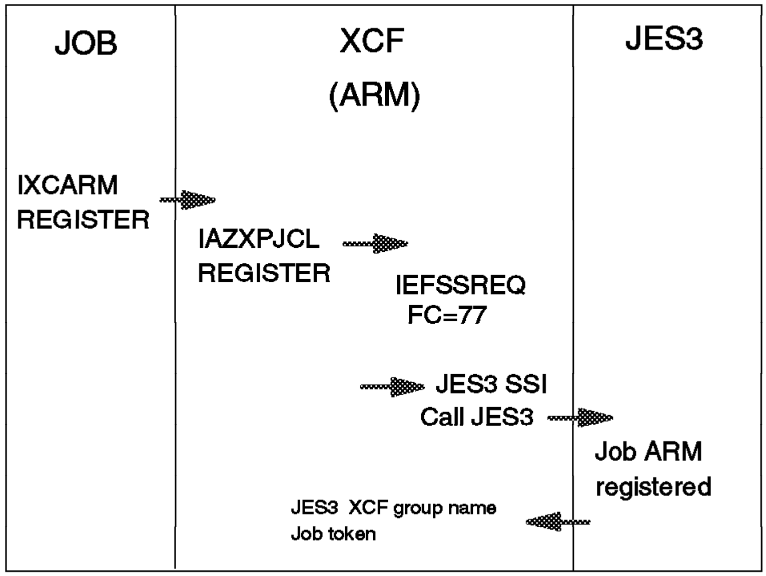


Figure 26. A Job Registering with ARM and JES3

ARM uses this job token and XCF group name as an identifier in all subsequent interactions with JES3 for this element. JES3 also uses the job token and XCF group name as an identifier when it subsequently asks or tells ARM something about an element.

Besides internal changes, JES3 provides a new command extension to the cancel command to support ARM.

```
*F J=jobno|jobname,C|CO|CP,ARMR
*C mainname,jobno|jobname,ARMR
```

The ARMR operand, an abbreviation for ARMRESTART, requests that ARM automatically restart the specified batch job after it is cancelled, if that job is registered with automatic restart manager. The ARMR operand is ignored if the batch job is not executing or it is not registered with ARM.

3.2 MVS Shared Tape Allocation

MVS/ESA SP 5.2 allocation is enhanced to support automatic tape switching, the ability for an installation to have a category of non-JES3 managed tape devices to be simultaneously online for several systems in a Parallel Sysplex. These devices are not dedicated to any one system; rather they can be used by all systems that are connected through a coupling facility in the Parallel Sysplex. The coupling facility is used to maintain the sysplex-wide allocation status of the automatically switchable tape devices. Tape drives eligible to be automatically switchable must have the ASSIGN/UNASSIGN feature. Allocation uses single system assign to control which system at any given time can access the tape device.

The MVS VARY ONLINE command, when it is issued for an automatically switchable device, brings the device online but does not assign it. The automatically switchable device is assigned at step allocation or dynamic allocation time and is unassigned at step termination or dynamic unallocation time. When the MVS VARY OFFLINE command is issued for an automatically switchable device, the device is taken offline from MVS.

The MVS "VARY dddd,AUTOSWITCH,ON or OFF" command allows you to dynamically change the AUTOSWITCH attribute of a tape device without using HCD. This command is accepted only for non-JES3 managed tape devices that are offline.

Note: When a JES3 managed tape device is brought online to a JES3 main, it is also automatically varied online to MVS on that system. If the device is assignable, JES3 uses multi-system assign that allows the device to be accessible from all the systems in the JES3 complex. When a JES3 managed tape device is varied offline from a JES3 main, it is also automatically varied offline from MVS on that system. If the device is assignable, JES3 unassigns the device on that system.

3.2.1 JES3 Coexistence with MVS Shared Tape Allocation

JES3 main device scheduler (MDS) on the global processor reserves resources (data sets, devices, and volumes) for jobs based on the global knowledge of resource status (maintained in the JES3 global address space). The resources are reserved for jobs before they are passed to MVS for execution. When jobs go through the MDS phase, it is not necessarily known on which processor they will be executing. This is one of the reasons why JES3 uses multi-system assign for assignable tape drivers. During job step initialization, MVS allocation calls JES3 to get the device numbers that MDS has reserved for the job. At MVS unallocation time, JES3 is informed about the released resources.

Some considerations when both JES3 managed and automatically switchable tape drives are used:

- JES3 managed devices and automatically switchable devices must form separate non-overlapping device groups.
- JES3 managed devices must be defined in the JES3 initialization stream such that subgeneric splits are avoided.
- The JES3 SETNAME initialization statements must define the UNIT names that will be managed by JES3.
- The automatically switchable devices must have UNIT names other than the ones defined on the JES3 SETNAME initialization statements.
- Input and output tape data sets must be directed to either JES3 managed devices or to the automatically switchable devices by using the proper UNIT definition on the JCL.

  Note: When the system obtains volume serial information from catalogs for a cataloged tape input data set, the retrieved device information must be overridden on JCL if JES3 manages the generic tape device type, in order to direct the allocation to the automatically switchable devices, or vice versa.

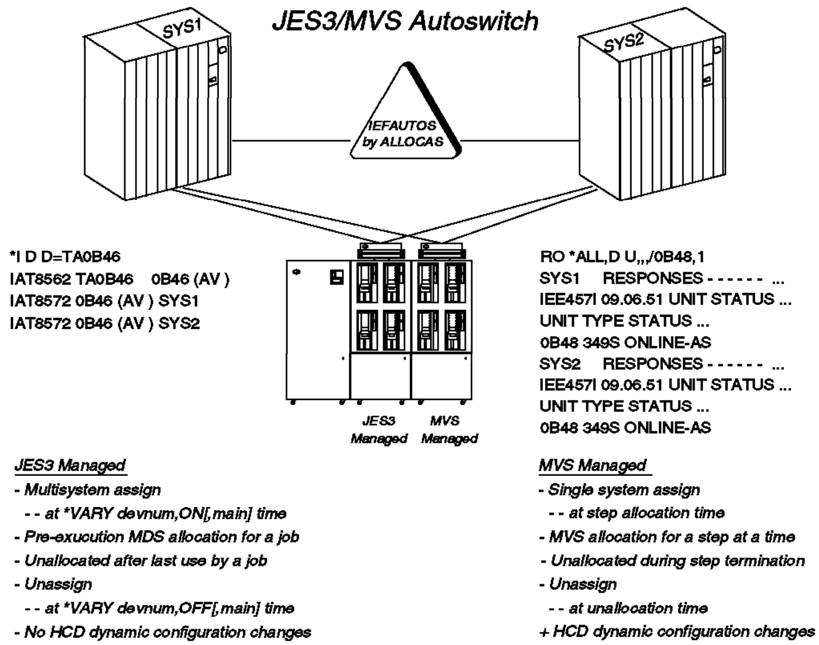


Figure 27. JES3 versus MVS Tape Management

When you configure your Parallel Sysplex, both from a hardware and software perspective, you need to plan for maximum availability and for future changes. As hardware configuration definition (HCD) does not allow dynamic I/O configuration changes for JES3-managed devices, you should define planned devices ahead of time into your hardware configuration and the JES3 initialization stream. At the next planned JES3 warmstart you should activate the new configuration even if the planned devices are not installed. A JES3 warmstart requires a JES3 complex-wide MVS reIPL.

3.3 System Symbols and JES3

The Parallel Sysplex environment is generally replicated with a similar setup on each of the MVS systems. However, despite this similarity, there are a number of unique definitions for each system. For example, each MVS system in the sysplex must have a unique system name. Job names for multisystem applications should be unique to simplify problem determination, particularly in the ARM environment.

To allow installations to use a single parmlib member, a single procedure, or a single command to handle multiple MVS systems in a sysplex, MVS/ESA SP 5.2 provides system symbols substitution, which facilitates replication of definitions.

System symbols act like variables in a program. They can take on different values based on the input to the program. When you specify a system symbol in a parmlib member, the system symbol acts as a “place holder.” Each system that shares the definition replaces the system symbol with a unique value during initialization. Symbolic substitution can be used in:

- SYS1.PARMLIB member parameters
- MVS commands
- JES3 commands

- Started task and TSO logon JCL

System symbols have their names defined to the system. Installations define substitution texts or accepts system defaults for the named symbols. The defined system symbols are:

- &SYSCLONE
- &SYSNAME
- &SYSPLEX

MVS/ESA SP 5.1 provided base support for system symbol variables by defining two system symbol variables (&SYSNAME and &SYSPLEX). MVS/ESA SP 5.2 defines additional symbolic variables:

&SYSCLONE Represents a one- or two-character value that can be set through parmlib member IEASYMxx or it defaults to the last two non-blank characters of system name if no explicit value is assigned. &SYSCLONE is intended to be an abbreviated identifier for a MVS system image. During IPL, the &SYSCLONE value for a system may be checked against the values for the active systems in the sysplex (at the MVS/ESA SP 5.2 level), and if a duplicate &SYSCLONE value already exists on any system, it is rejected.

user symbols Up to 100 system symbol variables that the user can define with a name of his own choosing.

The symbolic variables are defined at IPL time. The term "static system symbols" is also used to refer to the system symbolic variables because they are defined at IPL time and remain fixed for the life of the IPL.

Static system symbols in source JCL are supported for demand select jobs (started tasks and TSO logons). JCL rules are used for determining where the symbols can be used and how they are resolved. Both system symbols and JCL symbolic parameters are used to resolve the symbols in the source JCL for demand select jobs. If the JCL symbolic parameters redefine a system symbolic, the value associated with the JCL symbolic parameter is used. For batch jobs, only the JCL symbolic parameters are used.

Figure 28 shows the principles of how the global JES3 receives the system symbol tables from the local JES3 main processors, and how they are passed to converter/interpreter processing for demand select jobs.

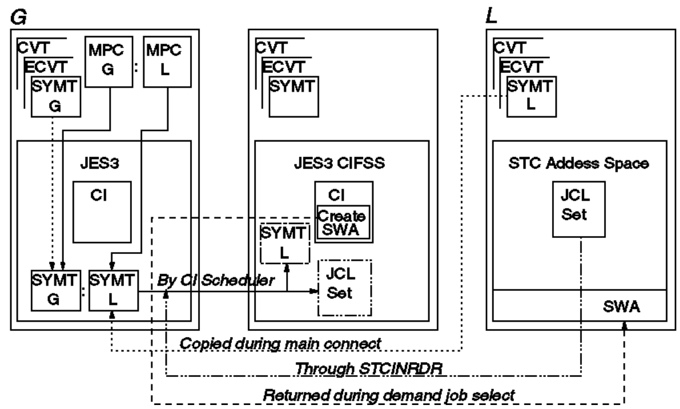


Figure 28. JES3 Support for System Symbols

JES3 Converter/Interpreter (CI) service controls the conversion of JCL statements to its internal representation. The three principal phases of CI service are:

- The *converter/interpreter* phase uses MVS services to convert and interpret (through the MVS interpreter) the JCL into scheduler control blocks (SWA) on the global main or in a CI functional subsystem address space.
- The *prescan* phase creates JES3 job tables from the scheduler control blocks for use in the postscan phase. At the end of this phase, JES3 moves the scheduler control blocks from the SWA to the JES3 spool.
- The *postscan* phase locates data sets and gathers information for use in JES3 device setup.

In the JES3 global address space, the CI dynamic support program (DSP) controls all three of the above phases. More than one copy of the CI DSP can be active at a time.

CI service can also take place in a CI functional subsystem (FSS) address space. A CI FSS address space contains many functions similar to a JES3 address space and can operate on the global or any local main.

In order for the replication of installation definitions (cloning) to work for demand select jobs, JES3 must use the correct system symbols table during the MVS C/I phase, even if the conversion occurs on a system other than where the demand select job is to execute. When the local connects, it passes the connecting system′s system symbols to the global. When a demand select job is scheduled for CI, the symbol table of the system where the job executes is passed with the JCL to the processor that performs the JCL conversion. For ARM restarts, JES3 uses the symbol table that was used for the original execution of the job.

3.3.1 Defining System Symbols

The IEASYMxx parmlib member provides a single place to define system parameters and system symbols for all systems in a sysplex.

IEASYMxx does the following for each system:

- Defines static system symbols
- Specifies the IEASYSxx parmlib members that the system is to use.

- Specifies the system name, which is also the substitution text for the &SYSNAME system symbol.

In the IEASYMxx member, you identify individual systems and, for each system, you can define a unique substitution text for the same system symbols. When the system symbols are specified in shared resource names, each system can substitute unique text for those system symbols.

Figure 29 shows you a sample definition of parmlib member IEASYMxx.

```
SYSDEF          SYMDEF(&CLNLST='AA')            /* CLONING NAME */
                SYSCLONE(&SYSNAME(3:2))
                SYMDEF(&IEASYSP='XX')
SYSDEF          HWNAME(P101)
                LPARNAME(A3)
                SYSPARM(49)
                SYSNAME(SYS1)
                SYMDEF(&PRIJES='JES3')
SYSDEF          HWNAME(P201)
                LPARNAME(A3)
                SYSPARM(50)
                SYSNAME(SYS2)
                SYMDEF(&PRIJES='JES3')
```

Figure 29. SYS1.PARMLIB Member IEASYMxx Example

The current settings and values of the symbols can be displayed using the MVS command `D SYMBOLS`.

Chapter 4. JES3 Continuous Availability Issues

A major design goal for the System/390 Parallel Sysplex is to provide a continuously available environment with scalable parallel servers, building on the robustness of MVS and System/390. Continuous availability is a combination of the hardware and software technology and the environment in which the technology exists. S/390 hardware and software running as a Parallel Sysplex offers a unique continuous availability platform, handling both fault tolerance and system management issues.

This chapter discusses the availability and configuration issues with JES3 in a Parallel Sysplex.

4.1 Availability and JES3-Managed Devices

A major concern or availability issue is adding or deleting DASD or tape devices to the JES3 complex. This requires a complex-wide IPL due to the changes required in the JES3 initialization stream. Currently, the only ways to avoid this warmstart is to:

- Use SETUP=NONE on the STANDARDS initialization statement and have no JES3-managed devices and data sets. This allows adding DASD and tape devices dynamically without IPL of the JES3 complex. This also requires the use of GRS or an equivalent product to provide data set protection in a multisystem complex. See 4.1.2, "Global Resource Serialization (GRS)" on page 52.

  Note: Removing tape management from JES3 control is not a highly recommended option. See 4.2, "JES3 Tape Management" on page 53.

  Another option is to define SETUP=THWS on the STANDARDS statement and allow JES3 to manage tapes and not define the DASD using the DEVICE statement. This allows JES3 management for tape and not for DASD and dynamic I/O reconfiguration can be done for the DASD. GRS or an equivalent product is required for data set integrity.

- Define all DASD as SMS-managed and remove all DEVICE statements for the DASD from the initialization stream. This allows dynamic I/O reconfiguration for the DASD devices without IPL of the JES3 complex. JES3 provides complex-wide data set awareness for DFSMS-managed data sets by communicating with DFSMS through the subsystem interface (SSI). See 4.4, "JES3 SMS Interactions" on page 57.

Note: Complete SMS management of the DASD devices would allow the removal of the DASD DEVICE statements from the initialization stream and dynamic I/O reconfiguration for DASD is possible without an IPL.

A brief summary of DFSMS/MVS and JES3 support for SMS-managed DASD is decribed beginning with 4.3, "DFSMS/MVS Considerations" on page 54.

4.1.1 Dynamic I/O Reconfiguration

Dynamic I/O reconfiguration allows you to make changes to your hardware configuration without an IPL or power-on-reset which improves continuous availability by avoiding planned system outages. Dynamic I/O reconfiguration, however, is not allowed for JES3-managed devices. All configuration changes for JES3-managed devices must be also made to the JES3 initialization stream. The activation of these changes requires a JES3 complex-wide warmstart. JES3 warmstart requires an MVS IPL. Note that JES3 provides data set integrity processing for SMS-managed DASD and SMS.

4.1.2 Global Resource Serialization (GRS)

The MVS sysplex support requires GRS to run in global serialization mode and thus be able to provide guaranteed resource serialization across the whole sysplex for the ENQ RNL=NO requests. A GRS complex that is fully contained within a sysplex uses XCF signaling to build the GRS ring instead of using dedicated GRS CTCs for the ring.

GRS uses XCF signaling paths for its CTC support. In addition GRS exploits the XCF status monitoring services to automatically invoke ring rebuild after a ring disruption caused by a system removal from the sysplex. GRS enhancements for the sysplex support include also the capability to allow dynamic resource name list (RNL) updates initiated through operator commands.

The disposition for all data sets defined in the JES3 and JES3CI procedures should have a DISP=SHR specified. This lets the JES3 global main, local mains and CIFSS address spaces start even if the installation has specified in the GRSRNL parmlib member generic inclusion RNLDEF entry for QNAME(SYSDSN).

Note: RACF should be used to control access to the JES3 checkpoint and spool data sets rather than the DISP=OLD specification on the JES3 JCL.

JES3 uses ENQs with QNAME=SYSZIAT to synchronize internal events. Some of these ENQs are at STEP level and are used to serialize activity within the address space among multiple tasks. Others are SYSTEM level and serialize within an MVS image.

During JES3 initialization, IATINTK issues an exclusive ENQ with QNAME=SYSZIAT RNAME=JES3ACTIVE at the SYSTEM level. If you convert this request to global (SYSTEMS) level, then none of the locals will be able to start. Or if a local is active, the global will not start as it cannot obtain the ENQ resource.

JES3 mains and JES3 CIFSS address spaces serialize access to the checkpoint data sets through a RESERVE request. The RESERVE QNAME is SYSZIAT, and the RNAME is the checkpoint data set VOLSER followed by the data set name. The reserve activity against the checkpoint data set is not great, nor is it of long duration.

The IBM recommendation is to have the generic QNAME(SYSZIAT) in the RNL exclusion list.

4.2 JES3 Tape Management

JES3 main device scheduling (MDS) provides management of DASD and tape devices in a JES3 complex when these devices are defined to JES3 in the initialization stream using the DEVICE statement. This is not an option you should turn off in most environments by specifying SETUP=NONE or by removing the tape DEVICE statements from the JES3 initialization stream.

JES3 management of tape devices is critical to the efficient use of tapes and MVS initiators in both global only and JES3 global/local complexes.

Without JES3 management of the tape drives, jobs go into execution and during step allocation, the MVS initiator attempts to allocate tape units. If the units are not available, the job can wait under the initiator until the allocation is satisfied. This could happen with many active initiators for jobs requiring units. Many initiators are then active but not really executing, causing poor turnaround time for jobs. Depending on the workload, various techniques might be tried to run jobs more efficiently such as:

- Starting more initiators
- Using specific job classes for tape jobs

4.2.1 Benefits of Tape Setup

MDS management of tape devices provides an efficient use of the devices by allowing them to be shared and online to all systems in the sysplex. During MDS allocation, units are assigned to jobs when all the resources a job needs (data sets, volumes, and device) are available. For non-library devices, MDS mount messages are issued for the first use of each device. For tape devices MDS provides:

- Volume awareness (Is the tape volume inuse by another job?)
- Volume unavailability (Is the volume not found and in the volume unavailable table?)
- Pre-execution mounting for non-library drives
- Scheduling of devices to jobs using a setup barrier. See 4.2.2, “Defining a Setup Barrier.”

4.2.2 Defining a Setup Barrier

The purpose of the setup barrier is to prevent lower priority jobs from obtaining resources such as tape drives when higher priority jobs cannot complete their obtaining of all the resources they require to execute. A job′s resource requirements may include devices, volumes, and data sets.

Figure 30 shows some tape units shared between the global and local. Four of the units are currently in use and the remaining tape drive is free to be used by jobs in the system. This figure is being used here to illustrate the use of the setup barrier. MDS processing for jobs on the allocate queue makes a complete pass through all the jobs on the queue. The setup barrier is priority 10 as specified on the current active SELECT mode. Jobs that have a priority equal to or greater than the barrier can reserve units during the current pass of the allocate queue. Once that pass of the allocate queue is finished, all reserves for devices are released. During the allocate queue pass, the job at priority 12 reserves the free unit. During this pass of the queue, the job at priority 10 and at priority 0 do not obtain the free unit. As tape units free up, eventually the jobs above the barrier will reserve units until they can obtain what they need.

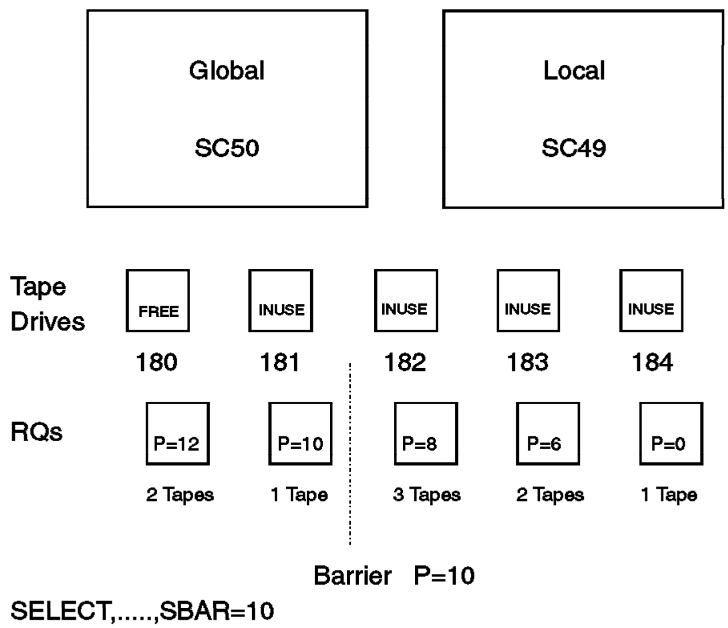


Figure 30. Using the Setup Barrier

4.3 DFSMS/MVS Considerations

The storage management subsystem (SMS) is a DFSMS/MVS facility used to automate and centralize the management of external storage in an SMS complex. An SMS complex is one or more systems that share a common SMS configuration and SMS communication data set.

An SMS complex consists of the systems or system groups (sysplexes) that share a common SMS configuration defined in a common active control data set (ACDS) and a common communication data set (COMMDS) pair. DFSMS/MVS Version 1.3 configuration supports up to 32 system names, system group names, or both. JES3 considers the storage SMS for a main when the main name matches the MVS system name. Note that starting with JES3 5.1.1, this requirement is automatically satisfied by the CPUID= and MODEL= parameter removal from the MAINPROC initialization statement, which also requires the JES3 main names to match the MVS system names.

The DFSMS/MVS software product, together with IBM hardware products, and your installation-specific requirements for data and resource management, comprise the key to system-managed storage in an MVS environment. The components of DFSMS/MVS automate and centralize storage management, based on policies your installation defines for availability, performance, space, and security. Figure 31 summarizes the DFSMS components and JES3 processes that interact with SMS.

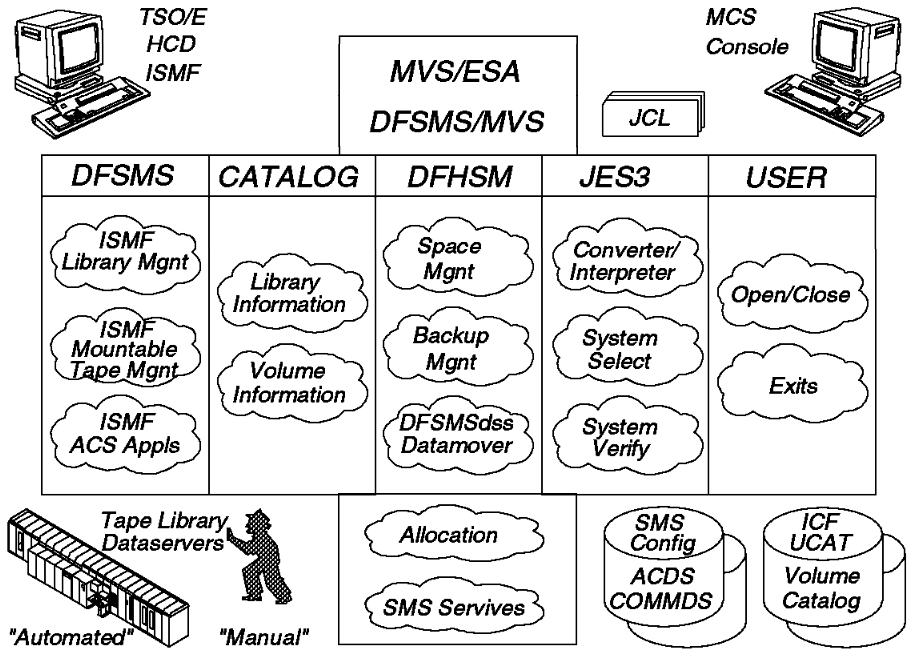


Figure 31. SMS Environment and JES3

Interactive storage management facility (ISMF) is used to define, manage, and analyze the DFSMS/MVS environment. This data and storage management tool is designed to give you interaction with DFSMS/MVS, RACF, and IBM NaviQuest for MVS. (NaviQuest is a testing and reporting tool that speeds and simplifies the tasks associated with DFSMS initial implementation and ongoing ACS routine and configuration maintenance.)

DFSMS/MVS is comprised of the following functional components.

- DFSMSdfp provides the foundation for:
  - Storage management

    DFSMSdfp includes ISMF, an interactive facility that lets you define and maintain policies to manage your storage resources. These policies help to improve the use of storage devices, and to increase levels of service for user data, with minimal effort required from users. SMS manages these policies for the operating system.

  - Data management

    DFSMSdfp helps you store and catalog information on DASD, optical, and tape resources, so that it can be quickly identified and retrieved from the system.

  - Program management

    DFSMSdfp combines programs into executable modules, prepares them to run on the operating system, stores them in libraries, and reads them into storage for execution.

  - Distributed data access

    Allows all authorized systems and users in a network to exploit the powerful features of system-managed storage, or automated storage management provided by DFSMS/MVS. DFSMSdfp uses the Distributed

FileManager (DFM) to support remote access of MVS/ESA data and storage resources from workstations, personal computers, or any other system on a SNA LU 6.2 network.

The hierarchical file system (HFS) works in conjunction with OpenEdition/MVS to provide a full UNIX environment within the MVS/ESA system. MVS/ESA becomes a full-feature UNIX server when coupled with the DFSMS/MVS Network File System. With HFS, MVS programs can directly access UNIX data.

- DFSMSdss is used for:
  - Data movement and replication

    DFSMSdss lets you move or copy data between volumes of like and unlike devices types. It can also copy data that has been backed up.
  - Space management

    DFSMSdss can reduce or eliminate DASD free-space fragmentation.
  - Data backup and recovery

    DFSMSdss provides you with host system backup and recovery functions at both the data set and volume levels. It also includes a stand-alone restore program that you can run without a host operating system.
  - Data set and volume conversion

    DFSMSdss can be used to convert your data sets and volumes to system-managed storage. It can also return your data to a non-system-managed state as part of a recovery procedure.
- DFSMShsm provides functions for:
  - Storage management

    DFSMShsm uses a hierarchy of storage devices in its automatic management of data, relieving end-users from manual storage management tasks.
  - Space management

    DFSMShsm improves DASD space usage by keeping only active data on fast-access storage devices. It automatically frees space on user volumes by deleting eligible data sets, releasing over-allocated space, and moving low-activity data to lower cost-per-byte devices.
  - Availability management

    DFSMShsm backs up your data to ensure availability in the event of accidental loss of the data sets or physical loss of volumes.

    Full exploitation of DFSMShsm services in a DFSMS environment requires the use of DFSMSdss for certain functions. You can also use the DFSMS/MVS Optimizer feature to monitor and tune DFSMShsm functions.
- DFSMSrmm manages your removable media resources, including tape cartridges and reels. It provides functions for:
  - Library Management

    You can create tape libraries, or collections of tape media associated with tape drives, to balance the work of your tape drives and operators.

    DFSMSrmm can manage:

- System-managed tape libraries, such as the IBM 3494 and IBM 3495 automated tape library dataservers, and the manual IBM 3495 tape library dataserver model M10.
- Non-system-managed, or traditional, tape libraries

– Shelf Management

  DFSMSrmm groups information about removable media by shelves into a central online inventory, and keeps track of the volumes residing on those shelves.

– Volume management

  DFSMSrmm helps to manage the movement and retention of tape volumes throughout their life cycle.

– Data set management

  DFSMSrmm records information about the data sets on tape volumes to validate volume and data set information and to help maintain data integrity. It can also control the retention of those data sets.

4.4 JES3 SMS Interactions

JES3 manages all requests for *SMS-managed data sets*. However, JES3 is not aware of specific units for SMS-managed data sets. If an installation does not want JES3 to manage SMS data sets, the SETPARAM initialization statement SMSSETUP=NO parameter specification turns off this feature.

Note: If the SMS complex does not include all JES3 mains, C/I functional subsystems should be defined only on mains where SMS is enabled. Note also that in an SMS configuration where the JES3 complex does not include all systems in the SMS complex, GRS global data set serialization should be used for SMS complex wide data set integrity processing.

JES3 interfaces with SMS during job processing from the converter/interpreter service and the system select phase of MDS.

4.4.1 Converter/Interpreter Phases

JES3 converter/interpreter (C/I) service controls the conversion of JCL statements into internal control blocks. This service comprises primarily the JES3 C/I phase, prescan phase, and postscan phase as shown in Figure 32.

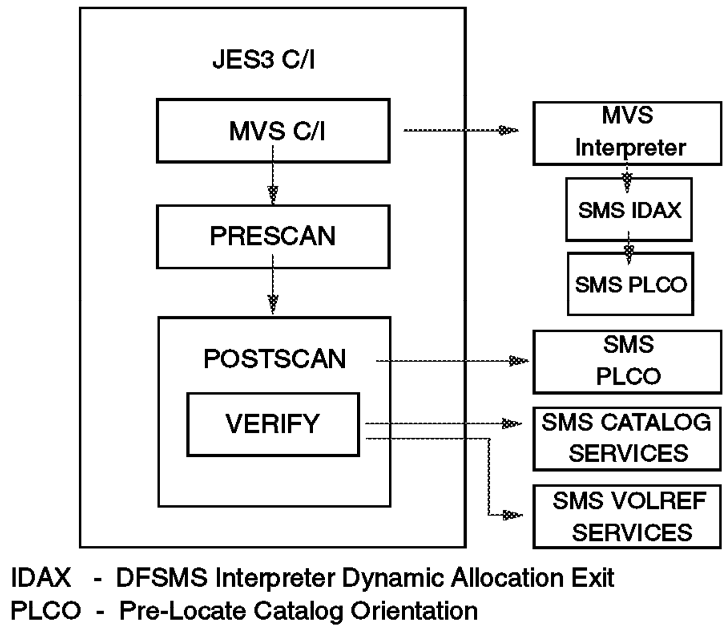


Figure 32. Converter/Interpreter Phases of Processing

- The C/I phase invokes the MVS C/I routines to convert and interpret JCL into scheduler control blocks. At this time, the scheduler control blocks are created in the scheduler work area (SWA).

  At the end of interpretation, the SMS routines are invoked to scan the SWA control blocks to determine the required catalogs for new data sets. Catalogs for jobs can be SMS-managed or JES3-managed. SMS uses the list of SMS-managed volumes to determine which mains are eligible for catalog searches.

- The prescan phase extracts job setup requirements from the scheduler control blocks for use in the postscan phase.

- The postscan phase locates data sets and gathers information for use in JES3 device setup.

  When SMS is active, JES3 calls it to search the system catalogs and collect scheduling information for the job. SMS searches the system catalogs to obtain information about a job′s data sets when the job′s JCL does not specify unit information.

  If a data set has been migrated (or is eligible to be migrated) by the hierarchical storage manager (DFSMShsm), the catalog locate request for that data set causes JES3 to associate the data set with a set of volumes to which it can be recalled. DFSMShsm limits the choice of volumes eligible for recall during catalog locate processing in accordance with its space management algorithms. JES3 processing continues as though the data set is recalled to all eligible volumes. However, the actual recall does not occur until job execution when MVS reissues the catalog locate request for the migrated data set. At that time, DFSMShsm determines which volume is the best choice to recall the data set to and then recalls the data set to that volume. DFSMShsm derives an SMS storage class and a SMS storage group for SMS-managed data sets.

DFSMShsm delays migration for any non-SMS-managed data sets that have been processed by the JES3 setup function. By delaying their migration, DFSMShsm ensures that non-SMS-managed data sets are not migrated between the time that they are processed by JES3 setup and the time that the job is actually run.

DFSMShsm controls migration prevention of non-SMS-managed data sets that have been processed by the JES3 setup function. The DFSMShsm default migration prevention is for the remainder of the processing day plus three more days. Through the delayed migration, DFSMShsm ensures the data integrity of non-SMS-managed data sets in a JES3 environment. A data integrity exposure would occur if a non-SMS managed data set is recalled to another volume than indicated by the earlier catalog locate. The defaults also apply to the period of time that a recalled data set is prevented from migrating again.

Note: In a JES3 environment, the SMS volumes and the non-SMS, DFSMShsm-managed volumes in the DFSMShsm general pool must be shared by all processors. However, if some of the volumes in a data set or volume pool are also in the general pool, all volumes in both pools must be shared.

4.4.2 Main Device Scheduling Phases

For SMS-managed data sets, there are two additional phases of MDS processing:

- System select
- System verify

4.4.2.1 System Select Phase of MDS

The system select phase of MDS, shown in Figure 33, is performed when a job requires one or more resources managed by SMS. JES3 is not aware of the availability or connectivity of SMS-managed resources. If a job requires SMS-managed resources, JES3 requests SMS to determine the availability of those resources and to determine which mains have access to those resources. When a required SMS-managed resource is temporarily unavailable, the job waits in system select until the resource becomes available.

If all required SMS-managed resources are available, SMS provides JES3 with a list of mains that have access to those resources. JES3 uses the list of eligible mains passed by SMS to determine which mains have access to all of the required resources for the job (both SMS-managed and MDS-managed.) If JES3 determines that one or more mains have access to all of the required resources, the job proceeds into the MDS allocation phase.

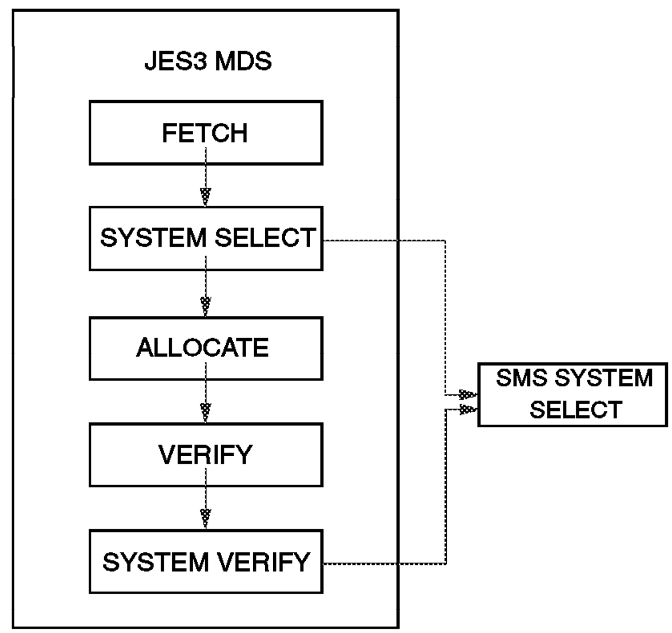


Figure 33. Main Device Scheduling (MDS) Processing Phases

4.4.2.2 System Verify Phase of MDS

The system verify phase of MDS is performed when a job requires SMS-managed resources. This phase of MDS ensures that all of the SMS-managed resources required by a job are still available before execution. For example, if a job spends too much time in MDS allocation or a long period of time elapses between a mount request and the actual mounting of a required volume, one or more of the SMS-managed resources could become unavailable.

If the status of SMS-managed resources required by the job has not changed, that is, all of the required SMS-managed resources are still available and are accessible by the eligible main(s), the job can proceed into execution.

However, if SMS-managed resources required by a job are no longer available, the job is sent to the breakdown phase where MDS deallocates resources held by the job. MDS then sends the job back to the system select phase, where MDS retries allocation.

4.4.3 SMS and RACF Considerations

When a batch job allocates a new SMS controlled data set, the data owner′s authority to the SMS constructs STORCLAS and MGMTCLAS is checked by SMS routines invoked during C/I processing in the JES3 address space or in a JES3 CIFSS address space. Therefore, it is necessary to:

- Have an entry for JES3 in the RACF started procedures table or STARTED class that defines it as privileged
- Have an entry for JES3CI in the RACF started procedures table or STARTED class with the same properties as for JES3
- Make sure that SETROPTS RACLIST processing has been activated for the RACF classes STORCLAS and MGMTCLAS.

Note also that batch jobs will be canceled out of the MVS Converter/Interpreter if the data owner is not authorized to the MGMTCLAS and STORCLAS of an SMS dataset that is to be created by the job.

4.4.4 JES3 and SMS Configuration Considerations

A JES3 complex can interact only with one SMS complex. If SMS is used in a JES3 complex, it must be active at least on the JES3 global processor. On local processors, SMS may or may not be installed and activated.

When a storage group is defined to SMS, the volumes belonging to that storage group have to be defined. Part of the definition is to specify which processors have access to the volume (ENABLE). If a specific processor does not have access to a volume, DISALL has to be specified for this processor. The other system status options (DISNEW, QUIALL, QUINEW) should be used in cases when volumes are being migrated into and out of storage groups.

Note: When a volume is added to a storage group, it will be enabled to all systems in the SMS configuration by default. If the volume cannot be accessed from all SMS processors, do not forget to change the system status for the volume accordingly.

JES3 provides data set integrity processing for SMS-managed data sets even if the SMS managed devices are defined to JES3 with a DEVICE initialization statement. If an SMS-managed device is defined to JES3 through a DEVICE initialization statement, it cannot be deleted with the HCD dynamic configuration change. If an SMS-managed device is *not* defined to JES3 through a DEVICE initialization statement, it *can* be deleted with the HCD dynamic configuration change. New devices can always be added to an I/O configuration using HCD dialogs and the new configuration can be successfully activated. If the new devices are made SMS-managed, JES3 will provide data set integrity processing for the data set on the newly added devices.

4.4.4.1 Changing the SMS Configuration

When a new or changed SMS configuration is activated, all jobs in the setup phase are rescheduled by MDS and their setup requirements are reviewed against the new configuration.

Jobs that are already queued for main service (or are executing) are not rescheduled or restarted, so it has to be expected that these jobs that are affected by the configuration changes might fail during execution.

4.4.5 Catalog Connectivity Considerations

Connections between catalogs and processors may be asymmetric in a JES3 complex. In an installation using SMS, all catalogs should be ICF catalogs. If a catalog resides on an SMS-managed volume, SMS determines which processors have access to the catalog. If a catalog resides on a JES3-managed device, JES3 catalog setup will determine which processors can access the catalog. If a catalog resides on a device that is MVS-managed only, JES3 cannot determine which processors have access to the catalog and, consequently, locates may be scheduled to the wrong processors or jobs may go into execution on processors where the catalogs that are needed for data set allocation are not available.

Therefore, catalogs should preferably reside on SMS-managed volumes. If a catalog does not reside on an SMS-managed volume, the device should be JES3-managed.

4.5 JES3 Local Main Failures

If a local main fails, TSO users or batch jobs running on that main cannot be restarted until the failed main is either re-IPLed or “flushed.” Both of these actions require operator intervention and may cause availability problems if the failure goes undetected for a while. JES3 does not use XCF monitoring to detect a local outage and inform the operators of the failure. See Appendix E, “Sample Program Using JESXCF Services” on page 131 that contains a JES3 DSP to flush the failed local main.

4.5.1 JES3 DSI Processing

For the Parallel Sysplex environment, JES3 does not support an automated dynamic system interchange (DSI) procedure. If the JES3 global processor fails, and it cannot be brought back within a reasonable period of time, DSI allows JES3 to continue operation by switching the global function to a local main. DSI normally requires operator intervention and takes some time. While DSI is in process, all work that requires JES3 services is suspended.

Changes in the DSI procedures are possible beginning with JES3 5.2.1 since consoles no longer need to be dedicated to the global processor. This eliminates having to switch the consoles from the failed global to the new global.

Chapter 5. Parallel Sysplex Migration Considerations

There are many important migration decisions that need to be made during a conversion to the sysplex environment. This chapter discusses these issues.

JES3 is the job entry subsystem that enables you to manage work for multiple MVS images in a JESplex as a single-system image. The advantage of this to an operator is the ability to control the sysplex as though it were a single entity. For example, through commands with sysplex-wide scope, operators can control all the MVS images in the sysplex almost as though only one MVS image existed.

5.1 Parallel Sysplex Application Environment

A Parallel Sysplex provides an ideal environment for online transaction processing. The transaction workload of a transaction processing system is typically composed of a large number of independent, relatively simple, processing requests from a large set of requestors (often a terminal network). Theoretically there is a high degree of parallelism in the workload that can be executed on a highly parallel architecture relatively easily. Figure 34 summarizes the OLTP Parallel Sysplex components.

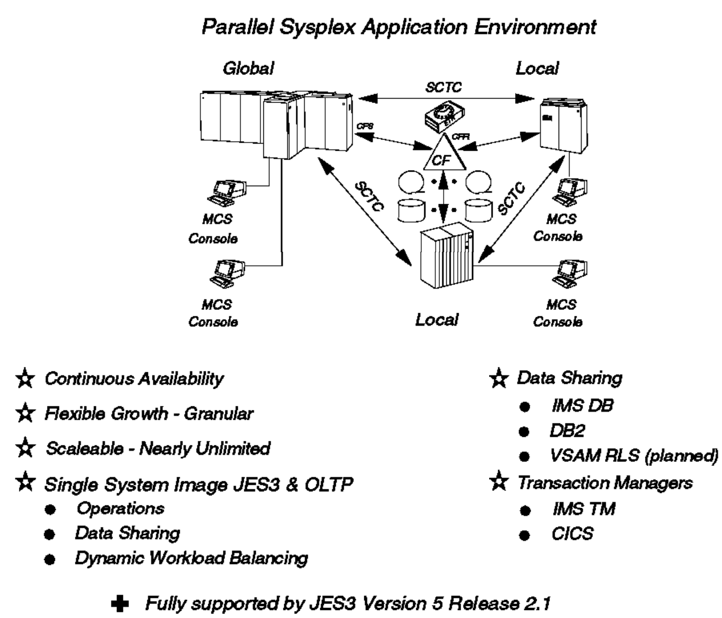


Figure 34. Parallel Sysplex Application Environment

5.1.1 Continuous Availability

Automation plays a key role in availability. Typically, automation routines are responsible for bringing up applications, and if something goes wrong, automation handles the application′s restart. The MVS automatic restart management provides restart capability for failed subsystems, components, and applications. For example, when a subsystem such as CICS, IMS DB, or DB2 fails, it is likely be holding resources, such as locks, that prevent other applications from accessing the data they need. Automatic restart management quickly restarts the failed subsystem; the subsystem can then resume processing and release the resources, making data available once again to other applications.

Note: Automated operations control/MVS (AOC/MVS) has awareness of MVS automatic restart management, so that restart actions are properly coordinated.

5.1.2 Single System Image

To meet the challenges to run transactions successfully in parallel requires:

- The parallel systems to provide a single system image for the user.
- Transactions be able to update or access the same data with integrity and with varying patterns running under several MVS images in several CPCs.
- Applications to be available without interruption, even when some components fail.
- The parallel systems be able to execute existing programs without modification.

The features of the Parallel Sysplex, exploited by IBM subsystems and products, are designed to produce a system that meets all of these challenges:

- The single system image advantage for the end user is the ability to log onto an application in the sysplex, and be able to access that application without being concerned about which system or systems the application resides on.

  For example, CICS uses VTAM generic resources function, which allows an end user to log onto one of a set of CICS terminal-owning regions through a generic name, thus providing single system image for VTAM logons to CICS TORs. With dynamic workload management, provided by CICSPlex SM, the transaction workload is then balanced across the CICS application-owning regions (AORs), providing single system image from an application perspective.

  CICSPlex System Manager/ESA (CICSPlex SM) is a system management tool that enables you to manage the multiple CICS regions in a CICSplex as a single-system image.

  The provision of a single-system image enables you to operate at the logical rather than the physical level, without regard to either the scale or location of CICS resources. That is, you can operate the CICSPlex as though it is a single CICS region. CICSPlex SM provides a general-purpose interface to a variety of CICS system management functions that allow operators to manage multiple CICS regions, within a sysplex, from a single point of control.

5.1.3 Coupling Facility and Data Sharing

The Parallel Sysplex environment coupling facility provides the solution for high performance multisystem data sharing, which is exploited by the database management component of IMS/ESA. This is used by CICS application-owning regions that need to access databases managed by IMS Database Control (IMS DBCTL).

5.1.3.1 IMS Data Sharing

IMS has provided a database and transaction processing environment for more than a quarter of a century. Over the years, IMS DB and IMS TM have been extensively enhanced to meet the needs of customer business applications, which have continued to grow and place increasing demands upon IMS services. IMS DB is enhanced to use the sysplex coupling facility to support high-performance, multisystem data sharing in the IMS DB/DC environment. The IMS DB data sharing through the coupling facility is no longer limited to two systems. Therefore, the total processing capacity that can be applied to access a given database is increased considerably.

IMS DBCTL data sharing ensures also that all CICS-DL/I transactions, running in any CICS region that is connected to an IMS DBCTL environment, have equal access to the databases. You are no longer restricted by the number of MVS images running DBCTL, therefore the total processing capacity that can be applied to access a given database is increased considerably. Note that IMS multisystem data sharing is only available through IMS DBCTL; it is not available for CICS local DL/I applications.

Note: There is no need to change the structure of your IMS applications, or application programs, or data, to migrate to a data-sharing Parallel Sysplex environment.

5.1.3.2 DB2 Data Sharing

DB2 Version 4 uses the sysplex to provide multisystem data sharing, which can provide many benefits, including more processing capacity, higher availability, and more ways to configure systems to meet the needs of various user groups. And DB2 can take advantage of these benefits while continuing to serve as your enterprise data server.

Having multiple DB2s capable of handling the transaction workload provides an opportunity for greater availability, especially when you perform maintenance, by allowing users on one system to be moved to another system. If you use CICSPlex SM, unplanned outages can be less disruptive because transactions can automatically be routed away from the failed system. DB2 can also takes advantage of MVS automatic restart management to get the DB2 subsystem up again as quickly as possible.

5.1.3.3 Dynamic Workload Balancing

With redundancy, symmetry, and dynamic workload balancing, your applications can remain continuously available across changes, and your sysplex remains resilient across failures.

Using CICS multiregion operation (MRO), you can separate CICS functions into individual regions (address spaces), the different types of CICS regions being classified as resource managers. With the latest enhancements to MRO, these CICS resource manager regions can reside in one or more MVS images.

A set of interconnected CICS regions acting as resource managers, and combining to provide a set of coherent services for a customer′s business needs, is often referred to as a CICSplex. In its simplest form, a CICSplex operates within a single MVS image, and uses CICS multiregion operation facilities. Within a sysplex environment, a CICSplex can be configured across all the MVS images in the sysplex environment.

The CICS resource manager regions are terminal-owning regions (TORs), application-owning regions (AORs), and file-owning regions (FORs). However, to migrate CICS applications to a sysplex environment successfully, you may need to extend and change the use of this resource manager principle even further in your installation. For example, combined regions such as a terminal- and application-owning region, or an application- and file-owning region, are not recommended in a sysplex environment, and you should plan to split these functions into separate regions.

You should also investigate how your applications use the CICS temporary storage and transient data facilities. If, after investigation, it is necessary to do so, create queue-owning regions for the management of CICS temporary storage and transient data queues.

In addition to the CICS-based resource managers, CICS also supports access to other (non-CICS) data managers through its resource manager interface (RMI). The two IBM-supplied resource managers that use the CICS RMI are DB2 and DBCTL (DBCTL being the strategic interface to IMS databases in preference to the CICS local DL/I interface). If your applications require access to DB2 or IMS databases, you must plan how to access these databases in a sysplex environment.

All the traditional benefits of splitting CICS into several resource manager regions and linking them into an MRO-based CICSplex are valid in a sysplex environment. With the introduction of the coupling facility data sharing sysplex, there are additional benefits from operating an MRO CICSplex across multiple MVS images in the sysplex environment.

There is no requirement to change the structure of CICS applications, or the application programs, or data, to migrate to a data sharing sysplex environment.

5.2 Sysplex Naming Conventions

In the sysplex environment, where you are likely to have large numbers of subsystems installed, meaningful naming conventions are very important. The more subsystems you have to deal with, the more important it is to have good naming conventions.

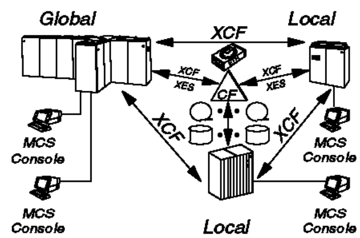
Naming conventions are particularly important where you have multiple instances of a subsystem, such as CICS and IMS, many of which are connected through communication links. Even when subsystems are not directly connected, use a common naming convention for all subsystems that coexist within a sysplex. Wherever possible, design your conventions with as wide a scope as possible (for example, global scope within an entire corporate network).

Good naming conventions and the use of system symbols allows you to create a number of identical application instances and to spread them across the sysplex to support OLTP workload balancing. Creating identical application instances is also called cloning.

5.3 JES3 Parallel Sysplex Migration

Figure 35 summarizes JES3 support for Parallel Sysplex.

JES3 in Parallel Sysplex Environment



- *Changed in JES3*
  - + *Inter-processor communication - based on common MVS components*
  - + *Parallel sysplex support - 32-way sysplex, 4-digit device numbers*
  - + *Operator communication - based on MVS sysplex operations*
  - + *Work recovery - MVS automatic restart management supported*
  - + *JCL enhancements - nearly all MVS JCL enhancements supported*
- *Not changed in JES3*
  - + *JES3 resource management - preserves JES3 unique features*
  - + *JES3 work flow - minimizes migration effort*
  - − *JES3 warmstart requirements - not so good for continuous availability*
  - − *JES3 DSI - not automated - requires manual intervention*

Figure 35. JES3 Support for Parallel Sysplex

5.3.1 Inter-Processor Communication

JES3 Version 5 has undergone major internal changes in the area of inter-system communication and console support. JES3 Version 5 uses of the MVS cross-system coupling facility (XCF). The MVS coupling services allow authorized programs on one or more MVS systems to communicate through ESCON channels. This eliminates the need for JES3-managed CTCs for global/local communications. Communication paths between processors can be added dynamically without concern for JES3 device definitions.

Because support for JES3-managed CTCs is eliminated, all systems in a JES3 complex running at the MVS/ESA SP Version 5 or higher level must be migrated to the new level of JES3 at the same time.

5.3.2 Operator Communication

JES3 5.2.1 improves systems management and automated operations by enabling the sysplex operations features of MVS/ESA. System and subsystem code can build on these features without the need to limit function or provide alternative implementations in the JES3 environment. Sysplex-wide control is available from consoles on any system rather than a subset of the console on the global processor.

5.3.3 Work Recovery

The JES3 release that best satisfies the Parallel Sysplex OLTP application migration and cloning requirements is JES3 5.2.1. You may start your OLTP application migration even before you have migrated to JES3 5.2.1. However, pre-JES3 5.2.1 releases do not support system symbols and ARM restarts, which are an essential part of the IBM unique OLTP support.

Note: Limited system symbol and ARM testing can be done under the MSTR (master) subsystem, but then again, you cannot use the JES3 spooling services.

One of the primary goals of a sysplex is continuous availability. The MVS automatic restart management provides restart capability for failed subsystems, components, and applications. Automatic restart management plays an important part in the availability of key MVS components and subsystems, which in turn affects the availability of data. See 3.1, "Automatic Restart Management" on page 41.

Data sharing, which is the ability to concurrently directly access and change the same data, while maintaining data integrity, may provide some scheduling relief to the subset of batch work that requires access to IMS and DB2 data bases. Prior to the availability of data sharing, batch work had to be scheduled to the processor where the resource manager was active. Data sharing allows multiple instances of the resource managers be active on multiple processors, which also allows batch work to be scheduled more freely, resulting in improved workload balancing and resource utilization.

The JES3 considerations for TSO or APPC work are not directly affected by the Parallel Sysplex support.

5.3.4 JES3 Resource Management

JES3 5.2.1 has preserved its resource management and workflow management the same as it had been in prior releases of JES3. The applications participating in the OLTP are in most cases "cloned" demand select jobs (started tasks) which do not present any special JES3 considerations from the resource management and workflow point of view.

5.4 Multiple JES3 Complexes in the Same Sysplex

JES3 Version 5 design allows multiple JES3 complexes be active simultaneously in a single sysplex. This may be desirable, for example, when an installation needs to combine multiple standalone JES3 complexes into a single data sharing environment, or when there is not enough hardware resources to run independent sysplex test and production environments without sharing critical resources (for example, DASD).

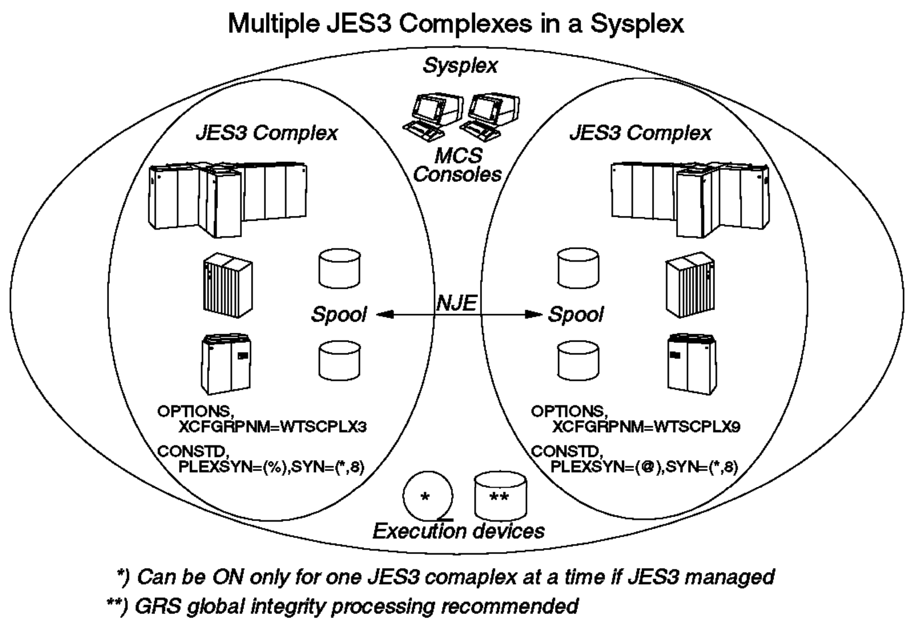


Figure 36. Multiple JES3 Complexes in the Same Sysplex

5.4.1 Multiple Complex Considerations

The following JES3 definition considerations should be followed when running more than one JES3 complex in the same sysplex:

- Each JES3 complex must be defined to have a unique name. This name is specified on the OPTIONS initialization statement or is defaulted to the NJE home nodename. See 5.5, "JES3 Initialization Changes for Sysplex" on page 70.
- Considerations for selecting the command prefix:

  JES3 internally issues commands using the "*" prefix. Thus, if more than one global is active in the sysplex, the "*" must be explicitly defined in each initialization stream as SYN=(*,..), a SYSTEM-scope prefix, as shown in Figure 36. Otherwise, any internally JES3-issued commands may be processed by a wrong global. See 5.5.4.1, "Command Prefix Definitions" on page 72.

  A unique SYSPLEX-scoped prefix must be specified for each global. If not specified, JES3 attempts to register the "*" as the SYSPLEX-scoped prefix. This also allows any console in the sysplex to send commands to the global and receive the responses.

  If the use of the character "8" as a prefix is required, then specify more than one SYN= prefix. Figure 36 shows the prefixes for a sysplex with more than one global.

5.4.2 Execution Devices

Other considerations include execution devices. The JES3 data set integrity processing does not protect data on jointly managed shared DASD devices *between* the JES3 complexes. GRS global serialization should be activated for data set integrity processing which, on the other hand, may lead work to wait for data sets during MVS allocation. Installations can specify in ALLOCxx parmlib member through the SDSN_WAIT policy whether to cancel jobs that must wait for data sets, or to allow wait.

A JES3-managed tape execution device can be online to only one JES3 complex at any time. The operators must manually control the online/offline status of that tape device for a given JES3 complex. When an assignable tape device is brought online to a JES3 complex, JES3 assigns the device and varies it online to MVS. The device cannot be varied online to other JES3 complexes as long as it is assigned (that is, it is online to a JES3 complex).

An alternative for JES3-managed assignable tapes is the MVS/ESA SP 5.2 shared tape allocation in a Parallel Sysplex. MVS/ESA SP 5.2 allocation supports automatic tape switching, the ability for an installation to have a category of non-JES3 managed tape devices known as automatically switchable. These devices are not dedicated to any one system; rather they can be used by all systems that are connected through a coupling facility in a Parallel Sysplex. The coupling facility is used to maintain the sysplex-wide allocation status of the automatically switchable tape devices. See 3.2, “MVS Shared Tape Allocation” on page 45.

5.5 JES3 Initialization Changes for Sysplex

This section discusses the JES3 initialization changes necessary for the sysplex environment.

5.5.1 Defining the XCF Group Name

The XCF group name is required for each system when it joins the sysplex. The group name can be defaulted, or you can choose the XCF group name on the OPTIONS statement.

```
OPTIONS   XCFGRPNM=groupname
```

JES3 determines the XCF group name to be used by making the following decisions:

1. Use the groupname from the OPTIONS XCFGRPNM= initialization parameter. The XCFGRPNM= parameter is optional.
2. Use the nodename from the NJERMT NAME= specification.
3. Use the default nodename of *N1* when no NJE networking is defined.

Note: The recommendation is to let the group name default to the node name. Due to syntax constraints, this may not be possible (the allowable node names in JES3 are a superset of allowable XCF group names).

5.5.1.1 JES3 XCF Member Names

Each member of an XCF group is identified by name. This member name is either the JES3 main name defined on the MAINPROC statement for JES3 global and local address spaces, or the FSS name on the FSSDEF statement.

Note: The MAINPROC name *must* match the SYSNAME parameter specified in the IEASYSxx parmlib member.

5.5.1.2 Defining System Names

Table 1 and Table 2 show where the specifications can be made to choose a naming convention in which the SYSNAME, JES3 main name, and SMF SID are the same.

Table 1. SYS1.PARMLIB Definitions

PARMLIB Member	Parameter Specification
IEASYSxx	SYSNAME=SC50
SMFPRMxx	SID(&SYSNAME(1:4))

Table 2. JES3 Initialization Statements

Initialization Statement	Parameter Specification
MAINPROC	NAME=SC50
MAINPROC	NAME=SC49

5.5.2 CTC Replacement Definitions

CTC addresses are no longer required in the JES3 initialization parmlib member. They were defined as follows on the DEVICE statement:

```
DEVICE,DTYPE=SYSMAIN,JNAME=SC51,JUNIT=(ctc-add,SC50,,ON,.......
```

This statement is still required, but the *ctc-add* is not. CTCs are now defined using the COUPLExx parmlib member as shown in A.2, “COUPLExx Parmlib Member” on page 90. The DTYPE of SYSMAIN defines the main processor as a JES3 support device and gives an initial online/offline status. This also allows the device to be varied on and off by the operator with the JES3 VARY command.

```
DEVICE,DTYPE=SYSMAIN,JNAME=SC51,JUNIT=(,SC50,,ON,...
```

5.5.3 CONSTD Statement

Where the new keywords are:

GLOBMPF GLOBMPF={YES|NO}

YES indicates whether messages routed by MCS to the global processor should be made available to MPF processing on the global in addition to MPF processing on the originating local. NO is the default.

PLEXSYN PLEXSYN={(syn1,...syn6)|*}

Specifies a sysplex scope for the command prefix. The sysplex scope means that any command issued with this prefix from any system in the sysplex executes on the global processor. The default is *. This keyword is used together with the SYN keyword to determine the prefix to be used. If a prefix is defined on keywords, the prefix on the SYN keyword is used as a system-scoped prefix.

DLOG DLOG={ON|OFF}

5.5.3.1 Changed Keywords

- EDIT=({escape,bkspace,newline,linedel})

5.5.4 Deleted Keywords (Ignored)

- FLAG=
- CONSBUF=
- HARDCOPY=
- WTP=

5.5.4.1 Command Prefix Definitions

Command prefix facility (CPF) processing is enabled sysplex-wide in a JES3 5.2.1 complex. The CPF scope is no longer limited to a single system as it is in a sysplex running JES3 releases prior to JES3 5.2.1. JES3 registers the SYSPLEX (PLEXSYN=) and SYSTEM (SYN=) scope command prefixes that are defined on the JES3 CONSTD initialization statement, as shown in Figure 37.

Up to six sysplex-wide JES3 prefixes (also known as synonyms) can be specified. The sysplex-wide prefixes are always registered with CPF on the global processor and are re-registered during the dynamic system interchange.

Note: To avoid conflicts with short-form WTOR replies, the JES3 default SYSTEM scope prefix (8) is not registered with CPF and therefore is not displayed by the operator command *D OPDATA*.

5.5.4.2 Defining the JES3 Command Prefix

A new keyword, PLEXSYN, defines the sysplex command prefixes.

```
CONSTD,EDIT=({escape,bkspace,,linedel}),
       GLOBMPF={YES|NO},
       SYN={(syn1,...syn6)|8}
       PLEXSYN={(syn1,...syn6)|*},
       CIFSS={FSSDEF|MSGROUTE},
       DLOG={ON|OFF}
```

Figure 37. Command Prefix Definition on CONSTD Statement

PLEXSYN PLEXSYN={(syn1,...syn6)|*}

Specifies a sysplex scope for the command prefix. The sysplex scope means that any command issued with this prefix from any system in the sysplex executes on the global processor. The default is *. This keyword is used together with the SYN keyword to determine the prefix to be used. If a prefix is defined on both keywords, the prefix is used as a system-scoped prefix.

5.5.4.3 SYSPLEX Scope Processing

JES3 5.2.1 registers with CPF sysplex-wide command prefixes that are defined with the PLEXSYN= parameter on the JES3 CONSTD initialization statement. Up to six sysplex-wide JES3 prefixes (also known as synonyms) can be specified. The sysplex-wide prefixes are always registered with CPF on the global processor and are re-registered during the dynamic system interchange. In general, commands routed explicitly with the MVS ROUTE command and with sysplex-wide prefixes are rerouted by MCS as required according to the sysplex-wide prefix definition after the ROUTE command routing completes. If a command is routed using a sysplex-wide prefix, MCS routes it only *once*, even if the command has multiple sysplex-wide prefixes.

JES3 defines both SYSTEM and SYSPLEX scope prefixes with the option FAILDISP=PURGE. This means that a command prefix is deleted when the defining system is removed from the sysplex or the defining address space terminates. The CPF REMOVE=NO option is used by JES3 for the prefix removal. This implies that both the command prefix and the command are presented on the receiving system. The length of both SYSTEM and SYSPLEX scope prefixes can be one to eight characters.

If a prefix is specified on both the SYN= and PLEXSYN= parameter, JES3 defines it as a system-wide (SYN=) prefix. If the JES3 global fails to define one or more of the prefixes specified on the PLEXSYN= parameter, a warning message is issued and the prefix is ignored. When every PLEXSYN= prefix registration fails, JES3 uses the default SYSPLEX-scope prefix * and issues a warning message stating that the default sysplex-wide prefix is being used.

Note: Even if * is the default sysplex-wide prefix, it is recognized as a *valid JES3 command prefix on a local* when it is passed to command processing on a local. This happens, for example, if a * JES3 command is routed to a local using double prefixing. The * has preserved the *guaranteed-to-work* JES3 command prefix role for compatibility reasons. This way, the * can be used as the fixed JES3 command prefix by automation products. In this case, the command transportation should be implemented using double command prefixing.

If JES3 fails to register the * prefix, there will *not* be any JES3 sysplex-wide prefixes, and all JES3 commands from locals must be explicitly routed to the global using either the MVS ROUTE command or double command prefixes. The first of the double prefixes must be defined as "REMOVE=YES," to route the command to the global. The second prefix should be the JES3 "*" or any of the SYSTEM-scope prefixes registered by the global.

Note: Double prefixes are allowed for commands. MCS uses the first prefix to transport the command. Once the command is transported, MCS then presents the command with the remaining prefix to the command processors on that system.

5.5.4.4 SYSTEM Scope Processing

The existing SYN= parameter on the CONSTD initialization statement continues to define up to six SYSTEM scope JES3 prefixes. The SYSTEM scope prefixes are registered on *each* system in the JES3 complex. Commands that begin with SYSTEM scope prefixes are processed on the processor on which the command is entered. Explicitly routed commands with SYSTEM-wide prefixes are processed on the target processor.

When JES3 is unable to register one or more of the SYSTEM scope prefixes specified on the SYN= parameter on any processors, a warning message is issued. If all prefixes specified on the SYN= parameter fail to be defined to CPF on a processor, JES3 uses the default prefix value 8 on that processor and issues a warning message stating the fact.

Note: The 8 prefix is implemented as a SYSTEM scope prefix but is not registered with CPF, to avoid conflicts with short-form WTOR replies.

5.6 JES3 5.2.1 Command Processing

Figure 38 illustrates the command processing flow in JES3 5.2.1. All commands go through the MCS MGCR(E) interface, which serves as the single point of the command entry, and all commands are presented to MPF command exits. The MCS MGCR(E) processing invokes the MPF command exits prior to entering the SSI loop for the command processing. JES3 SSI processing (IATSI34) examines input commands to see if they are valid JES3 commands. Valid JES3 command are sent to JES3 through the SSISERV service and further MGCR(E) processing is terminated.

Commands can be entered from the same sources as in prior JES3 releases except for the JES3 locally attached consoles, which are deleted. In addition, support for JES3 command entry through a BDT session is deleted. The JES3 commands authorization exit (IATUX56) for commands entered through BDT is also deleted.

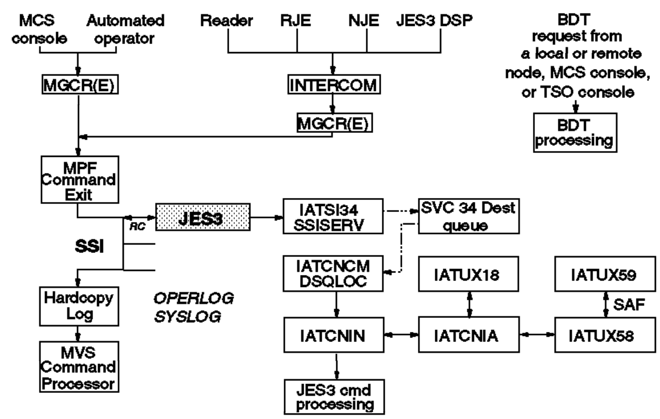


Figure 38. JES3 5.2.1 Command Processing

5.6.1 Command Stacking

The processing for commands entered from an MCS console or an automated operator interface through MGCR(E) remains the same. When multiple commands are entered through a single input from an MCS console, MCS unstacks commands and sends one command at a time to the SSI. When multiple commands are entered through a single MGCR(E) by an automated operator, the whole stack of commands is sent by MCS to the SSI. If the first command is an MVS command, the JES3 SSI routine returns the whole stack of commands back to MCS for processing. If the first command is a JES3 command, the JES3 SSI routine sends the whole stack of commands to the JES3 address space for processing. The JES3 command processing continues to provide the unstacking service as in prior releases of JES3.

5.6.1.1 Command Stacking through CONSOLxx

JES3 no longer uses the delimiter defined by the EDIT parameter in the CONSTD initialization statement to unstack the commands because the *newline* option has been removed.

```
CONSTD,EDIT=({escape,bkspace,newline,linedl})
```

Instead, it uses the MCS command delimiter that is defined by the CMDDELIM parameter on the INIT statement in the CONSOLxx parmlib member.

```
INIT  CMDDELIM(c)
```

5.6.2 JES3 DSPs and INTERCOM

A JES3 DSP can internally issue a JES3 command through INTERCOM macro (simulate input of operator command). The INTERCOM macro is converted into an MCS MGCRE.

JES3 command processing (IATCNCM) processes staging areas on the SVC 34 destination queue and enters the commands contained in the staging areas for execution. IATCNIN receives control from IATCNCM to perform input command processing.

5.7 JES3 5.2.1 Command Authorization and Exits

JES3 5.2.1 enters all commands, independent of their source, to the system through an internally issued MGCRE macro. The MGCRE processing may invoke one or more MPF command installation exits to affect command processing. The command exits receive control every time a command is entered.

5.7.1.1 Command Authorization with IATUX18

In JES3 5.2.1, exit IATUX18 is *not* entered for MVS commands issued from a JES3 source; it is entered only for JES3 commands.

During the JES3 5.2.1 migration, current IATUX18 installation exit code related to non-JES3 commands entered from JES3 sources must be moved to an MPF command installation exit. These exits are still running in the JES3 address space, and JES3 address space data, such as device group, continues to be available for the exits.

As shown in Figure 38 on page 74, for JES3 commands, IATCNIN calls the JES3 console authorization checking module (IATCNIA) to validate command authority. IATCNIA calls user exit IATUX18. This exit routine allows you to modify a JES3 command and validate the console′s authority to enter the command. If the operator enters a JES3 command at a console that has been defined as not having a high enough authority level for that command, the command is rejected. You could use this exit to allow a particular command to be issued from a console whose definition would reject the command.

Note: IATUX18 no longer sees every command entered from JES3 sources. Installation exit code related to non-JES3 commands entered from JES3 sources must be moved to an MPF command exit.

A new return code (16) is provided which may be used when a command is completely processed within the exit. When this return code is used, JES3 performs no further processing for the command. (Previously, JES3 would issue an error message for any unrecognized command processed within the exit.)

Changes in the exit interface and return convention will likely necessitate updates to exiting exit code as part of the migration to JES3 5.2.1

Return Codes from IATUX18: Indicates how the command is to processed:

0 The command is to be processed; bypass JES3 console authority default checking

4 The command is to be rejected; bypass JES3 console authority default checking

8 The decision is to be made by IBM; use JES3 console authority default checking

12 The decision is to be made by IBM; use JES3 console authority default checking. Also, the exit should be made a dummy exit and should not be called again.

16 The command has been processed; bypass JES3 console authority default checking

The JES3 command authorization process includes invocation to the security authorization facility. Before JES3 calls SAF to perform security processing, exit IATUX58 is given control. This exit allows you to modify security checks or to make your own security decisions for JES3. After the SAF call, JES3 calls exit IATUX59. Also, this exit gives you the opportunity to modify security checks or to make security decisions for JES3.

Note: IBM recommends that you use a security product such as RACF for command authorization instead of exit code whenever possible. When exit code is required, MPF command exits should be considered first. Use IATUX18 only for code that must run in the JES3 address space.

After successful completion of the command authorization checks, the command is passed to the appropriate command executor.

Exit IATUX18 continues to be called for *all* JES3 commands, independent of their source. In summary, it is recommended to continue to have all installation JES3 commands processing in the JES3 command exit, and move all installation non-JES3 commands processed in to MPF command installation exits (or other facilities).

5.7.1.2 Command Authorization with SAF/RACF

MCS consoles and MGCRE provide SAF command authorization processing, which became available with MVS/SP Version 3 Release 1.3 when RACF Version 1 Release 9 and JES3 Version 3 Release 1.3 were also installed. MCS uses a SAF-based command authorization scheme for controlling and auditing operator command execution. All operator commands are controlled through SAF and the security product (RACF) on a command by command basis. Of course, this support is optional and must be activated through the use of RACF profiles.

Each MVS and JES3 command has an entity name and an access level associated with it. Each command issuer has an associated UTOKEN (an encapsulation or representation of the security characteristics of the user). RACF uses the UTOKEN of the command issuer, the entity name and the access level of the command to determine if the command issuer is authorized to issue the command.

If SAF command authorization checking is not active, JES3 command authorization checking is used. JES3 command authorization checking uses the authorization level of the console issuing the command and the authorization level required to execute the command to verify whether the command should be executed.

5.7.1.3 Command Authorization Checking with MGCR(E)

MGCR(E) processing allows one or more MPF command exits to affect command processing. You specify on the MPFLSTxx parmlib member the MPF command exits. SET MPF operator command lets you dynamically change the MPFLSTxx member configuration and the exit specifications. Up to six command installation exits can be defined. These exits can:

- Change the text of commands.
- In a sysplex, change the destination of commands by routing them to a different system for execution.
- Modify a console′s authority to use a particular command. That is:
  - Authorize the command from a console that normally would not have the authority to issue the command
  - Reject the command from a console that normally would have the authority to issue the command
- Execute commands.
- Suppress commands.

5.8 JES3 5.2.1 Message Processing

As JES3 5.2.1 enables the MCS sysplex, MCS controls the flow of messages between systems in a JES3 sysplex. JES3 no longer transports messages to the global processor for display on MCS consoles. Support for the JES3-managed locally attached consoles is also removed. Message processing is determined by the following functions:

- JES3 issues all DSP messages through the WTO and WTOR macro service.
- MCS provides MPF processing on the originating system that issued the message.
- Prior to JES3 5.2.1, JES3 transported messages originating on a local JES3 processor to the global and reissued the messages (WTO) for display on MCS consoles attached to the global. Some JES3 installations rely on the JES3 global processor to be a “focal point” for MPF processing. To accommodate the “global-oriented” MPF processing, JES3 5.2.1 provides an option (GLOBMPF=YES on CONSTD initialization statement) to pass all messages routed to the global processor to MPF processing on the global. This is in addition to MPF processing on the originating system.
- Exits 69 and 70 allow you to implement JES3 environment-specific message processing in the global JES3 address space.
- Installation exit IATUX31 is deleted.
- The JES3 functional message routing is preserved.
- MCS ARMF for message retention is required as JMRF is deleted.

5.8.1 MPF Message Processing

Some JES3 installations have come to rely on the JES3 global processor to be a “focal point” for message processing and have implemented extensive MPF processing on the global for messages that originate on local processors. To accommodate “global-oriented” MPF processing that an installation may have, an option (GLOBMPF=) is provided on the CONSTD initialization statement.

```
CONSTD,EDIT=({escape,bkspace,,linedel}),
        GLOBMPF={YES|NO},
        SYN={(syn1,...syn6)|8}
        PLEXSYN={(syn1,...syn6)|*},
        CIFSS={FSSDEF|MSGROUTE},
        DLOG={ON|OFF}
```

Figure 39. Global MPF Processing on CONSTD Statement

GLOBMPF=YES Indicates that all messages routed by MCS to the global processor should also be made available to MPF processing on the global processor, in addition to MPF processing on the originating system.

GLOBMPF=NO The default. Indicates that messages routed to the global should not be presented to MPF on the global processor. In this case, MPF processing is only possible on the system that a message originates from.

It should be noted that the GLOBMPF option does not influence the routing of messages to the global processor. Installations wishing to use the GLOBMPF option must ensure that routing mechanisms are in place to guarantee the global is presented the proper set of messages. This could include the activation of DLOG which will result in the hardcopy message set being presented to the global, or the definition of a physical console or extended MCS console on the global that receives the proper set of routing codes (or all routing codes).

Note: Because JES3 5.2.1 does not transport messages to the JES3 global address space for display and logging purposes, two JES3 installation exits (exits 69 and 70) are provided to accommodate special message processing that is dependent on running in the JES3 address space and having access to JES3-maintained information.

5.8.2 User Exit 69

Exit 69 is called from the JES3 WTO SSI processing and allows you to examine a message and decide whether it should be sent to the JES3 global address space for additional processing. If the exit indicates that the message should be sent to the global, the WTO SSI sends the message through the JES3 SSISERV service. This exit is called on the system where the message is issued for all messages regardless of the origin of the message (for example: messages that originate on a local, messages that originate on the global, and messages issued by JES3 DSPs through the JES3 MESSAGE macro). In addition to calling exit 69, the WTO SSI processing continues to call IATUX57 to allow the installation to select a single routing code for JES3 message routing processing.

The following is a summary of the types of WTOs and WTORs that are presented to exit 69 and any special considerations that exist for the type of message. Flags in the exit parameter list indicate the type of request being passed to the exit:

Multi-Line WTO The first line of a multi-line WTO (the major line). Subsequent minor lines are *not* passed to the exit. Using the major line, the exit determines whether the message should be sent to the global.

If the exit determines that the message should be sent to the global, the major line and all minor lines are presented to exit 70 on the global. A separate call to exit 70 is made for each minor line.

Commands The text of an operator command.

WTORs The text of a WTOR message.

WTOR responses The full text of a WTOR response.

Exit 69 is managed through the MVS dynamic exit facility. JES3 defines this exit to the MVS dynamic exit facility with the name IAT_EXIT69. By default, JES3 does not define any exit routines to this exit. A sample exit module IATUX69 is provided in the SYS1.AJES3SRC data set.

You can use the EXIT statement of the PROGxx parmlib member, the SETPROG EXIT operator command, or the CSVDYNEX macro to control this exit and its exit routines. JES3 allows multiple exit routines to exist for this exit.

5.8.3 User Exit 70

Exit 70 is called in the global JES3 address space when the installation exit 69 has sent a message that requires further processing in the JES3 global address space. This exit cannot influence any routing, presentation, or retention attributes of the message.

Exits 69 and 70 allow you to implement JES3-specific processing in the JES3 global address space under CONSERV FCT. Note that the exits are not intended to be a replacement for MVS MCS message exits.

JES3 defines this exit to the MVS dynamic exit facility with the name IAT_EXIT70. By default, JES3 does not define any exit routines to this exit point. You can use the EXIT statement of the PROGxx parmlib member, the SETPROG EXIT operator command, or the CSVDYNEX macro to control this exit and its exit routines. JES3 allows multiple exit routines to exist for this exit. Note that the exit routines are *not* invoked with ASAVE linkage. A sample exit module IATUX70 is provided in the SYS1.AJES3SRC data set.

The pre-JES3 5.2.1 installation exit IATUX31 is deleted. This exit was called from the JES3 global address space prior to the display of a message on JES3-managed consoles and allowed many of the message attributes to be changed.

5.8.4 Functional Message Routing

JES3 functional message routing is currently a way to specify routing information for messages that pertain to functional areas, or for messages that pertain to a specific device. For messages issued by JES3 DSPs through the JES3 MESSAGE macro, the routing information is provided directly on the MESSAGE macro. For messages that are not directly issued by JES3 functions, the message routing is provided during the JES3 WTO SSI processing based on MSGROUTE initialization statement specifications.

An installation specifies the functional message routing information on the JES3 initialization stream parameters. *S1* and *S2* are JES3 destination classes that are the functional routing for messages issued for device B3A.

```
DEVICE,DTYPE=TA33490,JNAME=TAB3A,JUNIT=(B3A,MVS1,S1,OFF,B3A,MVS2,S2,OFF),
XTYPE=(D13490,TA),XUNIT=(B3A,MVS1,S1,OFF,B3A,MVS2,S2,OFF)
```

The routing information on the various JES3 initialization statements may be specified either as a JES3 console destination class or as a single MCS routing code. The continued support of destination classes ensures that existing initialization statements are accepted during migration to JES3 5.2.1.

5.8.5 JES3 Action Message Retention Facility

JES3 5.2.1 deletes the JES3 action message retention facility (JMRF) and the *I R commands used to display the JMRF messages. JMRF was used to maintain a queue of operator action messages. Messages retained by JMRF included messages issued by JES3 DSPs, messages issued by jobs, and messages issued by other subsystems and MVS components.

The MCS action message retention facility (AMRF) replaces the function provided by JMRF, and the MCS DISPLAY command must be used to display outstanding messages requiring operator action. AMRF also retains action messages issued when JES3 is not initialized, and action messages issued from programs running under the MASTER subsystem.

The default action messages that are retained by AMRF are messages with the following descriptor codes:

1 System failure

2 Immediate action

3 Eventual action

11 Critical eventual action

JES3 in previous versions retained messages with descriptor codes 1 or 2.

The MCS ARMF is activated through the ARMF(Y) setting on the INIT statement of the CONSOLxx parmlib member or through the CONTROL M,ARMF=Y operator command.

```
INIT  AMRF{(Y|N)
```

The message types retained by AMRF can be tailored using the MPFLSTxx RETAIN keyword. To be consistent with the JMRF message retention in pre-JES3 5.2.1 releases and have AMRF retain only messages with descriptor codes 1 and 2, include the following statement in your MPFLSTxx parmlib member:

```
.NO_ENTRY RETAIN(I)
```

Note: The sequence numbering for action messages changes when the MCS sysplex is active. Therefore, consider specifying a larger range on the K C command when deleting all outstanding action messages that the action message retention facility (AMRF) has retained, for example:

```
K C,A,1-999999
```

5.8.5.1 Display Action Message Commands

Outstanding WTOR requests are always retrievable using the D R command with JES3 5.2.1. Table 3 shows the deleted JES3 commands and the MCS command equivalents:

*Table 3. *I R and DISPLAY R Command Equivalency Summary*

Deleted JES3 Command	MCS DISPLAY	Equivalent Differences
*I R - Displays all outstanding action messages including WTORs. SETUP-related messages are not displayed.	D R,L	SETUP-related messages are displayed by the D R command. The D R command also displays messages that were issued before JES3 was fully initialized.
*I R,main - Displays all outstanding action messages associated with the named processor. Messages that are issued by a specific JES3 DSP via the JES3 MESSAGE macro are not displayed by this command.	D R,L,SYS=main	All messages issued from the named system are displayed by the D R command.
*I R,dspname - Displays all outstanding action messages associated with the named JES3 DSP.	D R,L,KEY=dspname	No major differences.
*I R,SETUP - Displays all outstanding action messages issued by the JES3 Setup DSP (for example, mount messages for a job).	D R,I,KEY=MOUNT	No major content differences. Change in the KEY name from prior releases.
*I R,SETUP,J=jobno - Displays all outstanding action messages issued by the JES3 Setup DSP for a specific job.	D R,I,KEY=MOUNT, JOB=jobname	The D R command uses job name rather than job number.
*I R,SETUP,C=Snn - Displays all outstanding action messages for the designated setup message destination class.	D R,I,ROUT=nnn	The route code equivalent of the JES3 destination class must be used on the display command.
*I R,J=? - Displays the job numbers of the three jobs with the most action messages currently being retained.	none	No equivalent display command
*I R,J=jobno - Displays the number of action messages currently being retained for the specified job	none	No equivalent display command
	D R,U	Return device numbers with unfulfilled mount requests and units requiring intervention are to be displayed.

5.9 JES3 5.2.1 Enhanced Console Processing User Considerations

The console destination block (CNDB) is a control block that encapsulates console routing information for a command or a message. JES3 5.2.1 intercepts JES3 commands from the SSI and saves the command origin including 4-byte console ID and related response processing information into a CNDB. CNDBs are used in MESSAGE macros to direct messages to a specific destination. CNDBs are also used in INTERCOM macros to identify the origin of an internally issued command. JES3 5.2.1 *INQUIRY and *MODIFY command processing flags the CNDBs to indicate that all messages in response to the command are to be marked as "command responses." When the JES3 MESSAGE to WTO Converter detects from the CNDB that a message is in response to a command, the DESC=5 parameter is included on the invocation of the WTO macro.

MCS uses 4-byte console IDs to identify a message receiver or command issuer. JES3 5.2.1 is also enhanced to support the 4-byte MCS console IDs.

MCS sysplex assigns symbolic console names for all MCS consoles. JES3 5.2.1 supports MCS console names in the command processing that are carried in the CNDB throughout the command processing.

JES3 5.2.1 continues to define console names for remote consoles through the JNAME keyword in the CONSOLE initialization statement.

JES3 5.2.1 supports the CART to insure accurate presentation of command responses to the extended MCS console. The CART is carried in the CNDB throughout the command processing.

5.9.1 Console Destination Block

The console destination block (CNDB, mapped by IATYCNDB) is a control block that maps and reserves space for origin information for a command, or destination information for a message. Only JES3 console support can directly reference fields in this control block. DSPs which have to copy or update fields in the CNDB should use the IATXCNDB macro service. The IATXCNDB macro can be used both inside and outside the JES3 address space. Most JES3 control blocks that contain console destination information are changed in JES3 5.2.1 to include a CNDB to hold the information.

The IATXCNDB macro allows you to:

- Initialize a console destination block by supplying the console name, console identifier, routing code mask, command and response token (CART), and command indicator.
- Transfer a copy of the specified console destination block to a specified area.
- Update the console identifier, routing code mask, command and response token (CART), and command indicator of the specified console destination block.
- Set to 0 (reset) the console identifier, routing code mask, command and response token (CART), and command indicator of the specified console destination block.
- Verify that the specified console destination block is at the correct level for the current JES3 release.
- Extract the console identifier, routing code mask, command and response token (CART), or command indicator of the specified console destination block.

JES3 5.2.1, during the MCS command processing SSI phase, intercepts JES3 commands, saves the command origin including 4-byte console ID and related response processing information into a CNDB, and finally routes (SSISERV) the command and the CNDB to the JES3 address space.

5.9.2 MESSAGE Macro Changes

The JES3 MESSAGE macro is used by DSPs to issue operator messages (action messages, command responses, and informational messages). In JES3 5.2.1, the following changes are made to the MESSAGE macro parameters:

CNDB= This parameter is added to specify the CNDB, which contains routing and destination information for the message. The CNDB also contains an indicator as to whether the message being issued is a command response and the CART of the command.

MLWOLST= This parameter is added and specifies the list of a multi-line message list (IATYMLWO).

ROUTCDE= This parameter is added and specifies the routing codes to be assigned to the message.

CONS= This parameter accepts a 4-byte console ID.

ROUT= This parameter may be specified at all times. Prior to JES3 5.2.1, ROUT= is only valid during JES3 initialization.

CLASS= This parameter may be specified at all times. Prior to JES3 5.2.1, CLASS= is not valid until JES3 initialization is complete.

BUSY= Certain BUSY return conditions are obsolete because of the deletion of JES3-managed consoles and JMRF.

PRTY= This parameter no longer affects the queuing of messages.

Note: The MPF=, DOMID=, and SYSNAME= parameters are deleted.

5.9.3 JES3 Multi-Line Messages

JES3 5.2.1 DSPs can issue true multi-line WTOs rather than a series of single line messages. Multi-line WTOs keep message lines together for display and provide definitive end line indication for a message.

Issuing a multi-line message in JES3 is a multi-step process:

1. Use a series of IATXMLWO REQUEST=BUILD,TEXT=text,LINETYPE=C|L|D,TOKEN=tkn to construct a multi-line message line-by-line.

```
IATXMLWO ...,TOKEN=0
ST   R1,MSG
IATXMLWO ...,TOKEN=MSG
IATXMLWO ...,TOKEN=MSG
IATXMLWO ...,TOKEN=MSG
```

2. Use MESSAGE MLWOLST= to issue the message:

```
MESSAGE MLWOLST=MSG,...
```

3. Use IATXMLWO REQUEST=CLEANUP to free resources associated with the message.

   The IATXMLWO multi-line WTO message setup macro has two functions for multi-line WTO message processing:

   - Building a single line of a multi-line WTO message

     An IATXMLWO macro invocation creates a IATYMLWO token that represents one line of a multi-line WTO message. These tokens are chained together. Once the message is complete, it is issued with the MESSAGE MLWOLST= macro. The MESSAGE macro processing converts the message text elements into a multi-line WTO.

     Each IATYMLWO token is chained to the next IATYMLWO token in the chain for the multi-line WTO message. This is done by passing in the IATYMLWO token address of the first line of the multi-line WTO message through the TOKEN parameter. The TOKEN parameter must be zero for the first line of the multi-line WTO message. The address of the first IATYMLWO in the chain must be passed to the MESSAGE macro through the MLWOLST parameter to issue the multi-line WTO message. The number of lines in a multi-line WTO message is limited to 999.

   - Cleaning up a multi-line WTO message

     The IATXMLWO REQUEST=CLEANUP frees the storage of all the IATYMLWO tokens in a chain for a multi-line WTO message. The

IATYMLWO token address of the first line of the multi-line WTO message is passed in via the TOKEN parameter.

5.9.4 INTERCOM Macro Changes

The JES3 INTERCOM macro simulates operator input of a console command. The INTERCOM service is used by DSPs to internally issue commands. The INTERCOM macro processing enters the command into the system with an MCS MGCRE macro. As a result, all commands go through the MCS interface, which serves as the single command entry into the MCS sysplex.

In JES3 5.2.1, the INTERCOM macro continues to be the primary means for JES3 DSPs to enter commands into the MCS sysplex. Some changes have been made to the INTERCOM macro:

- An optional CNDB= parameter is added to the INTERCOM macro. It addresses a CNDB that contains a 4-byte console ID, a console name, and a CART of the command issuer. If the CNDB= parameter is not specified, the INTERCOM macro uses the default dummy CNDB in the TVTX.
- Since the console ID is included in the CNDB, the CONS= parameter is deleted. The BUFFER= parameter is also deleted.

The MCS SVC 34 processing invokes the command exits prior to issuing the SSI call for the command processing. If a command is issued through MGCRE with the user data indicating to bypass the authority check (originally specify CHK=NO in the INTERCOM macro), this indicator is passed to the JES3 SSI command processing and is carried over to JES3 command processing to indicate “bypass authority checking.”

5.9.5 DEQMSG Macro Changes

The DEQMSG macro has been changed to support the following requests:

- Remove a specific input buffer for an FCT
- Remove all input buffers for an FCT
- Delete a specific action message for an FCT
- Delete all action messages for an FCT
- Remove all input buffers and delete all action messages for an FCT

Note: The TYPE=CONSOLE option, the CONS= parameter, and the DOM= parameter have been deleted. An MCS DOM is implicitly issued for all action messages.

5.9.6 4-Byte Console IDs

A console ID identifies a command issuer or a message receiver. Prior to JES3 5.2.1, JES3 used 2-byte console IDs for its command and message processing. The 2-byte console IDs of JES3 locally attached consoles and remote consoles (BSC/RJP, SNA/RJP, NJE, and BDT) were assigned during JES3 initialization. The IDs were assigned sequentially based on the order of the CONSOLE statements in the initialization stream. All MCS 1-byte console IDs were converted to the JES3 2-byte console IDs that were used internally by JES3. On the global, JES3 could receive commands from MCS consoles with a 4-byte console ID. However, JES3 could not send command responses back to the issuing console unless the console had also a 1-byte console ID.

Because MCS uses unique 4-byte console IDs for all consoles in the entire sysplex, JES3 5.2.1 is enhanced to support 4-byte console IDs. JES3 5.2.1 can receive and respond to commands from any console in the sysplex. The 4-byte console ID is also accepted by the MESSAGE macro.

User DSPs must be updated to use 4-byte console IDs and CNDBs.

5.9.7 Symbolic Console Names

Prior to JES3 5.2.1, all JES3-managed consoles had console IDs and console names associated with them. The console name was defined in the JNAME field of the CONSOLE statement and was used in JES3 commands, messages, and MLOG/DLOG. MCS consoles did not have JES3 names and were referenced only by 1-byte console IDs. The console names in JES3 messages or MLOG/DLOG were in the form CN(1-byte_console_ID).

MCS sysplex assigns symbolic console names for all MCS consoles. MCS WTO, WTOR, and MGCRE macro services accept symbolic console names in lieu of console IDs as the destination of a message or the issuer of the command. JES3 5.2.1 supports MCS console names in command and message processing. The console name is carried in the CNDB throughout processing.

JES3 5.2.1 also continues to define console names for remote consoles through the JNAME keyword in the CONSOLE initialization statement.

5.9.8 Command and Response Token Support

MCS sysplex introduced the command and response token (CART) for extended MCS consoles to allow a command response to be associated with the command. The CART is supplied by the command issuer. The command processor includes the CART in the command response message(s), thus allowing the issuing program to uniquely associate a command instance with its response.

Note: Prior to JES3 5.2.1, JES3 did not have CART support.

In JES3 5.2.1, JES3 also uses the CART to insure accurate presentation of command responses to the NJE and RJP extended MCS console. The CART is carried in the CNDB throughout the command processing.

When a command with a CART is issued from an EMCS console, JES3 obtains the CART from the SVC 34 subsystem interface and saves it in a CNDB. Later, when JES3 command processors issue command responses through the MESSAGE macro, the macro points to the CNDB that contains the CART that associates the command response with the command. The MESSAGE macro service routine then issues a WTO macro with CART= parameter to present the message for MCS processing.

5.9.9 Command Response Support

Prior to JES3 5.2.1, JES3 did not provide any indication that a specific message was issued in response to a command.

JES3 5.2.1 *INQUIRY and *MODIFY command processing flags the CNDB to indicate that all messages in response to the command are to be marked in the MESSAGE to WTO conversion as “command responses.”

When the JES3 MESSAGE to WTO Converter detects from the CNDB that a message is in response to a command, the DESC=5 parameter is included on the invocation of the WTO macro. If descriptor code 5 is not specified, command responses are considered unsolicited messages. Descriptor code 5 causes command responses to be considered as solicited messages. Extended MCS consoles (such as a TSO operator session), which may filter the message reception by selecting solicited messages only, can also receive JES3 command responses. This also enables automated operations to identify messages that are command responses.

Appendix A. Sysplex Terminology

There are a number of terms dealing with sysplex that are used throughout the sysplex documentation. Some terms you may be familiar with, but others are new.

Sysplex
: A sysplex is a set of one or more MVS systems that are initialized into the same sysplex with the same sysplex name and are fully connected by XCF signaling services. They must also all be connected to the same couple data set, except in the case of an XCFLOCAL mode.

Parallel Sysplex
: When the sysplex includes a coupling facility, it is referred to as a Parallel Sysplex. This applies even when the signaling paths are not coupling facility channels.

Sysplex Timer
: Also referred to as an external time reference (ETR). The sysplex timer (IBM 9037) is an external clock used for synchronizing the TOD clocks of CPCs that form the sysplex. It is only required if the sysplex consists of more than one CPC. MVS parameters in the parmlib and MVS system commands use the abbreviation ETR when referring to the sysplex timer.

CTC
: A CTC is used for channel-to-channel connectivity. It is a form of direct connection between two processor channels. The channels can be on the same or different processors. An ESCON CTC channel can be connected only to an ESCON CNC channel on the same or several other processors.

CF Channels
: Coupling facility channels are high bandwidth fiber optic links that provide high speed connectivity between the coupling facility and CPCs that use the coupling facility. Coupling facility channels are directly attached between the CPCs and the coupling facility. The coupling facilities can be used in a sysplex for XCF signaling. When used, they can replace the XCF CTC signaling paths and the sysplex connectivity is simplified and easier to manage.

Coupling Facility
: The coupling facility (CF) is either a S/390 microprocessor CPC, or a PR/SM LPAR on an ES/9000, 711 (H5)-based CPC. It is the base for high performance multisystem data sharing among certain ES/9000 processors and within a cluster of S/390 microprocessors. MVS/ESA services are available to authorized applications, such as subsystems and MVS components, to use the coupling facility to cache data, exchange status, and access sysplex lock structures so that techniques conducive to high performance data sharing and fast failure recovery can be implemented.

Couple Data Set
: There are six types of couple data sets: *SYSPLEX, CFRM, SFM, ARM, LOGR* and *WLM*. These data sets are shared by all the systems that make up the sysplex and are necessary depending on the policies you define to help manage resources and the workloads for the sysplex. The sysplex couple data set is mandatory and contains XCF related data about the sysplex, systems, groups, members, and general

status information. The sysplex couple data set is required to run a sysplex in either multisystem or monoplex mode. Data in the sysplex couple data set cannot be combined with the policy definitions for the couple data sets.

XCF
The cross-system coupling facility (XCF) is the operating system component that controls members and groups, provides inter-member communications (exchange data, programs, and so on), and monitoring services for members.

XES
Cross-system extended services (XES) is an extension to the XCF component. XES provides the sysplex services that applications and subsystems use to share data held in the coupling facility. These services support the sharing of cached data and common queues, while maintaining data integrity and consistency.

CFRM Policy
The coupling facility resource management (CFRM) policy lets you determine how MVS is to manage the coupling facility resources. It resides in the CFRM couple data set.

SFM Policy
The sysplex failure management (SFM) policy determines how the system reacts to system failures and signaling connectivity failures, and how PR/SM reconfiguration actions are to be managed. You can define a number of policies in the SFM couple data set, but only one policy can be active.

ARM Policy
Automatic restart management policy defines how to process restarts for started tasks and batch jobs that have registered with automatic restart management.

LOGR Policy
System logger (LOGR) policy defines, updates, or deletes structure or log stream definitions.

WLM Policy
In the workload management (WLM) policy, you set out the service goals for your various workloads. This policy resides in the WLM couple data set.

Group (XCF)
An XCF group is a collection of related members.

Member (XCF)
An XCF member is a specific function (one or more routines) of a multisystem application that is joined to XCF and assigned to a group by a multisystem application. When used in conjunction with a sysplex, a multisystem application refers to an application that has distributed functions on the sysplex systems.

A.1 XCF Modes of Operation

There are three sysplex modes of operation. You specify the mode through the PLEXCFG= keyword in the IEASYSxx parmlib member. The sysplex mode can be changed only through an IPL.

The three modes are:

XCFLOCAL A system in XCFLOCAL mode runs as a single-system sysplex. This mode is the nearest equivalent to the way systems run in MVS systems prior to MVS/ESA SP Version 4. In the XCFLOCAL mode, it is not possible to have more than one system in the sysplex. XCF does not provide any XCF inter-system signaling services. However, multisystem applications (members of an given XCF group) can use signaling services to communicate (exchange data) between their distributed functions in the same system. The members can be in the same address space or in different address spaces. An XCFLOCAL mode system does not use couple data sets or XCF CTCs. GRS in global serialization mode is not required in this XCF mode.

MONOPLEX The system runs as a single-system sysplex. MONOPLEX mode is also a single-system sysplex mode in which XCF does not provide any XCF inter-system signaling services. Multisystem applications (members of an given XCF group) can use signaling services to communicate (exchange data) between their distributed functions in the same system. The members can be in the same or different address spaces. XCF does not allow any other systems to join this sysplex. The difference in configuration between XCFLOCAL and MONOPLEX modes is that MONOPLEX mode requires a SYSPLEX couple data set. The couple data set makes allows applications use XCF monitoring services for permanent status recording. Just as in XCFLOCAL mode, XCF CTCs are not used in the MONOPLEX mode. GRS global serialization mode is also not required for MONOPLEX mode. MONOPLEX mode might be chosen as part of an intermediate sysplex migration stage to familiarize operators and systems programmers with sysplex before installing CTCs, and to allow application programmers to experiment with multisystem applications.

MULTISYSTEM MULTISYSTEM mode is the multi-system sysplex mode. In MULTISYSTEM mode, the sysplex consists of one or more MVS systems. All systems in the sysplex must use the same couple data set, and if they are not all on one processor, they must all be connected to the same sysplex timer. At least two inter-system connections through CTC links or coupling facility structures are required between every pair of systems in the sysplex. Multisystem applications can use XCF signaling services to communicate among members in the same system, as well as among members that reside on different systems. Systems running in Multisystem mode must also activate GRS in global resource serialization mode (that is, they must specify either TRYJOIN, JOIN, or START on the GRS= keyword in IEASYSxx parmlib member).

A.2 COUPLExx Parmlib Member

```
/******************************************************************/
/*  MEMBER = COUPLE00                                             */
/*                                                                */
/*  DESCRIPTION = THIS PARMLIB MEMBER IS ACCESSED THROUGH THE      */
/*               IEASYSXX MEMBER BY SPECIFYING COUPLE=00           */
/******************************************************************/

 COUPLE SYSPLEX(WTSCPLX1)
        PCOUPLE(SYS1.XCF.CDS10)
        ACOUPLE(SYS1.XCF.CDS20)
        INTERVAL(85)
        OPNOTIFY(85)
        CLEANUP(30)
        MAXMSG(500)
        RETRY(10)
        CLASSLEN(1024)

 /* DEFINITIONS FOR CFRM POLICY */

 DATA TYPE(CFRM)
      PCOUPLE(SYS1.XCF.CFRM01)
      ACOUPLE(SYS1.XCF.CFRM00)

 /* DATASETS FOR SFM POLICY */

 DATA TYPE(SFM)
      PCOUPLE(SYS1.XCF.SFM00)
      ACOUPLE(SYS1.XCF.SFM01)

 /* DATASETS FOR WLM POLICY */

 DATA TYPE(WLM)
      PCOUPLE(SYS1.XCF.WLM00)
      ACOUPLE(SYS1.XCF.WLM01)

 /* DATASETS FOR LOGR POLICY */

 DATA TYPE(LOGR)
      PCOUPLE(SYS1.XCF.LOGR00)
      ACOUPLE(SYS1.XCF.LOGR01)

 /* DATASETS FOR ARM POLICY */

 DATA TYPE(ARM)
      PCOUPLE(SYS1.XCF.ARM00)
      ACOUPLE(SYS1.XCF.ARM01)

 /* CTC PATH DEFINITIONS FOR DEFAULT SIGNALING */

    PATHIN  DEVICE(4010,4020,4030,4040,4050,4130,4150)
    PATHIN  DEVICE(4018,4028,4038,4048,4058,4138,4158)
    PATHOUT DEVICE(5010,5020,5030,5040,5050,5130,5150)
    PATHOUT DEVICE(5018,5028,5038,5048,5058,5138,5158)

 /* Coupling facility PATH DEFINITIONS FOR DEFAULT SIGNALING */

    PATHOUT STRNAME(IXC_DEFAULT_1,IXC_DEFAULT_2)
    PATHIN STRNAME(IXC_DEFAULT_1,IXC_DEFAULT_2)
```

Figure 40. SYS1.PARMLIB Member COUPLExx Example

Appendix B. Sysplex Evolution

Many of the MVS functions have been enhanced over the years to make use of XCF and XES services to improve the useability and operability of the system in a sysplex environment. JES3 in its most current release intended for a given MVS release coexists and supports MVS functions and enhancements where applicable.

The following sections summarize the MVS enhancements in chronological order.

B.1 MVS/ESA SP Version 4 Release 1

MVS/ESA SP Version 4 Release 1 includes following functions and enhancements:

- Cross-system coupling facility (XCF)

  This provides the MVS base services that allow programs on MVS systems in a multisystem environment to communicate with programs on the same system or other MVS systems. If there are two or more processors, they must:

  - Be connected by one or more IBM 3088 Multisystem Channel Communication Units or ES Connection (ESCON) channels operating in CTC mode.
  - Use the IBM Sysplex Timer (9037) hardware facility
  - Share an XCF couple data set.

- Hardware configuration definition (HCD)

  This provides an alternative to the MVS configuration program (MVSCP) for defining the I/O configuration. Either MVSCP or HCD can be used with MVS/ESA SP 4.1 to define the I/O configuration to the system. However, in a system that requires the dynamic I/O configuration capability, HCD must be used to define the configuration.

- Dynamic I/O reconfiguration

  This is the ability to select a new I/O reconfiguration definition without needing to perform a power-on-reset (POR) of the hardware or an initial program load (IPL) of the MVS system. Dynamic I/O reconfiguration allows the installation to add, delete, or modify the definitions of channel paths, control units, and I/O devices to the software and hardware I/O configuration.

  Note: Dynamic I/O reconfiguration is not supported for JES3-managed devices.

- External time reference feature (sysplex timer support)

  This provides a source for time information that enables an installation to set and maintain synchronization of the time of day (TOD) clock in each processor in a central processing complex (CPC) without operator intervention. The support also includes services to applications, subsystems, and operators for determining whether the clocks are synchronized with an external time reference (ETR).

- Enterprise systems connection (ESCON) architecture

MVS/ESA SP 4.1 supports the ESCON architecture and the ES Connection Manager Program Product.

- Other MVS/ESA SP 4.1 improvements:
  - Initialization changes

    Generalized nucleus loader, changes to the construction of the link pack areas, ALLOCxx parmlib member for specifying allocation defaults, and EXITxx parmlib member of SYS1.PARMLIB for specifying the names of the installation exit routines for certain allocation functions.

  - Enhancements to global resource serialization

    Use of XCF signaling services (instead of its own CTCs) for all communications when the GRS complex matches the sysplex, dynamic resource name list (RNLs) updates without having to reIPL the complex, and dynamic RESMIL tuning.

  - Enhancements to multiple console support (MCS)

    Improve the operation of a sysplex and extend the use of MCS consoles in any single or multi-system environment. An installation can define more than one console with master console authority, defining extended MCS (EMCS) consoles that can issue MVS and subsystem commands and can receive message output. MCS in MVS/ESA SP 4.1 allows the operator to control all systems in a sysplex from a single console or from a single EMCS console.

    Note: This is not supported by JES3 until JES3 5.2.1.

    Operator commands are routed in a sysplex either with the MVS ROUTE command or with a command prefix. Application programs can use the MCS CPF macro service to define command prefixes that allow the operator to direct commands.

    Note: This is not supported by JES3 until JES3 5.2.1.

    In MVS/ESA SP 4.1, an installation can specify one or more installation exits that affect the processing of a command. The exits may permit a console to issue the command, may modify the command text, or may execute or reject the command.

    The enhancements to the message presentation to console screens improve usability for console operators include the following:

    - Wrap mode

      MCS consoles support wrap mode in addition to roll and roll-deletable modes. The wrapping message display is similar to that found on a JES3 console screen.
    - Hold mode

      MCS consoles now support a “hold mode.” This mode enables an operator who is using the MCS console to temporarily suspend or resume the display of new messages on the screen. This allows the operator to view important messages before they are rolled off the screen.
    - Roll time

      1/2 and 1/4 second roll times are also supported.
    - Message display

System programmers can write message processing facility (MPF) exits to extend message highlighting and to alter the display format of a message.

– Enhancements to job control language (JCL)

  Several new and updated statements and parameters are included:

  - Job run time limit

    The maximum job run time limit, specified on either the JOB or EXEC JCL statement, is increased from 1439 minutes (1 minute shy of 1 day) to 357912 minutes (approximately 249 days).

  - IF/THEN/ELSE/ENDIF

    Statements allow users to conditionally execute job step(s) within a job. You can use IF/THEN/ELSE/ENDIF instead of the COND parameter on the JOB or EXECUTE statement.

  - JCLLIB

    Statement provides support for user procedure libraries. JCLLIB identifies private library names and specifies the search order for the libraries.

  - INCLUDE

    Statement specifies a member of a partitioned data set (PDS) to be included in the JCL stream. The PDS must be the system procedure library, an installation-defined procedure library, or a private library.

  - SET

    Statement allows you to define and assign initial values to parameters in JCL statements, or to change those values by assigning new ones.

  - Nested procedures

    These allow a procedure to invoke another procedure. MVS/ESA SP 4.1 supports up to 15 levels of nested procedures.

  - COMMAND

    This statement allows users to specify both MVS and JES commands in JCL. The commands are issued when encountered in the JCL stream.

  - System defined symbolic parameters

    JCL supports the system-defined symbolic parameters. &SYSUID is available in JCL to identify the user associated with the job.

  - OUTPUT

    This statement specifies processing options for a SYSOUT data set. New keywords on the OUTPUT statement assist in defining the distribution characteristics of the sysout data set: ADDRESS, BUILDING, DEPT, NAME, OUTDISP (not supported for JES3), ROOM, and TITLE.

    As further support, the following text unit identifiers are supported on the dynamic output OUTADD macro: DOADDRES, DOBUILD, DODEPT, DONAME, DOOUTDB, DOOUTDC, DOROOM, and DOTITLE.

  - SYSOUT

New keywords on the SYSOUT DD statement allow the user to specify when the data set is to be made available for printing:

- SEGMENT specifies the number of pages produced for a sysout data set before the segment of the data set is processed. This is available only for JES2; it is not supported for JES3.
- SPIN specifies when the data set is to be made available for printing, either immediately upon unallocation, or at the end of the job.

  The DYNALLOC macro supports the following new text unit identifiers for dynamic allocation, unallocation, and information retrieval: DALSPIN, DALSEGM, DUNSPIN, DINRSPIN, and DINRSEGM.

B.2 MVS/ESA SP Version 4 Release 2

MVS/ESA SP Version 4 Release 2 includes following functions and enhancements:

- Advanced program-to-program communication/MVS (APPC/MVS)

  This allows interconnected systems to communicate and to share processing of resources and access to data. APPC/MVS is a VTAM application that integrates APPC in the MVS/ESA operating system. The APPC/MVS facility provides the MVS services that manage communications between transaction programs, on the same system or across systems application architecture (SAA) systems.

- Callable services for APPC/MVS

  These provide transaction program (TP) conversation callable services that TPs use to communicate in conversations. These services are based on existing SNA LU 6.2 verbs and common programming interface (CPI) calls. In addition, APPC/MVS provides a group of advanced TP-callable services that are specific to MVS and have no LU 6.2 or CPI equivalent, and some services that have no SNA or CPI equivalent, and are intended for authorized users only. For example, a user-written transaction scheduler must use these services to interact with APPC/MVS.

- Dynamic I/O configuration

  The initial support for dynamic I/O configuration was provided in MVS/ESA SP 4.1. With MVS/ESA SP 4.2, and using the hardware configuration definition (HCD), the full capabilities of dynamic I/O configuration are available for the ES/9000 processor unit 9021 and processor unit 9121.

- Other enhancements in MVS/ESA SP 4.2:

  – Hardware configuration definition allows an authorized user to define a hardware component (channel path, control unit, I/O device) as DYNAMIC. HCD also allows authorized personnel and programs to activate a configuration dynamically.

    HCD also provides following:

    - Additional display information about the configuration definition
    - Two utility functions for input/output definition file (IODF) initialization and backup. These utility functions are invoked by calling HCD in batch mode.

- Multiple Console Support extends the enhancements to multiple console support begun in MVS/ESA SP 4.1. Console switch processing is improved to include extended MCS consoles, to permit authorized users to dynamically switch the routing and other attributes of a console to another compatible console, and to allow console attributes to be restored to a previous state.

  The installation can also initialize and operate a single system or multi-system complex using the system console, extended MCS consoles, or both. A system console attached to an IBM ES/9000 processor unit 9021 or processor unit 9121 can initialize MVS without using an MCS console.

- SVC dump processing is changed to operate in two phases. The first phase is for capturing the data (that is, for dumping the data to system data spaces and then returning the dumped address space to a dispatchable state). The second phase is for writing the captured data. By separating the two phases, the amount of time that an address space is set non-dispatchable is decreased (to only that amount of time that the dump data is being captured for that address space). The system then can write multiple captured data spaces to different SYS1.DUMPnn data sets concurrently.

- System Resources Manager (SRM) and Real Storage Manager (RSM) provide support for block paging. Block paging is a method of grouping pages together and then paging them as one large block, rather than as individual pages. The system uses implicit block paging automatically. Explicit block paging can be used by an installation based on the known characteristics of an application (through an installation-provided reference pattern).

  The use of new and existing response time, turnaround, and work priority SRM parameters is simplified by reducing the number of parameters. Other SRM improvements include address space monitoring to ensure that the system is not hindered from running effectively due to a lack of available central storage, and installation controls for the use of expanded storage to allow installations to select on a domain basis whether a page is to be sent to expanded storage or auxiliary storage.

- Job Control Language-Several changes were made to existing JCL statements. Optional new keywords on the JOB statement provide controls to limit the volume of sysout data produced: BYTES, CARDS, LINES, and PAGES specify the maximum output to be printed for a job′s sysout data sets.

  New optional keywords on the OUTPUT statement provide additional processing specifications for sysout data sets: NOTIFY requests to send an AFP print completion message, and USERLIB specifies the names of user libraries that contain resources for Advanced Function Printing (AFP).

  JES3 support for dynamic output macros OUTADD and OUTDEL.

B.3 MVS/ESA SP Version 4 Release 2.2

MVS/ESA SP Version 4 Release 2.2 includes following functions and enhancements:

- Name/token callable services allow an application program to save and then retrieve 16 bytes of application-related data.
- Other enhancements in MVS/ESA SP 4.2.2 include:
  - APPC/MVS supports VTAM persistent sessions, which keep LU-LU sessions active during interruptions in APPC services.

    In MVS/ESA SP 4.2, asynchronous processing by APPC/MVS TP conversation call services was available to authorized callers only. In MVS/ESA SP 4.2.2, unauthorized as well as authorized callers can use some key services asynchronously.
  - Initialization improvements include a new NUCLSTxx parmlib member to specify the names of members of SYS1.NUCLEUS that are to be included in the nucleus at IPL time.
  - Sysplex message automation

    EMCS consoles can be defined to receive automation messages. On the WTO(R) macro descriptor, code 13 can be used to indicate that the message is to bypass automation processing.

B.4 MVS/ESA SP Version 4 Release 3

MVS/ESA SP Version 4 Release 3 includes the following functions and enhancements:

- APPC/MVS server facilities provide an alternative way of processing inbound transaction program requests. Address spaces (batch, started tasks, TSO/E users, or APPC/MVS initiators) can directly receive specific inbound requests, instead of requiring the transaction scheduler to process them. And, unlike scheduled TPs, these address spaces can receive multiple inbound conversations concurrently. With this capability, installations can write client/server applications to manage access to resources (such as data sets) required by the installation′s transaction programs. Address spaces that perform these server functions are called APPC/MVS servers.

  In addition, APPC/MVS provides LU 6.2 receive persistent verification which allows conversations to be started without having to reverify the password each time, once the initial password verification has been done.
- OpenEdition MVS extends support to the set of standards being developed as the Portable Operating System Interface (POSIX). This support offers:
  - Program portability (with support for POSIX.1 and POSIX.1a) across multivendor operating systems
  - A hierarchical file system in MVS (with support for POSIX.1)
  - Exchangeability between MVS and POSIX-organized data
  - A UNIX-like user interface (with support for POSIX.2)
  - Application threads (with support for a subset of POSIX.4a)
  - New commands and capabilities for interacting with TSO/E

  This support is superimposed on the basic strengths of MVS/ESA, both for existing MVS applications and for new POSIX-conforming applications and data.

OpenEdition MVS provides an implementation of, and specific conformance to, these approved or draft standards:

- POSIX.1
- A subset of POSIX.1a
- POSIX.2
- A subset of POSIX.4a
- Federal Information Processing Standard (FIPS) 151-2

The X/Open consortium includes POSIX.1 as a requirement for branding.

- ESCON multiple image facility

  When a processor complex has EMIF capability, and is running in LPAR mode, all logical partitions can access the same shared channel paths, thereby reducing the number of required physical connections.

- IBM 3495 tape library dataserver supports system-managed tape libraries, which are introduced in DFSMS/MVS 1.1.

- Dynamic authorized program facility (APF)

  An installation can now create and maintain the list of APF-authorized program libraries in a dynamic format through the PROGxx parmlib member.

- Storage tracking

  The common storage tracking function collects data about jobs or address spaces that own storage in the common service area (CSA), extended CSA (ECSA), system queue area (SQA), and extended SQA (ESQA). You can use the resource measurement facility (RMF) or interactive problem control system (IPCS) to format the data.

- Global resource serialization latch manager provides services that authorized programs can use to serialize resources within a single MVS system.

- I/O timing facility allows the installation to limit long I/O requests to DASD devices. The system abnormally ends I/O requests that exceed the limit. You can specify a time limit on a device basis through the IECIOSxx parmlib member.

- Stand-alone dump (SADMP) allows the dumping of storage to either a 3380, 3390, or 9345 direct access storage device (DASD). Dumping to DASD produces stand-alone dump output more conveniently, reducing the level of operator intervention.

B.5 MVS/ESA SP Version 5 Release 1

MVS/ESA SP Version 5 Release 1 includes following functions and enhancements:

- Cross-system extended services (XES) adds a new set of coupling services to those introduced in MVS/ESA SP Version 4. XES uses one or more coupling facilities to satisfy customer requirements for:

  - High-performance data sharing across the systems in a sysplex
  - Maintaining the integrity and consistency of shared data
  - Maintaining the availability of a sysplex

  A coupling facility is a special logical partition on certain ES/9000 and 9672 Parallel Transaction Server models. To allow the connected systems to share data within the coupling facility, new types of channels attach the coupling facility to MVS systems in a sysplex.

Storage in the coupling facility can be allocated as three different types of named objects called structures. Units of data in each structure are accessed as logical entities. The ability to access data in this manner frees applications from concerns about the physical location of data in the sysplex.

Each structure has specific attributes, assigned by the application that uses the structure. Structures can be used to store or cache data (a cache structure), to maintain information in a list or a queue (a list structure), and to serialize access to resources (a lock structure). The application using the structure must be an authorized program in the sysplex. The application manipulates the structure or data in the structure by issuing one of a new set of authorized macros.

XES in a sysplex with the coupling facility provides a cost effective way to increase workload data sharing. In MVS/ESA SP 5.1, both CICS/ESA and IMS/ESA can take advantage of coupling facility data sharing.

MVS/ESA SP 5.1 can also use structures in a coupling facility for XCF signaling. Signals are sent through the coupling facility, instead of channel-to-channel (CTC) connections, thus simplifying the planning and configuration tasks required to establish signaling in a sysplex.

To enhance the continuous availability of a sysplex, XES provides sysplex failure management (SFM). SFM allows an installation to predefine the actions that MVS is to take when certain types of failures occur in the sysplex. For example, if one system loses signaling connectivity to another system in the sysplex, SFM provides the method of reconfiguring the sysplex so that operations can continue. SFM allows an installation to specify a relative value to each system in the sysplex. Therefore, the most important systems in the sysplex can continue without operator intervention.

SFM also supports the Processor Resource/Systems Manager (PR/SM) automatic reconfiguration facility. It allows the installation to specify the PR/SM reconfiguration actions that are to be taken before a system is partitioned out of the sysplex.

MVS/ESA SP Version 4 introduced a sysplex consisting of multiple systems coupled together by hardware elements and software services. The number of systems supported by MVS/ESA SP Version 4 was from one to eight. MVS/ESA SP 5.1 enhances this multiple system support by allowing up to 32 systems in the sysplex.

- Workload management (WLM) simplifies the definition, control, and reporting of the performance requirements for workloads. It provides a direct way to specify performance goals for work.

  To reduce the complexity of managing system resources, MVS WLM provides goal-oriented dynamic resource management. Using WLM, an installation defines performance goals for CICS, IMS, JES, APPC/MVS, TSO/E, and OpenEdition MVS work on the basis of business importance. The objective of workload management is to attain these goals through dynamic resource distribution.

  An installation defines performance goals in a service policy. Service policies are defined through an interactive system productivity facility (ISPF) application, and they set goals for all types of MVS-managed work, online and batch. An installation can create multiple service policies to adjust performance goals for different periods of time.

The service policy applies to a sysplex, whether a single system or a multisystem sysplex. Each service policy has a name, and can be activated by the WLM ISPF application, or an operator command. Only one policy can be active at a time. When the policy is activated, all systems in the sysplex process to achieve the goals defined in the policy.

Workload management coordinates and shares performance information across the sysplex. Using this information, each MVS system handles its own system resource management and dynamically optimizes resources to work according to the goals defined in the service policy. During processing, the system monitors how well the goals are being met across the sysplex and adapts accordingly as the environment changes. If there is contention for resources, each system makes the appropriate trade-offs based on the importance of the work and how well the goals are being met. This way, all systems can cooperate to protect work that is critical to your installation.

Resource measurement facility (RMF) collects and combines system management facilities (SMF) data for the sysplex, and reports how well the sysplex is doing to achieve the goals defined in the service policy. In addition, execution delay information is available in SMF records that show where delays are occurring. An installation can use this information to help adjust the performance goals, focus on specific subsystems having a problem, or make work scheduling adjustments.

Enterprise performance data manager (EPDM) and service level reporter (SLR) combine sysplex performance data into a database allowing interactive after-the-fact reporting and identification of performance problems.

Although workload management strives to achieve performance goals for all work in a sysplex, it remains the responsibility of the individual subsystems to distribute work across the sysplex to satisfy those goals. To make the most of workload management, work needs to be properly distributed across the sysplex and within each system. An installation must use the routing and distribution mechanisms of VTAM, TSO/E, CICS, IMS, JES and OpenEdition MVS to distribute work properly. CICSPlex system manager (SM) is also available, offering a goal-oriented algorithm for dynamic transaction routing.

Workload management replaces the existing set of SRM controls (the IEAICSxx and the IEAIPSxx parmlib members). However, in MVS/ESA SP 5.1, you have the option of processing with your existing IEAIPSxx and IEAICSxx until you are comfortable with the workload management service definition. Processing with an existing IEAIPSxx and IEAICSxx is called compatibility mode. Processing towards goals defined in the workload management service definition is called goal mode. This way, you can switch over to processing towards a service policy according to your own pace.

- OpenEdition DCE Base Services MVS include the base DCE functions:
  - Remote procedure call
  - Directory service
  - Distributed time service
  - Threads service
  - Security service

  Optional features for OpenEdition DCE include:

- OpenEdition DCE base services MVS application support, CICS feature
- OpenEdition DCE base services MVS application support, IMS feature

This selection of services enables a closer coupling of IBM services, as well as enabling IBM systems to function as parts of a heterogeneous distributed system.

- Other enhancements in MVS/ESA SP 5.1:
  - Display RESERVE status

    When a start pending missing interrupt (MIH) condition occurs in a multisystem environment (including non-MVS systems), MVS automatically displays the central processor serial number of a failed system that is holding a device reserve. If the failed system is in a sysplex, the system name is included in the display. This enables an installation to distinguish between MIH conditions caused by long-held reserves and other, more complex, configuration problems.

  - Dynamic exits provide system control of multiple exit routines called for an exit, and allows updates to exit processing without an IPL.

    The SMF and allocation installation exits exploit this capability. Users can associate multiple exit routines with the SMF and allocation exits and control their use at IPL or while the system is running.

    Users can also use the dynamic exits capability to define their own exits and control the use of those exits within a program.

  - JOB support for started tasks allows users to assign job names to started tasks and associate job-level characteristics with the started task. For example, for a specific started task, users can now:
    - Specify accounting information. This helps to reduce the number of SMF installation exits (or other installation-written programs) that are required for specifying accounting information.
    - Control started task output. For example, the installation can now specify that output should always be generated for a started task when an abnormal disposition occurs. Also, all output from a started task need not go to a single destination; the installation can direct the output for each instance of a started task to a different place or person.
    - Use SYSIN data. This allows users to pass parameters and data to associated programs in a started task.

    Being able to assign job names to a started task can provide more granular security through RACF Version 2 (which provides dynamic security assignment, thereby avoiding IPLs needed to change the security of a started task).

  - Support for open systems adapter (OSA)

    The open systems adaptor (OSA) is a new hardware feature in the S/390 that allows direct attachment of LANs to the main frame (ES/9000). The OSA hardware appears to the system as a new type of channel path and control unit, and as two new types of I/O device. These new device types are mapped to the CTC device class so that they can be used by existing MVS software without impacting the application. One device type, OSA (open systems adapter), is used by the LAN communication; the other device type, OSAD (open systems adapter diagnostic) is used by the OSA manager to control the OSA configuration.

Definition of the OSA hardware to the I/O configuration is done by using HCD.

OSA support provides direct network LAN connectivity to S/390 processors/servers. OSA support also provides greatly improved host utilization through the efficient off-load protocol processing of TCP/IP.

– Multisystem management enhancements include the ability to:

  - Share a master catalog across multiple systems more easily, through system symbols and being able to specify unique and data set names for STGINDEX, LOGREC and pageing data sets.

  - Specify SMF data set names more easily, by removing the requirement that the SMF data sets include SYS1.MAN as a prefix, and by expanding the maximum length of the data set name to 44 characters. System symbols are also supported for SMF data set names.

  - Retrieve device description information more efficiently using the device self-descriptor services. Authorized applications can retrieve device-description information, such as serial numbers and model numbers, that uniquely identifies I/O hardware located along a specific I/O path.

  - Use the Hardware Configuration Definition (HCD) to specify definitions for 9672 parallel transaction server, coupling facility, enterprise system connection (ESCON) switch configurations and open system adapter channels and ESCON converter byte channels.

    MVS/ESA SP 5.1 enables the use of four-digit device numbers. This theoretically increases the maximum number of devices that can be defined to a single MVS image from 4,096 to 65,536. However, if an installation wants to limit the amount of virtual storage below 16 Mb, used for devices to the amount used in the previous release, it must limit the number of additional devices defined to approximately 1500.

    Four-digit device specification makes it easier for installations to use the same I/O definition files (IODFs) for multiple systems, and provides more flexibility for unique device numbers across a sysplex.

  - JES common coupling (JES XCF) services-Before MVS/ESA SP 5.1, a JES2 member of a multi-access spool (MAS) configuration communicated with another JES2 member only through the job queues and the checkpoint data sets, and JES3 members of a JES3 complex communicated with each other through channel-to-channel (CTC) hardware. With this release of MVS and JES Version 5, JES2 members can communicate with each other and JES3 members can communicate with each other using XCF services.

    The JES common coupling services provide a macro interface that enables enhanced communication between JES members of a sysplex. An installation can use the JES common coupling macro interface in JES2 and JES3 exit routines, or in JES3 dynamic support programs (DSPs) to send messages to any other JES2 or JES3 in the sysplex.

  - Subspace group facility prevents application programs from overwriting each other. An authorized program can assign a unique section of address space private storage to each program running in the address space. The programs can reference only the storage assigned to them, which prevents them from accidentally overwriting each other′s data and code. CICS Version 4 Release 1 uses the subspace group facility for this purpose.

- APPC/MVS increases the ability to diagnose errors in APPC/MVS transaction programs (TPs), supports conversation-level pacing allowing APPC/MVS TPs to send and receive large amounts of data without affecting the sending and receiving of data by other APPC/MVS TPs, and can use SCOPE=COMMON data spaces (also known as common area data spaces) to make data accessible to all programs in a system,

B.6 MVS/ESA SP Version 5 Release 2

MVS/ESA SP Version 5 Release 2 includes following functions and enhancements:

- Automatic restart manager provides improved availability for batch jobs and started tasks (called elements), by automatically restarting them after they (or the system they are running on) unexpectedly terminate. This function is available only in a sysplex environment, and only for MVS/ESA SP 5.2 systems. Automatic restart manager is fully functional with either JES2 5.2.0 or JES3 5.2.1 (or subsequent levels).

  Note: Limited automatic restart management function is available for systems running JES2 4.1 or JES3 4.2.1:

  – Single system restarts are possible for started tasks running under the master subsystem.

  – Restart groups running on a JES2 SP5.2.0 or JES3 SP5.2.1 (or subsequent level) system, in which all elements are started tasks running under the master subsystem, can be restarted on a JES2 4.1 or JES3 4.2.1 system.

- System symbols enhance the ability of systems in a sysplex to share system definitions. Systems can use the same commands, dynamic allocations, parmlib members, and job control language (JCL) for started tasks while retaining unique values where required through system symbols.

  You can have all the systems share the definition, and use a system symbol as a “place holder” for the unique value. When each system processes the shared definition, it replaces the system symbol with the unique value it has defined to the system symbol. If all systems in a multisystem environment can share definitions, you can view the environment as a single system image with one point of control. In addition, application or vendor programs can call the system substitution service to provide additional symbols, called user symbols.

  MVS/ESA SP 5.1 introduced support for system symbols in a limited number of parmlib members and system commands. MVS/ESA SP 5.2 enhances that support by allowing system symbols in:

  – Dynamic allocations
  – JES2 initialization statements and commands
  – JES3 commands
  – Job control language (JCL) for started tasks and time sharing option extensions (TSO/E) logon procedures
  – Most MVS parmlib members
  – Most MVS system commands.

If your installation wants to substitute text for system symbols in other interfaces, such as application or vendor programs, it can call a ASASYMBM service to perform symbolic substitution.

- System initialization:
  - Master JCL in parmlib

    An alternate version of the master scheduler job control language (JCL) can be specified in the parmlib.

  - Message suppression indicators (IMSI) provides new granularity for initialization message suppression. The new IMSI characters allow you to suppress any combination of the following system initialization messages:
    - Informational messages
    - Prompt for master catalog response
    - Prompt for system parameters response.

- Workload balancing across multiple servers in a sysplex

  MVS/ESA SP 5.2 supports work balancing across multiple servers in a sysplex for DB2 Version 4.1 distributed data facility (DDF). DDF work is automatically routed across multiple DB2 systems. In addition, MVS/ESA SP 5.2 assists DDF to break up complex queries into smaller work requests to be processed in parallel within a system. A customer can define goals for DDF work in the same way as for other environments in a workload management service policy.

- UCB virtual storage constraint relief allows you to specify that a UCB exist above 16 megabytes to conserve common virtual storage while maintaining compatibility with existing interfaces, such as access methods and services. This support enables you to increase the number of devices in your I/O configuration beyond the limits that existed before MVS/ESA SP 5.2. You can take advantage of four-digit device numbers without the limit caused by common storage constraints below 16 megabytes.

  Hardware configuration definition (HCD) is used to define the UCB location for a device to be below 16 megabytes or above 16 megabytes.

- Dynamic reconfiguration of XES

  The structure alter function allows XES to dynamically change the size of a coupling facility structure (to either expand or contract the structure) and to reapportion the use of storage in the structure between entries and data elements without disrupting its use by structure connectors. This new function, which enhances the continuous availability of coupling facility structure data, permits installations to continue running their applications while changes in the structure are being made.

- System logger is a set of services that allows you to manage log data across the systems in a sysplex. The log data is in a log stream that resides on both a coupling facility structure and DASD data sets. System logger also provides for the use of staging data sets to ensure that log data is protected from a single point of failure. A single point of failure can exist because of the way the sysplex is configured or because of dynamic changes in the configuration. Using system logger services, you can merge log data across a sysplex.
  - Logrec log stream

Logrec can be defined to system logger as a log stream to provide sysplex-wide single system image "logrec."

  - Operations log (OPERLOG) log stream

    The MCS hardcopy medium can be defined to system logger as a log stream to provide sysplex-wide single system image "syslog."

- Dynamic SSI allows installations to define and manage subsystems without requiring an IPL.
- Tape allocation

  The way allocation selects tape devices is enhanced to choose the optimal device to allocate. Allocation considers the type of request, unit information on the request, and the characteristics of each available tape device. Based on these factors, the optimal device is allocated. An installation might notice a difference in tape allocation for a particular job when migrating to MVS/ESA SP 5.2.

- Automatic tape switching allows an installation to have a category of assignable tape devices known as automatically switchable. These devices are not dedicated to any one system; rather they can be used by all systems in a sysplex that have connection to the device and are connected through a coupling facility. The system that receives a tape device request assigns an automatically switchable device without requiring that the device be taken offline from another system. In other words, "automatic" means that an operator is not required to enter a VARY command between uses of the tape drive among different systems in the sysplex. In some ways, automatically switchable tape devices have similarities to JES3-managed devices: they are online to all using systems, but can be allocated to only one at any given time. Automatic tape switching is not available for JES3-managed devices.

B.7 MVS/ESA SP Version 5 Release 2.2

MVS/ESA SP Version 5 Release 2.2 includes following functions and enhancements:

- OpenEdition MVS

  The OpenEdition shell is modeled after the UNIX shell. As implemented for OpenEdition MVS, this shell conforms to POSIX standard 1003.2, which has been adopted as ISO/IEC International Standard 9945-2: 1992. All optional facilities are supported, except these two User Portability Utilities: ctags and nm.

- Shared pages (IARVSERV) allow more control of the target area and allow users to change the type of storage access for already created target areas. MVS/ESA OpenEdition uses the new options for memory mapping of files. Many of the new options require the Suppression-on-protection (SOP) hardware feature.
- SOMobjects for MVS runtime feature is a runtime library that provides a set of functions used for creating objects and invoking methods on them. It contains:
  - The executable code for the System Object Model (SOM) kernel classes, Interface Repository (IR) Framework, and Emitter Framework.
  - An initial Interface Repository with information about the SOM kernel classes, IR Framework, and Emitter Framework.

– The IRDUMP utility for producing a formatted listing of IR files that displays IDL descriptions in a hierarchical manner.

The SOMobjects for MVS runtime library is also available with the SOMobjects for MVS program product, which contains an application development environment and the runtime library.

Appendix C. JES3 Version 5 and JES Common Coupling Services

This appendix provides information on JES common coupling services (JESXCF), XCF services, XES services, and gives an overview of the JES3 and the JES common coupling services interactions. These components provide the basis for a Parallel Sysplex environment. Figure 41 shows a multisystem JES3 environment and the relationship of the sysplex services provided by JESXCF, XCF, and XES. JES3 uses JESXCF services, which calls XCF to provide communication between the members of a MAS. The communication path is determined by XCF. XCF drives the CTC paths and calls XES to use the coupling facility as the communication path.

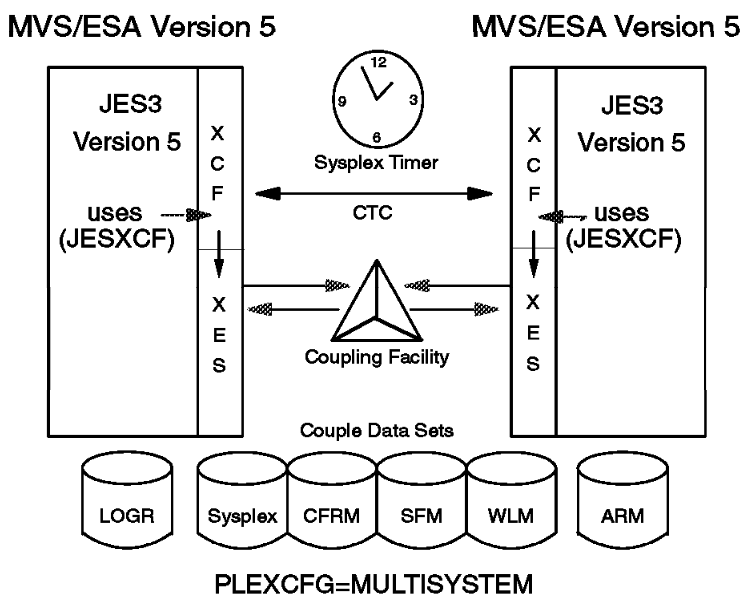


Figure 41. JES3 Multisystem Environment

Note: The general term "sysplex" is used in this appendix to denote any of the three sysplex configurations (XCFLOCAL, MONOPLEX, and MULTISYSTEM) that can be specified on the PLEXCFG= parameter in the IEASYSxx parmlib member. See Appendix A, "Sysplex Terminology" on page 87 for a description of the modes.

C.1 XCF Concepts

The MVS cross-system coupling facility (XCF) allows up to 32 MVS systems to communicate in a sysplex. XCF provides the MVS coupling services that allow multisystem application functions (programs) on one MVS system to communicate (send and receive data) with functions on other MVS systems. The multisystem application functions can:

- Define themselves to XCF and leave XCF
- Inform others of their status (for example, active or failed)
- Obtain information about the status of the other instances of the application
- Send and receive messages

In XCF terms, a specific function (one or more routines) of a multisystem application is a *member*. A member resides on one system in the sysplex and can use XCF services to communicate with other members of the same *group*. A group in XCF is defined as the set of related members defined to XCF by the multisystem application. A group can span one or more of the systems in a sysplex and represents a complete logical entity to XCF.

Once a member becomes defined, XCF associates the member with a specific task or job step task (if specified), an address space, and a system. The association of the member with a specific task, job step task, or address space is for termination (resource clean-up) processing only. When the task, job step task, or address space that a member is associated with terminates, XCF terminates the member, thus preventing the member from further use of XCF services. Note, however, that XCF services may be requested by all tasks and SRBs in the member address space as long as the member is active.

Members of XCF groups are unique within the sysplex. However, XCF allows you to define more than one member from the same task or address space, and have those members belong to different XCF groups. This option may be used if the number of members required exceeds the maximum 511 members in a group. Figure 42 shows the XCF group and member association.

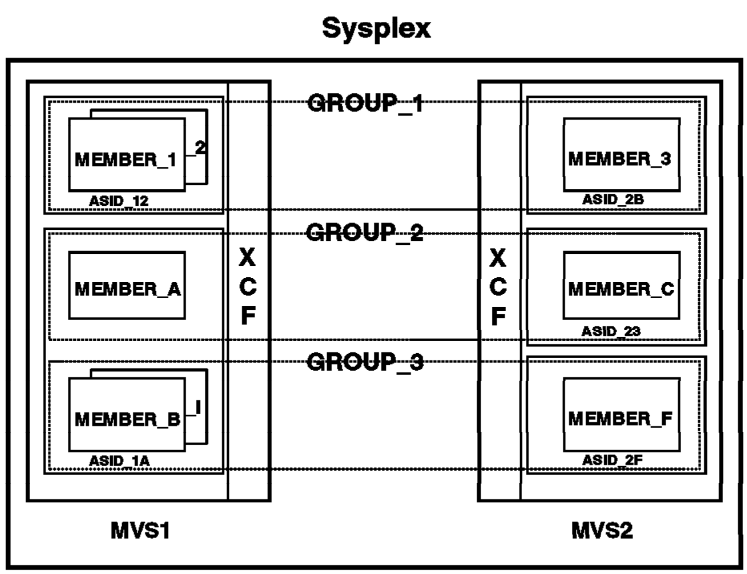


Figure 42. XCF Group and Member Relationship

C.1.1 XCF Services

The following services are provided by XCF:

- **Group Services**

  XCF group services provide ways for defining members to XCF, establishing them as part of a group, and allowing them to find out about the other members in the group. If a member identifies a group user routine, XCF uses this routine to notify the member about changes that occur to members of the group, or systems in the sysplex. With a group user routine, members can have the most current information about the other members in their group without having to query the system.

Specifically, XCF provides the following for setting up, making changes to, and obtaining information about groups and members:

IXCJOIN This macro service defines a member to an XCF group so the member can use the XCF signaling and status monitoring services.

IXCCREAT This macro service defines a member to XCF to be used later during execution.

IXCLEAVE IXCLEAVE, IXCQUIES, IXCDELET, and IXCTERM macro services disassociate members from XCF services. (IXCLEAVE and IXCDELET also disassociate a member from its group.)

IXCSETUS This macro service changes a member′s user state value (to be explained later in this appendix).

IXCMOD This macro service changes a member′s status-checking interval (to be explained later in this appendix).

IXCQUERY This macro service provides you with information about groups, members, and systems in the sysplex.

- **Status monitoring services**

  Status monitoring services are authorized services that notify member exit routines of changes to the status of other members of their group and the systems in the sysplex (the group exit routine). Members can request that XCF monitor their activity (the status exit routine).

  There are two kinds of monitoring services in XCF:

  – System status monitoring

    System status monitoring services monitor systems in the sysplex. The monitoring function uses the system status fields, which are periodically updated. All active application group exits receive control if a system fails to update its status field within a defined interval.

  – Member status monitoring

    Member status monitoring lets a member actively monitor its status. The services are optional, but if a member does use the services, then it has to update its status field periodically. The member status exit gets control if the member fails to update its status field within a specified interval. When the member status exit confirms that the member′s state changed, XCF notifies other members about the change through the group exit routines.

  XCF status monitoring services provide a way for members to actively participate in determining their own operational status, and to notify other members of their group when that operational status changes.

  The concept of permanent status recording is closely related to both member states and user states. When a member has permanent status recording, XCF maintains a record of the member′s existence (including the member′s current member state and user state values) even when the member is dormant or has failed. For permanent status recording to be available, an XCF configuration with a couple data set is required.

- **Signaling services**

  Signaling services include authorized assembler macros and an exit routine that members of a group use to communicate with other members of their group (macros IXCMSGO and IXCMSGI, and the message exit routine).

  The signaling services control the exchange of messages between members. The sender of a message requests services from XCF signaling services. XCF uses storage areas to communicate between members in the same system, and it uses the signaling paths to send messages between systems in the sysplex.

  XCF signaling services are the primary means of communication between members of an XCF group. Members may send messages to each other as follows:

  – A member sends a message by invoking the IXCMSGO macro service. Note that XCF does not necessarily deliver messages in the order in which they were sent.

  – To receive the message, the receiving XCF member must be active and must have provided a message user routine. XCF gives control to the message user routine under an SRB. The message user routine receives the message by invoking the IXCMSGI macro service.

Prior to MVS/ESA SP 5.1, the message data had to be sent from and received into a single buffer. In MVS/ESA SP 5.1, a message may consist of message data in a single buffer or in multiple buffers. Message data can also be received into a single buffer or multiple buffers.

Note: XCF services are available only to authorized assembler programs. See *MVS/ESA Programming: Sysplex Services Guide* for more information.

C.1.2 XES Services

Sysplex services for data sharing (XES) allows subsystems running in a sysplex to have data sharing by using a coupling facility. XCF uses XES services when the XCF signaling is defined to go through a coupling facility list structure. JES3 uses XCF signaling services indirectly through JESXCF and XCF:

- Through a coupling facility, when the XCF signaling is directed to use the coupling facility instead of XCF CTCs
- Through XCF CTCs, when the XCF signaling is directed to use XCF CTCs instead of a coupling facility

XCF calls XES services, as shown in Figure 43 on page 111, when the path of communication is a coupling facility instead of CTCs. The communication path is determined by the PATHIN and PATHOUT definitions. The definitions for PATHIN and PATHOUT are initially defined in the COUPLExx parmlib member and can be modified by the SETXCF operator command.

JES3 systems use XES services when the signaling paths used by XCF are a structure in a coupling facility.

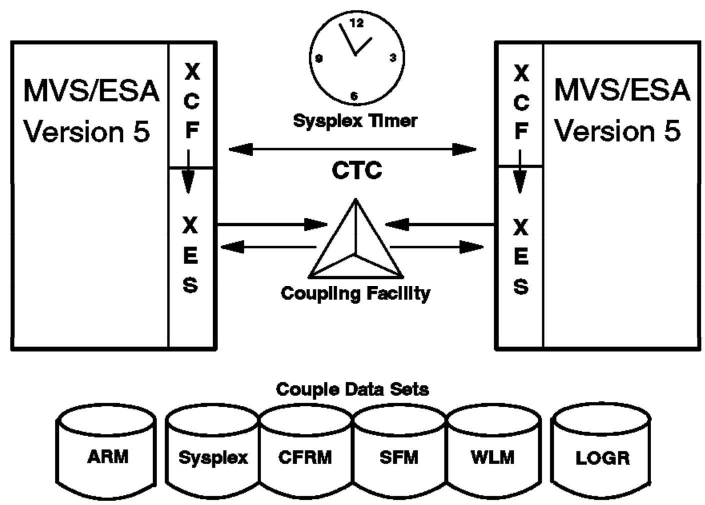


Figure 43. Sysplex Service for Data Sharing

C.2 JESXCF Concepts

The JESXCF address space is used by JES3 to exchange data with other JES3 members in a JES3 complex. This allows JES3 members to communicate using JESXCF. JESXCF uses XCF to do the communication. XCF has the choice of using defined paths through either CTCs or the coupling facility, as shown in Figure 44.

JESXCF is an XCF multisystem application and provides, based on XCF coupling services, common inter-processor and intra-processor communication services for both JES3 and JES2 subsystems. The JESXCF services are tailored to meet JES specific requirements and are provided through macros that enable communication among JES members in a sysplex. The macros are available to either JES3 or JES2 and can be used *only in JES environments*, except for the send message service, which can be used in all environments. The following macro services are available:

IXZXIXAT Create an attachment to JESXCF

IXZXIXMB Build a mailbox

IXZXIXSM Send a message

IXZXIXRM Receive a message

IXZXIXAC Acknowledge receipt of a message

IXZXIXIF Request JES member information

IXZXIXMC Clear a mailbox of messages

IXZXIXMD Delete a mailbox

IXZXIXDT Detach from a JESXCF group

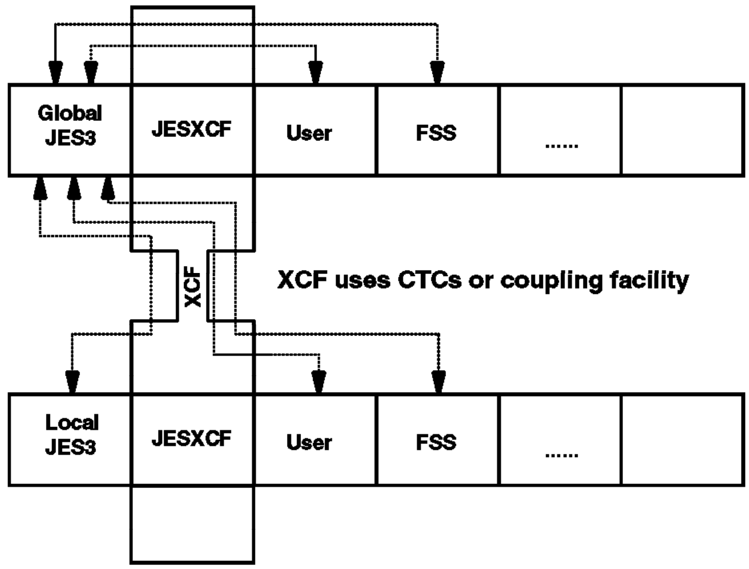


Figure 44. JESXCF Communication in a JES3 Environment

C.2.1 JESXCF in JES3 Environment

On each main in the JES3 complex, JES3 executes in its own address space as the primary subsystem of MVS. The JES3 address space on the global is responsible for maintaining information on:

- Jobs introduced into the system
- Scheduling jobs
- Allocation and deallocation of spool space
- Allocation and deallocation of JES3 managed devices
- Initialization and termination of functional subsystems

If JES3 or a user address space on a local, or a user address space on the global, requires information that only the JES3 global address space can supply, the requesting address space must be able to pass a request to the JES3 global address space for the information. An address space on a local or a user address space on the global uses the JES3 SSISERV macro to pass the request to the global. After the global has received and processed the request, the JES3 global address space passes the requested information back to the local. JES3 on the global uses the JSERV macro to pass information to a user or to JES3 on a local.

An address space on a local, or a user address space on the global, usually initiates communications. However, the JES3 global address space can initiate communication to another JES3 address space or to an FSS address space on a local.

When a subsystem communication request is issued, it can be processed on the same main or passed to another main. If an address space issues a request or a response, and the address space that receives the information resides on the same main, JES3 uses intra-processor communications. If an address space issues a request or response, and the address space that receives the information resides on another main, JES3 uses inter-processor communications.

The JES3 JSERV or SSISERV macro services provide communication between address spaces:

JSERV JES3 global address space issues a JSERV macro to:

- Send a response or information required by a user address space or a JES3 address space on a local
- Start the initialization of an FSS address space

SSISERV Subsystem interface modules and other subtasks of JES3 use the SSISERV macro to communicate with JES3. Parameters on the SSISERV macro are used to specify the following:

- Types of requests:

  WAIT The requestor waits for a response.

  REPLY The requestor waits or uses an exit routine to process the response.

  ACK The requestor receives an acknowledgement.

  COMM The requestor does not require a response.

  RESP The requestor is responding to an earlier WAIT, REPLY, or ACK type of request.

  EOMT The end-of-memory type is a special staging area cleanup request where the requestor wants to free all outstanding staging areas.

  PURGE A JES3 function uses JSERV or SSISERV to signal that a staging area (a message in JESXCF terms) has been completely processed and can be discarded.

- Function code of service provided by the JES3 global address space
- The main processor to which the request is to be sent

JES3 Version 5 uses the JES common coupling services as a replacement for its inter-processor CTC communication and intra-processor communication among JES3 mains processors, FSS address spaces, and user address spaces. JES3 continues to use the JSERV and SSISERV services for communication requests. Internally these requests are transformed to corresponding JESXCF requests to invoke the JESXCF services for the actual data transmission.

The JESXCF communication is in the form of messages. In the JESXCF context, a message is any data, including XCF events, that comprises a data packet. A data packet is the data that one JES3 member sends to itself or another member within the JESXCF group. The maximum size of the data is 60KB. If the quantity of data being sent is larger than 60KB, it can be broken into parts and transported as a multi-segment message.

JESXCF messages are received through mailboxes. A mailbox is a logical queue of ordered messages and is maintained by JESXCF in a location associated with a JES3 member. Messages are held in mailboxes until the receiving JES3 receives them or clears the mailbox. When a message is received, it is preceded by a message envelope. The message envelope is the header for the message and contains information that includes:

- The addresses of the sender and receiver of the message
- An identifier of the message type
- An offset to the actual message

JESXCF requires an acknowledgement for every message that is received. This is required so that JESXCF knows when it can free whatever resources it held for that message.

C.2.2 Create an Attachment to JESXCF (IXZXIXAT)

Before JESXCF messaging (communication) can be used by a JES3 member, the member has to create an attachment to a JESXCF group. During JES3 initialization, each member attaches to a JESXCF group and identifies itself by a unique member name and a JESXCF group name representing the whole JES3 complex. Figure 45 shows how JES3 initialization processing relates to the JESXCF processing.

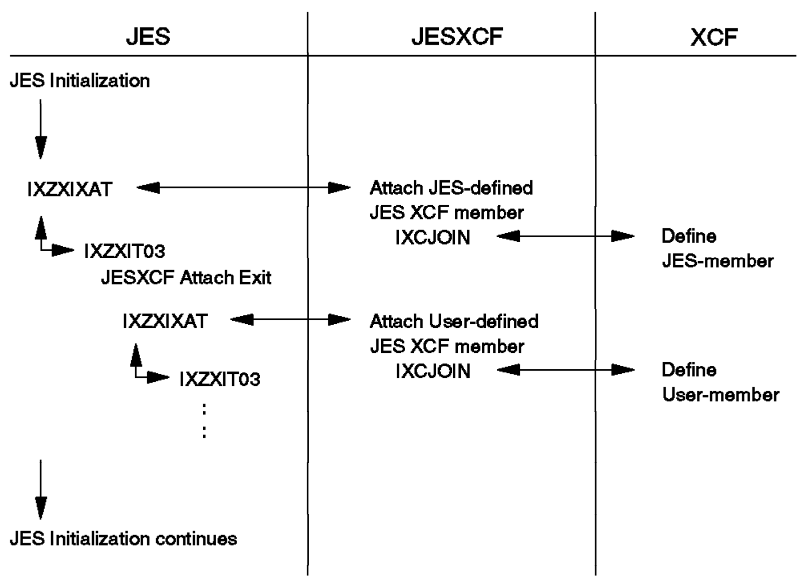


Figure 45. JES3 Initialization Process for JES3 5.1.1

Note: With JES3 5.2.1, the JESXCF address space is started before JES3 initializes.

For MVS/ESA Version 5.1 systems, during JES3 initialization or JES3 FSS connect processing, the JESXCF attach service is invoked (through the IXZXIXAT macro). The JESXCF attach service creates an address space for JESXCF (if it does not already exist) and attaches the JES3 member to a JESXCF group. During the JESXCF attach processing, the MVS XCF group services are invoked (through IXCJOIN) to create an XCF member for the JES3 member. JES3 uses as the XCF group name the XCFGRPNM= specification from the OPTIONS initialization statement if present. If the XCFGRPNM parameter is not specified, the home node name is used. The default name for the home node is N1. See 5.5, "JES3 Initialization Changes for Sysplex" on page 70 for specification of the XCF group name. JES3 uses as the XCF member name the main processor name defined on the NAME= parameter on the MAINPROC initialization statement. The JES3 FSSs use as the XCF member name the FSSNAME= specification on the FSSDEF statement.

Installations may create their own independent attachments to JESXCF. If an installation-defined JESXCF group is required, it should be created in the JESXCF attach/detach exit IXZXIT03. Exit IXZXIT03 receives control during the JESXCF attach/detach processing. Care should be taken not to inadvertently cause JESXCF to enter a recursive loop caused by the attach service calling the attach exit.

When JESXCF returns control after a successful attach JESXCF group request, a default mailbox (named SYSJES$DEFAULT) is supplied. This mailbox collects system event data. System event data includes any XCF events on any JES3 member of the JESXCF group and events on the MVS system under which it runs. JES3 does not use the default mailbox and deletes it. The default mailbox is deleted by first attaching to it using the IXZXIXMB service, and then deleting it through the IXZXIXMD service.

Note: Building the SYSJES$DEFAULT default mailbox at this time allows the collection of data while the member is waiting for initialization processing to complete. All messages sent to the default mailbox must be processed and acknowledged. If the default mailbox messages are not processed, JESXCF is not notified to clean up resources held for each message sent to it. This can eventually cause an “out of buffer” condition.

A successful attach JESXCF group request also returns a JESXCF group token that identifies the XCF group. JES3 Version 5 saves the group token in the SVTJXGT field in IATYSVT and TVTXJXGT field in IATYTVTX. A JES3 Version 5 FSS saves its group token in the FSCBJXGT field in IATYFSCB. All subsequent JESXCF service requests require this group token as input.

MVS XCF permanent status recording is used for JESXCF members if the sysplex configuration permits it.

C.2.2.1 Attach/Detach Exit (IXZXIT03)

Exit IXZXIT03 receives control during the IXZXIXAT JESXCF processing, and it may obtain a data area for later IXZXIT01 and IXZXIT02 processing, or it may provide an additional attachment to JESXCF. The IXZXIT03 exit receives control after the JES3 member has joined an XCF group but before either IXZXIT01 or IXZXIT02 receives control.

During detach processing, IXZXIT03 receives control to free the data area obtained during the attach processing or to drop the attachment from JESXCF. After IXZXIT03 receives control, neither IXZXIT01 nor IXZXIT02 receive control for the XCF group from which JES3 is detaching.

Exit IXZXIT03 communicates with JESXCF by setting flag bits and changing data fields in the parameter list passed to the exit.

C.2.3 Build a Mailbox (IXZXIXMB)

JES3 functions that intend to receive messages must build a mailbox (a logical queue of ordered messages that is maintained by JESXCF). This also implies that all JES3 functions that are to receive the JES3 global initiated communication (for example, FSS order processing) must create a mailbox. When a JES3 function retrieves and acknowledges a message, JESXCF removes that message from the mailbox. A mailbox may also be cleared (IXZXIXMC) without receiving queued messages. When a mailbox is cleared, JESXCF acknowledges those messages it clears.

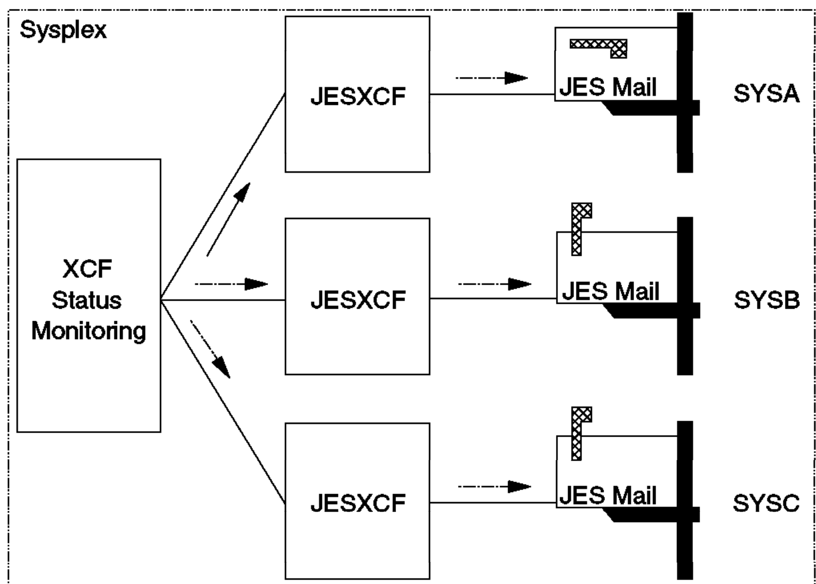


Figure 46. JES3 Mailboxes

Note: A message sender that requested an acknowledgement for messages (for example, a user address space issues SSISERV type ACK) is not required to create a mailbox to receive acknowledgements for the messages.

Mailboxes separate from the default mailbox are used by JES3 to collect messages and acknowledgements sent from and to JES3 functions (including user defined attachments). JES3 builds mailboxes in the following functions:

- FSS/FSA Listen Task - IATFCLT builds a mailbox to receive orders and post FSI requests from the JES3 global.
- Subsystem Communication Initialization - IATINM3 establishes the JESXCF attachment for the JES3 address space and processes the default mailbox by building it and then deleting it.
- Main Service DSP Driver - IATMSDR builds mailboxes that are used in the main connect processing. The mailbox names are formed by appending "SYSJES" with a main processor name.
- FSS/FSA Connect/Disconnect SSI - IATSICD establishes the JESXCF attachment for the CIFSSs and the WTR FSSs. IATSICD processes the default mailbox by building it and then deleting it.

- JES3 Subsystem Communication DESTQ Services - IATSSDS creates a JESXCF mailbox for the DESTQ entries. This is done by specifying as the mailbox name the name that is included in the destination queue entry prefixed by “SYSJES.” All IBM provided mailbox names are prefixed with “SYSRES.”

  The exceptions to this naming convention are the GMS-type and FSS-type destinations. The FSS mailbox names are constructed by converting the FSID into an EBCDIC value and appending it onto the end of the prefix “SYSJES.” The GMS mailbox names are created by appending the main processor name after prefix “SYSJESMS.” One mailbox for each main processor is created.

The build a mailbox request (IXZXIXMB) must identify a “post exit” routine. This post exit routine receives control under the task that issued the IXZXIXAT macro when JESXCF places a message into a mailbox. The exit routine posts the function that owns the mailbox and returns control to the calling program. This posting action notifies the function that there is a message to receive.

C.2.4 Send a Message (IXZXIXSM)

The JESXCF send a message service is the process of one JES3 member sending a data packet to the same or another JES3 function in the same JESXCF group. A send message request can be issued from a function running under a JES3 main task, a JES3 subtask, JES3 FSS address space, or a user address space. Once a message is sent, JESXCF delivers the message to the requested receiver determined by both the XCF member name and mailbox name. See Figure 47 on page 119 for an example of sending a message.

Unlike XCF, JESXCF maintains the order of the messages sent. All messages are received in the order in which they are sent. Multi-segment messages are not sent until the entire message (all segments) is available.

The message sender identifies the destination of the message by supplying the member name and the mailbox name of the receiver. This receiving member can be any member of the JESXCF group, including the sending member.

C.2.4.1 Message Types

Delivery and subsequent message processing is determined by the message type characteristics that are specified on the IXZXIXSM macro:

COMM This is the default request type for the IXZXIXSM macro. COMM indicates that the message being sent is communicated asynchronously, the sender does not request an acknowledgement, and JESXCF does not attempt multisystem recovery if either the sending or receiving member fails prior to message delivery.

This type is used when the message data is informational and there is no critical follow-on processing.

ASYNCACK Indicates that the message being sent is sent asynchronously and the sender requests an acknowledgement from the receiver once the message is processed.

This type is used when the message data is critical and the sender does not need to know immediately that the message data was received, but the sender requires to know that the message is processed successfully before any further processing can be done by the sending function.

ASYNC Indicates that the message being sent is sent asynchronously. This option is similar to REQTYPE=COMM, except that ASYNC requests that JESXCF provide recovery by resending the message if the receiving member fails prior to acknowledging the message.

This option is used when the message data is critical and the sender does not need to know that the message data was received but must ensure that JESXCF delivers the message to the receiver.

SYNC Indicates that the message is sent synchronously. The sender is placed in a wait state until an acknowledgement is received for the message. JESXCF provides recovery even if the receiving member fails prior to receiving or acknowledging the message.

This option is used when the message data is critical and the sender cannot continue until an acknowledgement (with data) is received.

JES3 Version 5 subsystem communication uses for the JSERV and SSISERV types the following JESXCF services:

WAIT IXZXIXSM request SYNC

REPLY IXZXIXSM request ASYNCACK

ACK IXZXIXSM request ASYNC

COMM IXZXIXSM request COMM

RESP IXZXIXAC (with data)

PURGE IXZXIXAC

C.2.4.2 Transport Exit (IXZXIT01)

Exit IXZXIT01 enables you to view, modify, add to, or reroute a message prior to the message being delivered to the receiving member. This exit gets control for the send message and the acknowledge message macro invocations.

Exit IXZXIT01 receives control under the same task that issues the IXZXIXSM or IXZXIXAC macros. JESXCF invokes the exit after a message packet has been created but before that packet is sent to its target member, and when the acknowledging member replies to the originating system for all acknowledgements except for the request COMM type message acknowledgements.

The data passed to this exit is in a data space; therefore, the data can be accessed through an access register. The exit receives control in access register address space control (AR ASC) mode. AR1 contains the ALET of the data space containing the exit parameters.

C.2.5 Receive a Message (IXZXIXRM)

Once a message is sent to a mailbox, the receiving member is informed through the POST exit routine that was specified on the POSTXIT= parameter of the IXZXIXMB macro. This posting action notifies the mailbox owner that there is a message to receive.

When the receiving function issues the receive macro, IXZXIXRM, the receive message exit (IXZXIT02) gets control first. This exit allows the receiving member to view or modify the message and to receive any extents that have been added to it by the transport exit. Once the message is received, the receiving function acknowledges the mail, informs JESXCF of its receipt, and if requested by the sending member, also returns acknowledgement data to the sender.

The receiver of a message must:

- Provide the group token that represents the JESXCF member and group
- Identify the mailbox from which to retrieve the message
- Indicate the type of message to be received:
  - Only system event messages
  - Only acknowledgement messages
  - Only JES- or installation-created messages

  Each segment of a multi-segment message is presented as a single message, and all segments are ordered as originally sent. However, individual messages from multiple members might not be ordered in the mailbox in a timestamp order.

The receiver may also request, using MSGFETCH=ALL, all messages that are in the mailbox. The messages are passed (as shown in the preceding section) by message type.

JES3 Version 5 subsystem communication services receive messages (staging areas) sent to the global and place them on a destination queue located off the destination queue table (DSQ). The destination queue that the staging area is placed on is determined by the FUNC parameter on the SSISERV macro. JES3 DSPs retrieve the queued staging areas through the DSQLOC macro service.

See Figure 47 for an overview summary of JESXCF message transmission and reception.

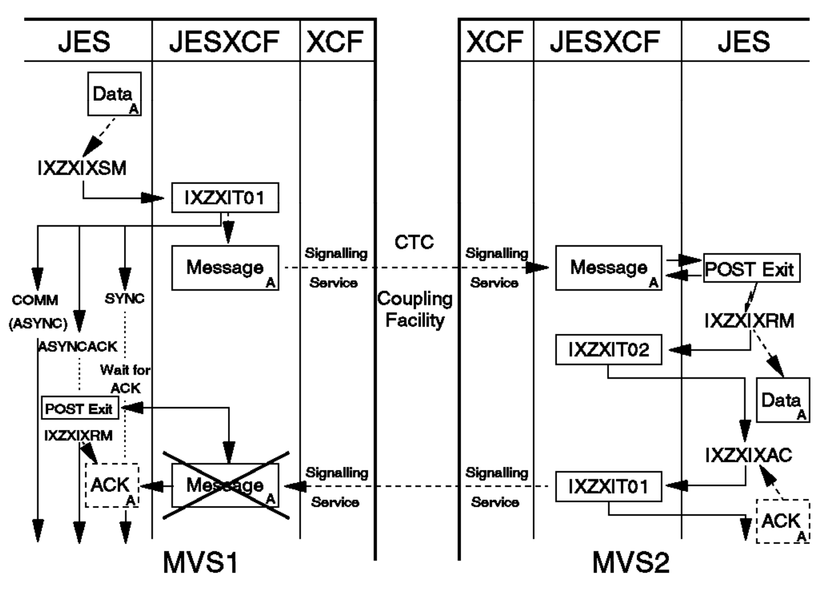


Figure 47. Summary of JESXCF Message Transmission and Reception

C.2.5.1 Receive Exit (IXZXIT02)

The IXZXIT02 exit enables you to view or modify messages before they are retrieved from a mailbox. The exit can also retrieve any message extents that the transport exit IXZXIT01 might have added. This exit gets control for IXZXIXRM (receive message) macro invocations.

The data passed to this exit is in a data space; therefore, the data can be accessed only through an AR/GPR pair. The exit receives control in AR ASC mode. AR1 contains the ALET of the data space containing the exit parameters.

C.2.6 Acknowledge Receipt of a Message (IXZXIXAC)

All received messages *must* be acknowledged to JESXCF independent of whether the sender requested an acknowledgement or not. After a message is acknowledged, JESXCF releases all resources held by the original message.

If the sender of the message requests an acknowledgement, the JESXCF returns an acknowledgement message as follows:

- To the mailbox specified on the ASYNCACK type send request as the mailbox into which the acknowledgement message is to be returned
- To the area specified on the SYNC type send request

An acknowledgement may contain message data and a return code that is to be returned with the acknowledgement. The data is made available as part of the acknowledgement message. The format and content of the data and meaning of the return code must have been agreed to by the sending and receiving JES3 functions.

C.2.7 Request JES Member Information (IXZXIXIF)

The request JES3 member information service (IXZXIXIF) returns a record of data about the specific group member, and also information about the MVS system on which the member is running. The data is returned either through a mailbox or is placed directly into the requestor′s data area in an array format. If the data is returned to the mailbox, JESXCF treats it as system event data.

The returned member information includes:

- Request token that was returned from IXZXIXIF service
- The release level of the JES product
- JESXCF maintenance level
- XCF group name
- XCF member name
- MVS system name that the JES is running on
- User state information as set by IXZXIXUS macro service
- XCF member token
- XCF sysplex token
- Member status:
  - Member is active, connection between the JESXCF address space and the JES address space is functioning.

- – MVS XCF state of the member is active but the connection between the JESXCF address space and the JES3 address space is not functioning. The probable cause is a JES3 ABEND.
  – Both MVS XCF status and JESXCF connection status indicate that the member is not active.
- XCF system token

C.2.8 Clear a Mailbox of Messages (IXZXIXMC)

The clear a mailbox service (IXZXIXMC) cleans the specified mailbox of all messages that are in the mailbox but have not yet been received. Every message that is removed from the mailbox is acknowledged to the sending JES3 member with an indication that the message was cleared but not processed by the receiver.

Typically, the clear mailbox service is used by a JES3 function that has restarted and does not want to process messages that were received from the previous JES3 start.

C.2.9 Delete a Mailbox (IXZXIXMD)

The delete mailbox service (IXZXIXMD) deletes a mailbox and frees any associated resources. All messages still in the mailbox are deleted, and appropriate acknowledgements are returned to the sender of the messages.

When JES3 terminates (except JES3 FSS), mailboxes are not deleted. However, when JES3 is restarted, the mailboxes are rebuilt with the same names they had prior to the termination.

C.2.10 Detach from a JESXCF Group (IXZXIXDT)

The detach from a JESXCF group service (IXZXIXDT) deactivates and removes a JES from an XCF group.

The JES3 FSS/FSA Connect/Disconnect SSI routine invokes IXZXIXDT during FSS disconnect processing. JES3 mains do not detach from the JESXCF group during JES3 termination.

C.2.11 JESXCF Message Types

Each message is prefixed by a message envelope, mapped by IXZYIXEN, for the following types of messages in a mailbox:

- System events (SYSEVENT)
- Acknowledgements (ACKS)
- Messages (MESSAGES)

Figure 48 on page 122 shows the JESXCF message mapping structures.

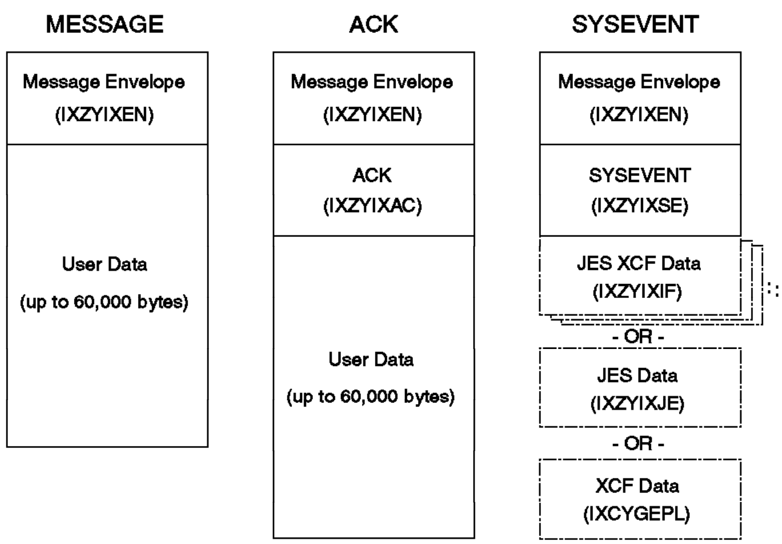


Figure 48. JESXCF Message Mapping

C.2.11.1 Message Envelope (IXZYIXEN)

The message envelope includes information such as:

- Status of the message in the mailbox:
  - Message has been re-sent to the receiving system because the receiving system was re-IPLed.
  - Message has been rerouted.
  - Message was present in the mailbox when the attacher disconnected.
  - Message has been received.
  - Message has been checkpointed.
- Maintenance level of the JESXCF component
- Message sequence number
- Address information of the receiver of the message (group name, member name, and mailbox name)
- Address information of the sender of the message (group name, member name, and mailbox name)
- Type of the send message request:
  - Synchronous message
  - Asynchronous message that does not return an acknowledgement message to the sender
  - Asynchronous message that returns an acknowledgement message to the sender
  - Asynchronous message that will not be re-sent to the receiver if the receiving system re-IPLs. No acknowledgement will be sent to the sender of the message
  - Acknowledgement message
  - Acknowledgement message sent because time specified on the IXZXIXSM macro has expired
- Single segment or multi-segment message indicators.

- Content of the message:
  - A system event
  - An acknowledgement message
  - Application message
- Length of the message not including the envelope

C.2.11.2 System Event (SYSEVENT)

The system event data includes three types of system event data:

JESXCF Data (IXZYIXIF) JESXCF data provides information about the JES and XCF connections, such as notification of an event detected by the JESXCF address space or such as termination of the connection between JESXCF and the JES address space. The data returned by the IXZXIXIF service is described under C.2.7, "Request JES Member Information (IXZXIXIF)" on page 120.

JES Data (IXZYIXJE) JES data provides a notification of events that the JESXCF address space has detected, such as termination of the connection between JESXCF and the JES address space. The JESXCF data includes:

- Event type:
  - Connection between JESXCF and specified JES terminated
- Group name of the member whose connection terminated
- Member name of the member whose connection terminated
- The request token for the message that timed out

XCF Data (IXCYGEPL) The XCF data is the same that is passed to the XCF group user exit of an active member. XCF data notifies about changes that occur to the XCF members of a group, or changes to the systems in the sysplex. See *MVS/ESA Programming: Sysplex Services Guide* for more information. The XCF data includes:

- Member data provided through the IXCJOIN that established the group exit
- Invocation event:
  - 1 - Member state event
  - 2 - User state event
  - 7 - Member status update missing reported by subsystem′s status exit
  - 8 - Member status update missing detected by subsystem monitor
  - 9 - Member status update no longer missing
  - 11 - System reported active
  - 12 - System update missing
  - 13 - System update resume
  - 14 - System reported going
  - 15 - System reported gone
  - 16 - System detected missing
  - 17 - System detected gone

  - 18 - System failure detection interval updated
- Member state before the action (listed above) was taken
- Member state after the action
- Member′s current status
- XCF group name
- XCF member name
- Member token
- System name where member is (was) active
- Time this event occurred
- User data

C.2.11.3 Acknowledgement (ACK)

An acknowledgement provides notification information on delivery of messages issued through the IXZXIXSM macro service. The acknowledgement data includes:

- Request token for the message that this acknowledgement is for
- Return code information returned by the receiving routine
- Length of the data returned to the sender through the IXZXIXAC macro service

C.2.11.4 User Data (MESSAGE)

JESXCF does not impose any format requirements on user data.

For more information about JESXCF, see *MVS/ESA Programming: JES Common Coupling Services*.

A sample JES3 DSP that uses JESXCF services is shown in Appendix E, "Sample Program Using JESXCF Services" on page 131. The DSP monitors system events from JESXCF. When an XCF "system reported gone" is received for a local JES3 main, the local is flushed (using the *START local_name FLUSH operator command). The DSP varies a local main online to the global (using the *VARY local_name ON operator command) when an XCF "system reported active" system event is received. The *VARY local_name ON is required because the flush processing varies the local main offline.

Appendix D. Coupling Facility for XCF Communication

To use a coupling facility, a couple data set must be defined which allows MVS to manage the coupling facility resources. This couple data set is called the coupling facility resource management (CFRM) and it contains what is called the CFRM policy.

This appendix shows the utilities that create the couple data set and defines couple data set policies. If you decide to use the coupling facility for XCF communication between MVS images and dedicate its use to JES3 global/local communication, the following steps may be necessary:

- Format CFRM couple data set - *IXCL1DSU*

  If a CFRM couple data set does not exist, one must be created and formatted. See D.1, "Format CFRM Couple Data Set."

- Define CFRM policy - *IXCMIAPU*

  A coupling facility structure must be predefined in the coupling facility resource management (CFRM) policy. See D.2, "Define Structure in CFRM Policy" on page 126.

- Define a transport class in the COUPLExx member.

  See D.3, "Modify the COUPLExx Member" on page 127.

  Note: IBM recommends using the XCF default transportation class for JES3 messages.

- Activate the policy

  See D.4, "Activate the CFRM Policy" on page 128.

  SETXCF START,POLICY,POLNAME=CFRM01,TYPE=CFRM

For a complete description of couple data sets, the utilities, and XCF signaling paths and their definitions, see *MVS/ESA Setting Up a Sysplex*.

D.1 Format CFRM Couple Data Set

The CFRM couple data set must be defined and formatted. The example in Figure 49 shows the format definitions for two CFRM couple data sets. Each CFRM couple data set should have an alternate defined.

```
//STEP1    EXEC PGM=IXCL1DSU
//STEPLIB  DD   DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
     DEFINEDS SYSPLEX(WTSCPLX1)
              DSN(SYS1.XCF.CFRM10) VOLSER(CDS001) MAXSYSTEM(16) CATALOG
          DATA TYPE(CFRM)
               ITEM NAME(POLICY) NUMBER(5)
               ITEM NAME(CF) NUMBER(2)
               ITEM NAME(STR) NUMBER(20)
               ITEM NAME(CONNECT) NUMBER(32)
     DEFINEDS SYSPLEX(WTSCPLX1)
              DSN(SYS1.XCF.CFRM10SP) VOLSER(CDS002) MAXSYSTEM(16) CATALOG
          DATA TYPE(CFRM)
               ITEM NAME(POLICY) NUMBER(5)
               ITEM NAME(CF) NUMBER(2)
               ITEM NAME(STR) NUMBER(20)
               ITEM NAME(CONNECT) NUMBER(32)
     DEFINEDS SYSPLEX(WTSCPLX1)
              DSN(SYS1.XCF.CFRM20) VOLSER(CDS002) MAXSYSTEM(16) CATALOG
          DATA TYPE(CFRM)
               ITEM NAME(POLICY) NUMBER(5)
               ITEM NAME(CF) NUMBER(2)
               ITEM NAME(STR) NUMBER(20)
               ITEM NAME(CONNECT) NUMBER(32)
     DEFINEDS SYSPLEX(WTSCPLX1)
              DSN(SYS1.XCF.CFRM20SP) VOLSER(CDS001) MAXSYSTEM(16) CATALOG
          DATA TYPE(CFRM)
               ITEM NAME(POLICY) NUMBER(5)
               ITEM NAME(CF) NUMBER(2)
               ITEM NAME(STR) NUMBER(20)
               ITEM NAME(CONNECT) NUMBER(32)
```

Figure 49. Format of a CFRM Couple Data Set

D.2 Define Structure in CFRM Policy

Coupling facility structures are defined through a CFRM policy, which is used to select a coupling facility, as shown in Figure 50 on page 127.

Note: An installation may have other structures to define besides the JES3 signaling structure.

If a CFRM policy already exists, you could modify it with the JES3 signaling structure and modify the COUPLExx parmlib member to specify a transport class for JES3.

Add the *IXC_JES3* structure definition to the CFRM job, as shown in Figure 50 on page 127.

```
//STEP1    EXEC PGM=IXCMIAPU
//SYSPRINT DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SYSIN    DD   *
     DATA TYPE(CFRM) REPORT(YES)
  /* DSN(SYS1.XCF.CFRM10) */
     DEFINE POLICY NAME(CFRM01) REPLACE(YES)
        CF NAME(CF01) TYPE(009672) MFG(IBM) PLANT(02) DUMPSPACE(2000)
           SEQUENCE(000000040104) PARTITION(1) CPCID(00)
        CF NAME(CF02) TYPE(009672) MFG(IBM) PLANT(02) DUMPSPACE=(2000)
               SEQUENCE(000000040104) PARTITION(1) CPCID(01)
        STRUCTURE NAME(IXC1_GRS) SIZE(1024)
               PREFLIST(CF01,CF02)
        STRUCTURE NAME(IXC1_DEFAULT)
               SIZE(4096)
        STRUCTURE NAME(IXC_CICS) SIZE(4096)
               PREFLIST(CF02,CF01)
        STRUCTURE NAME(IXC_JES3) SIZE(1024)
                PREFLIST(CF01,CF02)
```

Figure 50. Example of Modifying a CFRM Policy

The STRUCTURE statements in Figure 50 define the name, the size, and the location of the structure, where:

NAME The NAME is the one that is used to access the structure. This name must begin with *IXC* for XCF signaling.

SIZE The SIZE is the size of storage requested for this structure in 1K units.

PREFLIST The PREFLIST specifies a preferred coupling facility to contain this structure.

D.3 Modify the COUPLExx Member

To segregate the JES3 XCF signaling to a coupling facility list structure, modify the COUPLExx parmlib member for the next IPL.

Figure 51 shows the modified COUPLExx parmlib member with the JES3 signaling paths defined through a coupling facility.

```
/*****************************************************************/
/*  MEMBER = COUPLE00                                            */
/*****************************************************************/
 COUPLE SYSPLEX(WTSCPLX1)
        PCOUPLE(SYS1.XCF.CDS10)
        ACOUPLE(SYS1.XCF.CDS20)
        INTERVAL(30)
        CLEANUP(30)
        MAXMSG(500)
        RETRY(10)
        CLASSLEN(1024)

 /* DEFINITIONS FOR CFRM POLICY */
 DATA TYPE(CFRM)
      PCOUPLE(SYS1.XCF.CFRM10)
      ACOUPLE(SYS1.XCF.CFRM20)

 /* DATA SETS FOR SFM POLICY */
 DATA TYPE(SFM)
      PCOUPLE(SYS1.XCF.SFM10)
      ACOUPLE(SYS1.XCF.SFM20)

 /* DEFINITIONS FOR JES3 SIGNALING */
 CLASSDEF CLASS(JES3) GROUP(WTSCPLX3)
 PATHOUT STRNAME(IXC_JES3) CLASS(JES3)
 PATHIN STRNAME(IXC_JES3)

 /* DEFAULT CTC PATHS FOR SIGNALING */
 PATHIN  DEVICE(4010,4020,4030,4040,4050,4060,4070,4080)
 PATHIN  DEVICE(4018,4028,4038,4048,4058,4068,4078,4088)
 PATHOUT DEVICE(5010,5020,5030,5040,5050,5060,5070,5080)
 PATHOUT DEVICE(5018,5028,5038,5048,5058,5068,5078,5088)
```

Figure 51. COUPLExx Member of SYS1.PARMLIB

D.4 Activate the CFRM Policy

The following command is used to activate the CFRM policy:

```
SETXCF  START,POLICY,POLNAME=CFRM01,TYPE=CFRM
```

D.5 Activate the Signaling Paths

Once the updated CFRM policy is activated, JES3 XCF signaling can be dynamically switched to use the coupling facility structure with the following SETXCF operator commands:

```
SETXCF START,CLASSDEF,CLASS=JES3,GROUP=WTSCPLX3
SETXCF START,PATHOUT,STRNAME=IXC_JES3,CLASS=JES3
SETXCF START,PATHIN,STRNAME=IXC_JES3
```

When these commands complete, JES3 communication between processors is through the coupling facility.

If the coupling facility should fail, XCF automatically switches the JES3 signaling to the DEFAULT transportation class, which is defined through a set of CTC paths and allows JES3 communication between the processors to continue.

Resource Measurement Facility Version 5 (RMF) generates XCF activity reports. These reports include XCF usage by system, XCF usage by member, and XCF path statistics sections. You should use these reports to verify that JES3 XCF signaling performs properly.

Appendix E. Sample Program Using JESXCF Services

The JES3 local monitor DSP is activated with a *CALL dsp_name operator command and is terminated with a *CANCEL dsp_name command. The DSP may be tested using the TST DSP already defined in the IATGRPT DSP dictionary module. The TST DSP driver module has the name IATST00, and the data CSECT name is IATSTD0.

The installation of the JES3 local monitor consists of the IATUSMN and IATUSMD assemblies (the IATYUSM macro must be available in the assembly) followed by the link-edit of the objects into the SYS1.JES3LIB.

E.1 JES3 Local Monitor Driver - IATUSMN

Note: Source code for both IATUSMN and IATYUSM has statements flagged with @521 and @511. If you are installing this sample code on a JES3 5.1.1 system, delete the @521 lines and vice versa. The @521 and @511 lines show how a DSP could be changed to support the new console environment introduced in JES3 5.2.1.

```
USMN     TITLE 'IATUSMN - JES3 5.1.1 LOCAL MONITOR'
IATUSMN  AMODE 31
IATUSMN  RMODE ANY
         IATYASM
*-------
*
*        A sample JES3 DSP that uses JES XCF services to monitor
*        JES3 local systems.  The DSP monitors system events from
*        JES XCF.  When an XCF "system reported gone" is received for
*        a local JES3 main, the local is flushed (using the *START
*        local_name FLUSH operator command).  The DSP varies a local
*        main online to the global (using the *VARY local_name ON
*        operator command) when an XCF "system reported active"
*        system event is received.  The *VARY local_name ON is
*        required because the flush processing varies the local main
*        offline.
*
*        Link-Edit Attributes: RENT REUS REFR
*-------
IATUSMN  START 0
         PRINT NOGEN
* ====== If you are on JES3 511, delete the @521 lines          @521
**       TITLE 'IATYCNDB - CONSOLE DESTINATION BLOCK'           @521
         IATYCNDB DSECT=YES                                     @521
**       TITLE 'IATYCNS - CONSOLE BUFFER (INPUT MESSAGE) DSECT.'
         IATYCNS TYPE=(FCTQ,INPUT),CODES=YES
**       TITLE 'IATYDSP - DSP DICTIONARY DSECT.'
         IATYDSP
**       TITLE 'IATYEQU - SYMBOLIC MASK DEFINITIONS.'
         IATYEQU
**       TITLE 'IATYFSW - FAILSOFT WORK-AREA DSECT.'
         IATYFSW
**       TITLE 'IATYMPC - MAIN PROCESSOR DESCRIPTION.'
         IATYMPC
**       TITLE 'IATYREG - SYMBOLIC REGISTER DEFINITIONS.'
         IATYREG
**       TITLE 'IATYRSQ - RESQUEUE ENTRY DSECT.'
         IATYRSQ
**       TITLE 'IATYTVT - TRANSFER VECTOR TABLE DSECT.'
         IATYTVT
**       TITLE 'IATYSVT - SUBSYSTEM VECTOR TABLE FOR JES3.'
         IATYSVT
         PRINT GEN
         SPACE 3
         TITLE 'IATUSMN - JES3 LOCAL MONITOR'
*-------
IATUSMN  CSECT
         USING IATUSMN,R15         REG.15 ==> DRIVER ENTRY POINT
         USING TVTABLE,R12         REG.12 ==> TVT
         USING FCTSTART,R11        REG.11 ==> FCT
         IATYMOD BR=YES            MODULE ID.
         LR    R10,R15             ESTABLISH REG.10 AS BASE
         DROP  R15                 (THIS IS CONVENTIONAL BASE
         USING IATUSMN,R10         FOR JES3 MODULES).
         USING IATUSMND,R13        REG.13 ==> CSECT 'ENTRY POINT'
         L     R15,FCTDSPDC        REG.15 ==> DSP DICTIONARY ENTRY
         USING DSPSTART,R15        FOR IATUSMN.
         OI    DSPFLAG1,DSPNUDRV+DSPNUDAT  SET FLAGS TO FORCE
         DROP  R15                 REFRESH NEXT TIME.
* ------ JESTAE & LOGIN
         LA    R0,USMNXIT1         POINT AT JESTAE RETRY POINT.
         ST    R0,USMNMRYA         SAVE FOR RETRY.
         LR    R1,R13
         JESTAE USMNTAE,PARAM=(R1)
         LOGIN ENTER=USMNME00      LOGIN TO CONSOLES.
* ------ ENQ IATUSMN ACTIVE - PREVENT ANOTHER MONITOR FROM STARTING
         LA    R0,USMNDEQ          POINT AT JESTAE RETRY POINT.
         ST    R0,USMNMRYA         SAVE FOR RETRY.
         LA    R15,USMNMJ10        POINT AT SERVICE ROUTINE
         BALR  R14,R15             ENTER ENQ ROUTINE
         LTR   R15,R15             GOOD RC?
         BZ    USMNSMP             YES - CONTINUE
* ------ ISSUE ALREADY ACTIVE MESSAGE AND EXIT
* ====== If you are on JES3 521, delete the @511 lines          @521
         MESSAGE MF=(E,USMNMS00),CNDB=UCON   CONSOLE MESSAGE     @521
* ====== If you are on JES3 511, delete the @521 lines          @511
         MESSAGE MF=(E,USMNMS00)   CONSOLE MESSAGE              @511
         B     USMNXIT1
* ------ FIND ALL LOCAL MAIN PROCESSOR NAMES
USMNSMP  DS    0H
         L     R9,TVTSSVT
         USING SSVT,R9
         L     R9,SVTMPCDA         POINT AT MAIN PROCESSOR TABLES
         DROP  R9
         USING MPCSTART,R9
         LA    R6,USMNMMPC         POINT AT MP NAME SLOTS
USMNLMP  DS    0H
         LTR   R9,R9               ANY MPC'S LEFT?
         BZ    USMNJAT             NO - DONE HERE
         TM    MPSYSTEM,MPGLBL     THIS ONE FOR GLOBAL?
         BO    USMNNMP             YES - TRY NEXT
         MVC   0(L'USMNMMPC,R6),MPNAME SAVE LOCAL'S NAME
         LA    R6,L'USMNMMPC(,R6)  POINT AT NEXT MP NAME SLOT
USMNNMP  DS    0H
         L     R9,MPNEXT           NEXT MP
         B     USMNLMP             LOOP...
```

```
         DROP  R9
* ------ CREATE INSTALLATION DEFINED ATTACHMENT TO JESXCF
USMNJAT  DS    0H
         LA    R15,USMNMJ00        POINT AT SERVICE ROUTINE
         BALR  R14,R15             ENTER IXZXIXAT ROUTINE
         LTR   R15,R15             GOOD RC?
         BZ    USMNJMB             YES - CONTINUE
* ------ ISSUE IATU200 ERROR MESSAGE AND EXIT
         MVC   USMNMS23,=C'AT'
         BAL   R6,USMNMSG2
         B     USMNDEQ
* ------ ATTACH TO JESXCF DEFAULT MAILBOX
USMNJMB  DS    0H
         LA    R0,USMNJDT          POINT AT JESTAE RETRY POINT.
         ST    R0,USMNMRYA         SAVE FOR RETRY.
         LA    R15,USMNMJ20        POINT AT SERVICE ROUTINE
         BALR  R14,R15             ENTER IXZXIXMB ROUTINE
         LTR   R15,R15             GOOD RC?
         BZ    USMNJMC             YES - CONTINUE
* ------ ISSUE IATU200 ERROR MESSAGE AND EXIT
         MVC   USMNMS23,=C'MB'
         BAL   R6,USMNMSG2
         B     USMNJDT
* ------ CLEAR JESXCF DEFAULT MAILBOX
USMNJMC  DS    0H
         LA    R0,USMNJMD          POINT AT JESTAE RETRY POINT.
         ST    R0,USMNMRYA         SAVE FOR RETRY.
         LA    R15,USMNMJ30        POINT AT SERVICE ROUTINE
         BALR  R14,R15             ENTER IXZXIXMC ROUTINE
         LTR   R15,R15             GOOD RC?
         BZ    USMNSIM             YES - CONTINUE
* ------ ISSUE IATU200 ERROR MESSAGE AND EXIT
         MVC   USMNMS23,=C'MC'
         BAL   R6,USMNMSG2
         B     USMNJMD
* ------ IATUSMN ACTIVE MESSAGE
USMNSIM  DS    0H
* ====== If you are on JES3 521, delete the @511 lines           @521
         MESSAGE MF=(E,USMNMS10),CNDB=UCON   CONSOLE MESSAGE      @521
* ====== If you are on JES3 511, delete the @521 lines           @511
         MESSAGE MF=(E,USMNMS10)   CONSOLE MESSAGE               @511
* ------ AWAIT FOR OPERATOR MESSAGE OR JESXCF MESSAGE
USMNAWT  DS    0H
         LA    R0,USMNSECF
         AWAIT TYPE=ON,ECFMASK=USMNMSG+USMNJSX,ECFADD=(R0)
         XR    R0,R0               SET ALL X'FF'S..
         BCTR  R0,0                .INTO R0
         ICM   R0,B'1000',=AL1(X'FF'-USMNJSX) COMPLEMENT OF JESXCF ECF
         L     R14,USMNSECF        PICK UP ECF
USMNDPT  DS    0H
         LR    R15,R14             DROP..
         NR    R15,R0              .JESXCF ECF..
         CS    R14,R15,USMNSECF    ..POSTING
         BNE   USMNDPT             TRY AGAIN...
         N     R14,USMNMED2        WAS THE POST FOR JESXCF MESSAGE?
         BC    NZERO,USMNJRM       YES - PROCESS JESXCF DATA.
         B     USMNJMD             MUST BE *C - EXIT
* ------ RECEIVE MAILBOX MESSAGE
USMNJRM  DS    0H
         LA    R15,USMNMJ40        POINT AT SERVICE ROUTINE
         BALR  R14,R15             ENTER IXZXIXRM ROUTINE
         LTR   R15,R15             GOOD RC?
         BZ    USMNMRS             YES - CONTINUE
         CH    R15,=H'4'           RC = 4?
         BC    NE,USMNMRE          NO - MUST BE AN ERROR
         CH    R0,=H'20'           RC = X'14' - NOTHING TO RECEIVE?
         BC    EQ,USMNAWT          CORRECT  - WAIT
* ------ ISSUE IATU200 ERROR MESSAGE AND KEEP GOING
USMNMRE  DS    0H
         MVC   USMNMS23,=C'RM'
         BAL   R6,USMNMSG2
         B     USMNJMD
* ------ PROCESS MAILBOX MESSAGE
USMNMRS  DS    0H
         L     R6,USMNMJDA         GET MESSAGE ADDRESS
         USING IXZYIXEN,R6
         CLC   YIXENEYE,=CL6'YIXEN ' IS THIS AN ENVELOPE?
         BC    NE,USMNJAC          NO - ACK IT ANYWAY
         TM    MESSAGE_CONTENT,SYSTEM_EVENT ?
         BZ    USMNJAC             NO - ACK IT
         LR    R7,R6
         AH    R7,MESSAGE_OFFSET   POINT AT SYSEVENT SECTION
```

```
         USING IXZYIXSE,R7
         CLC   YIXSEEYE,=CL6'YIXSE ' IS THIS AN SYSEVENT?
         BC    NE,USMNJAC          NO - ACK IT
         TM    YIXSE_TYPE,YIXSE_SYSEVENT  XCF SYSTEM EVENT?
         BZ    USMNJAC             NO - ACK IT
         LR    R8,R7
         AH    R8,YIXSE_OFFSET     POINT AT XCF DATA
         USING GEPL,R8
         MVC   USMNMS34,=C'GESYSACT' SYSTEM REPORTED ACTIVE
         CLI   GEPLTYPE,GESYSACT   SYSTEM REPORTED ACTIVE?
         BC    EQ,USMNJCL          YES - CHECK FOR LOCAL
         MVC   USMNMS34,=C'GESYSGON' SYSTEM REPORTED GONE
         CLI   GEPLTYPE,GESYSGON   SYSTEM REPORTED GONE?
         BC    EQ,USMNJCL          YES - CHECK FOR LOCAL
         MVC   USMNMS34,=C'GESYSDG ' SYSTEM DETECTED GONE
         CLI   GEPLTYPE,GESYSDG    SYSTEM DETECTED GONE?
         BC    NE,USMNJAC          NO - ACK IT
USMNJCL  DS    0H
         MVC   USMNMS33,GEPLSYS    SAVE SYSTEM NAME
         LA    R5,USMNMMPC         POINT AT MP NAME SLOTS
USMNLNL  DS    0H
         CLI   0(R5),X'FF'         END OF LOCAL NAMES?
         BE    USMNLGM             YES - ISSUE SYSTEM GONE MESSAGE
         CLC   0(L'USMNMMPC,R5),USMNMS33 JES3 LOCAL'S NAME
         BE    USMNLFL             YES - *S LOCAL FLUSH / *V LOCA ON
         LA    R5,L'USMNMMPC(,R5)  POINT AT NEXT MP NAME SLOT
         B     USMNLNL             LOOP...
* ------ DETERMINE ACTION FOR SYSTEM (DETECTED) GONE & SYSTEM ACTIVE
USMNLFL  DS    0H
         CLI   GEPLTYPE,GESYSACT   SYSTEM REPORTED ACTIVE?
         BC    EQ,USMNLVO          YES - VARY IT ON
         CLI   GEPLTYPE,GESYSGON   SYSTEM REPORTED GONE?
         BC    NE,USMNLGM          NO - JUST ISSUE INFO MESSAGE
         DROP  R6,R7,R8
* ------ INTERCOM *S LOCAL FLUSH (FLUSH NEEDS TO BE ISSUED TWICE)
*        MVC   USMNMCM2,USMNMS33   SYSTEM NAME
         MVI   USMNMCM2,C' '              TEMPORARY..
         MVC   USMNMCM2+1(L'USMNMCM2-1),USMNMCM2 .FIX..
         LA    R2,L'USMNMCM2              ..TO..
         LA    R14,USMNMS33+L'USMNMS33-1  ...*S MAIN FLUSH..
         LA    R15,USMNMCM2+L'USMNMCM2-1  ....WHICH..
USMNLF1  DS    0H                         .....CURRENTLY..
         CLI   0(R14),C' '                ......CANNOT..
         BE    USMNLSB                    .......STEP..
         MVC   0(1,R15),0(R14)            ........OVER..
         BCTR  R15,0                      .........BLANKS..
USMNLSB  DS    0H                         ..........AFTER..
         BCTR  R14,0                      ...........MAIN..
         BCT   R2,USMNLF1                 ............NAME
         LA    R2,2                LOOP COUNT
USMNLF2  DS    0H
* ====== If you are on JES3 521, delete the @511 lines           @521
         INTERCOM TEXT=USMNMCM0,MSG=NO (USE CONS ID 0 - INTERNAL) @521
* ====== If you are on JES3 511, delete the @521 lines           @511
         INTERCOM CONS=DUMMY,                                    @511 *
               TEXT=USMNMCM0,MSG=NO                              @511
         BCT   R2,USMNLF2          LOOP...
         B     USMNLGM
* ------ INTERCOM *V LOCAL ON    (AS FLUSH VARIED IT OFF)
USMNLVO  DS    0H
* ====== If you are on JES3 521, delete the @511 lines           @521
         MVC   USMNMCD2,USMNMS33   SYSTEM NAME
         INTERCOM TEXT=USMNMCM0,MSG=NO (USE CONS ID 0 - INTERNAL) @521
* ====== If you are on JES3 511, delete the @521 lines           @511
         INTERCOM CONS=DUMMY,                                    @511 *
               TEXT=USMNMCD0,MSG=NO                              @511
         B     USMNLGM
* ------ SYSTEM GONE MESSAGE
USMNLGM  DS    0H
* ====== If you are on JES3 521, delete the @511 lines           @521
         MESSAGE MF=(E,USMNMS30),CNDB=UCON   CONSOLE MESSAGE      @521
* ====== If you are on JES3 511, delete the @521 lines           @511
         MESSAGE MF=(E,USMNMS30)   CONSOLE MESSAGE               @511
* ------ ACKNOWLEDGE THE RECEIVED MESSAGE
USMNJAC  DS    0H
         LA    R15,USMNMJ50        POINT AT SERVICE ROUTINE
         BALR  R14,R15             ENTER IXZXIXAC ROUTINE
         LTR   R15,R15             GOOD RC?
         BZ    USMNJRM             YES - TRY TO RECEIVE MORE
* ------ ISSUE IATU200 ERROR MESSAGE AND KEEP GOING
         MVC   USMNMS23,=C'AC'
         BAL   R6,USMNMSG2
```

```
         B     USMNJRM
* ------ DETACH FROM JESXCF DEFAULT MAILBOX
USMNJMD  DS    0H
         LA    R0,USMNJDT          POINT AT JESTAE RETRY POINT.
         ST    R0,USMNMRYA         SAVE FOR RETRY.
         LA    R15,USMNMJ70        POINT AT SERVICE ROUTINE
         BALR  R14,R15             ENTER IXZXIXMD ROUTINE
         LTR   R15,R15             GOOD RC?
         BZ    USMNJDT             YES - CONTINUE
* ------ ISSUE IATU200 ERROR MESSAGE AND CONTINUE TERMINATION
         MVC   USMNMS23,=C'MD'
         BAL   R6,USMNMSG2
         B     USMNJDT
* ------ DELETE INSTALLATION DEFINED ATTACHMENT TO JESXCF
USMNJDT  DS    0H
         LA    R0,USMNDEQ          POINT AT JESTAE RETRY POINT.
         ST    R0,USMNMRYA         SAVE FOR RETRY.
         LA    R15,USMNMJ80        POINT AT SERVICE ROUTINE
         BALR  R14,R15             ENTER IXZXIXDT ROUTINE
         LTR   R15,R15             GOOD RC?
         BZ    USMNDEQ             YES - CONTINUE
* ------ ISSUE IATU200 ERROR MESSAGE AND CONTINUE TERMINATION
         MVC   USMNMS23,=C'DT'
         BAL   R6,USMNMSG2
* ------ DEQ IATUSMN ACTIVE
USMNDEQ  DS    0H
         LA    R0,USMNXIT1         POINT AT JESTAE RETRY POINT.
         ST    R0,USMNMRYA         SAVE FOR RETRY.
         LA    R15,USMNMJ90        POINT AT SERVICE ROUTINE
         BALR  R14,R15             ENTER DEQ ROUTINE
* ------ CLEANUP, LOGOUT & OTHER RETURN PROCESSING
USMNXIT1 DS    0H
         LA    R0,USMNXIT2         POINT AT JESTAE RETRY POINT.
         ST    R0,USMNMRYA         SAVE FOR RETRY.
* ====== If you are on JES3 521, delete the @511 lines           @521
         MESSAGE MF=(E,USMNMS80),CNDB=UCON   CONSOLE MESSAGE      @521
* ====== If you are on JES3 511, delete the @521 lines           @511
         MESSAGE MF=(E,USMNMS80)   CONSOLE MESSAGE                @511
USMNXIT2 DS    0H
         JESTAE 0
         LOGOUT
         SR    R15,R15             SET RETURN CODE TO JSS.
         L     R14,JSSRETRN        JSS RETURN POINT.
         BR    R14                 RETURN TO JSS.
* ------ SUBROUTINE: ISSUE IATU200 ERROR MESSAGE
USMNMSG2 DS    0H
         UNPK  USMNMS24(5),USMNMJRR+2(3)
         TR    USMNMS24(4),USMNMH2C-C'0'
         MVI   USMNMS25,C'/'
         UNPK  USMNMS26(5),USMNMJRR+6(3)
         TR    USMNMS26(4),USMNMH2C-C'0'
         MVI   USMNMS27,C' '
* ====== If you are on JES3 521, delete the @511 lines           @521
         MESSAGE MF=(E,USMNMS20),CNDB=UCON   CONSOLE MESSAGE      @521
* ====== If you are on JES3 511, delete the @521 lines           @511
         MESSAGE MF=(E,USMNMS20)   CONSOLE MESSAGE                @511
         BR    R6
* ------ JESTAE ROUTINE
USMNTAE  DS    0H
         USING FSWSTART,R1
         L      R10,USMNBASE-USMNTAE(,R15) PICK UP BASE ADDR.
         ST     R10,FSWSR10       SET THE BASE FOR RETRY
         L      R13,FSWPARAM      PT AT DATA CSECT
         ST     R13,FSWSR13       SET DATA CSECT ADDR FOR RETRY
         L      R0,USMNTRYA-USMNTAE(,R15) PICK UP RETRY ADDR.
         LA     R15,4             REQUEST RETRY.
         BR     R14               ATTEMPT RETRY
         DROP   R1
USMNBASE DC     A(IATUSMN)
USMNTRYA DC     A(USMNTRY)
* ------ JESTAE RETRY ROUTINE
USMNTRY  DS     0H
* ====== If you are on JES3 521, delete the @511 lines           @521
         MESSAGE MF=(E,USMNMS90),CNDB=UCON   CONSOLE MESSAGE      @521
* ====== If you are on JES3 511, delete the @521 lines           @511
         MESSAGE MF=(E,USMNMS90)   CONSOLE MESSAGE                @511
         L     R15,USMNMRYA        POINT AT RETRY ADDRESS
         BR    R15                 ENTER RETRY
*-------
         IATXPTCH LT
         DROP  R10
*-------
         TITLE 'IATUSMD - DATA CSECT MAPPING.'
IATUSMND DSECT
         IATYUSM
IATUSMN  CSECT
APARNUM  DC   CL8' '
PTFNUM   DC   CL8' '
         END  IATUSMN
```

E.2 JES3 Local Monitor Data Csect - IATUSMD

```
USMD     TITLE 'IATUSMD - JES3 LOCAL MONITOR DATA CSECT'
IATUSMD  AMODE 31
IATUSMD  RMODE ANY
         IATYASM
         PRINT NOGEN
**       TITLE 'IATYCNS - CONSOLE BUFFER (INPUT MESSAGE) DSECT.'
         IATYCNS TYPE=(FCTQ,INPUT),CODES=YES
**       TITLE 'IATYEQU - SYMBOLIC MASK DEFINITIONS.'
         IATYEQU
**       TITLE 'IATYREG - SYMBOLIC REGISTER DEFINITIONS.'
         IATYREG
**       TITLE 'IATYTVT - TRANSFER VECTOR TABLE DSECT.'
         IATYTVT
**       TITLE 'IATYSVT - SUBSYSTEM VECTOR TABLE FOR JES3.'
         IATYSVT
         PRINT GEN
IATUSMD  CSECT
         IATYUSM
IATUSMD  CSECT
APARNUM  DC    CL8' '
PTFNUM   DC    CL8' '
         END   IATUSMD
```

E.3 JES3 Local Monitor Macro - IATYUSM

```
         MACRO
         IATYUSM
         IATYMOD BR=NO,ID=IATUSMD,
               ENVIRON=J3MAINTASK
* ------ CONSOLE MESSAGE APPENDAGE
         USING TVTABLE,R12
         USING CONSMESS,R1
USMNME00 DS    0H
         LR    R10,R15              SET REG.10 AS BASE.
         USING USMNME00,R10
         LA    R15,8                RC=08 TO REJECT MESSAGE.
         CLI   CONACTN,CNCANCEL     *C REQUESTED?
         BC    NE,USMNME03          NO - IGNORE IT
USMNME01 DS    0H
         LA    R15,USMNSECF         POINT AT ECF.
         L     R2,0(0,R15)          ECF.
USMNME02 LR    R3,R2                POST ECF USING COMPARE
         O     R3,USMNMED1          AND SWAP.
         CS    R2,R3,0(R15)
         BC    NE,USMNME02
         LA    R15,4                RC=04 TO QUEUE MESSAGE.
         USING FCTSTART,R11
USMNME03 ARETURN RC=(R15)
         DROP  R1,R10,R11
         LTORG
* ------ ATTACH TO JESXCF GROUP
USMNMJ00 DS    0H
         LR    R6,R15               SET REG.6 AS BASE.
         USING USMNMJ00,R6
         LR    R8,R14               SAVE RETURN ADDRESS
         L     R9,TVTSSVT
         USING SSVT,R9
         LA    R7,SVTYMOD           POINT AT MODULE IDENTIFIER
         USING MODSTART,R7
         IXZXIXAT GROUP=USMNMJGR,  ATTACH JES3 TO JESXCF
               MEMBER=USMNMJMB,    MEMBER NAME IS MAIN NAME
               WHICHJES=JES3,
               RELEASE=MODREL,     CURRENT RELEASE FROM TVT
               GROUPTOKEN=SVTRSVU4 GROUP TOKEN IS TO GO HERE
         STM   R15,R0,USMNMJRR      SET RETURN/REASON CODES
         LR    R14,R8
         BR    R14
         DROP  R6,R7,R9
* ------ ENQ IATUSMN ACTIVE - ONLY ONE IATUSMN CAN BE ACTIVE
USMNMJ10 DS    0H
         LR    R6,R15               SET REG.6 AS BASE.
         USING USMNMJ10,R6
         LR    R8,R14               SAVE RETURN ADDRESS
         ENQ   (USMNMJGR,USMNMJMB,E,,STEP),RET=USE
         ST    R15,USMNMJRR         SET RETURN CODE
         LR    R14,R8
         BR    R14
         DROP  R6
* ------ ATTACH TO THE JESXCF DEFAULT MAILBOX.
USMNMJ20 DS    0H
         LR    R6,R15               SET REG.6 AS BASE.
         USING USMNMJ20,R6
         LR    R8,R14               SAVE RETURN ADDRESS
         L     R9,TVTSSVT
         USING SSVT,R9
         IXZXIXMB MBOXNAME=USMNMDMB, ATTACH TO DEFAULT MAILBOX         *
               POSTXIT=USMNMJXP,   EXIT ADDRESS                        *
               POSTDATA=TVTSSVT,   EXIT DATA IS DATA CSECT ADDRESS     *
               SYSEVENT=YES,       WANT TO SEE JESXCF SYSEVENTS        *
               GROUPTOKEN=SVTRSVU4  GROUP TOKEN IS HERE
         STM   R15,R0,USMNMJRR      SET RETURN/REASON CODES
         LR    R14,R8
         BR    R14
         DROP  R6,R9
* ------ CLEAR THE JESXCF DEFAULT MAILBOX.
USMNMJ30 DS    0H
         LR    R6,R15               SET REG.6 AS BASE.
         USING USMNMJ30,R6
         LR    R8,R14               SAVE RETURN ADDRESS
         L     R9,TVTSSVT
         USING SSVT,R9
         IXZXIXMC MBOXNAME=USMNMDMB, ATTACH TO DEFAULT MAILBOX         *
               GROUPTOKEN=SVTRSVU4  GROUP TOKEN IS HERE
         STM   R15,R0,USMNMJRR      SET RETURN/REASON CODES
         LR    R14,R8
         BR    R14
         DROP  R6,R9
**------ RECIVE DATA FROM JESXCF DEFAULT MAILBOX.
USMNMJ40 DS    0H
         LR    R6,R15               SET REG.6 AS BASE.
         USING USMNMJ40,R6
         LR    R8,R14               SAVE RETURN ADDRESS
         L     R9,TVTSSVT
         USING SSVT,R9
         IXZXIXRM MBOXNAME=USMNMDMB, DELETE THE DEFAULT MAILBOX        *
               MSGFETCH=ALL,        SELECT EVERYTHING                  *
               DATA=USMNMJDA,       DATA ADDRESS                       *
               DATALEN=USMNMJDL,    DATA LENGTH                        *
               MSGTOKEN=USMNMJMT,   MESSAGE TOKEN                      *
               GROUPTOKEN=SVTRSVU4  GROUP TOKEN IS HERE
         STM   R15,R0,USMNMJRR      SET RETURN/REASON CODES
         LR    R14,R8
         BR    R14
         DROP  R6,R9
* ------ ACKNOWLEDGE JESXCF MESSAGE.
USMNMJ50 DS    0H
         LR    R6,R15               SET REG.6 AS BASE.
         USING USMNMJ50,R6
         LR    R8,R14               SAVE RETURN ADDRESS
         L     R9,TVTSSVT
         USING SSVT,R9
         IXZXIXAC MSGTOKEN=USMNMJMT,   MESSAGE TOKEN                   *
               GROUPTOKEN=SVTRSVU4  GROUP TOKEN IS HERE
         STM   R15,R0,USMNMJRR      SET RETURN/REASON CODES
         LR    R14,R8
         BR    R14
         DROP  R6,R9
* ------ DETACH FROM THE JESXCF DEFAULT MAILBOX.
USMNMJ70 DS    0H
 *       LR    R6,R15               SET REG.6 AS BASE.
 *       USING USMNMJ70,R6
 *       LR    R8,R14               SAVE RETURN ADDRESS
 *       L     R9,TVTSSVT
         USING SSVT,R9
         IXZXIXMD MBOXNAME=USMNMDMB, DELETE THE DEFAULT MAILBOX        *
               GROUPTOKEN=SVTRSVU4  GROUP TOKEN IS HERE
         STM   R15,R0,USMNMJRR      SET RETURN/REASON CODES
         LR    R14,R8
         BR    R14
         DROP  R6,R9
* ------ DETACH FROM JESXCF GROUP.
USMNMJ80 DS    0H
         LR    R6,R15               SET REG.6 AS BASE.
         USING USMNMJ80,R6
         LR    R8,R14               SAVE RETURN ADDRESS
         L     R9,TVTSSVT
         USING SSVT,R9
         IXZXIXDT GROUPTOKEN=SVTRSVU4 GROUP TOKEN IS HERE
         STM   R15,R0,USMNMJRR      SET RETURN/REASON CODES
         LR    R14,R8
         BR    R14
         DROP  R6,R9
* ------ DEQ IATUSMN ACTIVE.
USMNMJ90 DS    0H
         LR    R6,R15               SET REG.6 AS BASE.
 *       USING USMNMJ90,R6
 *       LR    R8,R14               SAVE RETURN ADDRESS
 *       DEQ   (USMNMJGR,USMNMJMB,,STEP)
 *       ST    R15,USMNMJRR         SET RETURN CODE
         LR    R14,R8
         BR    R14
         DROP  R6
* ------ JESXCF POST EXIT SUBROUTINE.
*        POST THE IATUSMN ECF ASSOCIATED WITH THE MAILBOX.
*        THIS ROUTINE DOES NOT RUN UNDER THE JES3 MAIN TASK,
*        BUT DOES RUN IN THE JES3 ADDRESS SPACE.
USMNMJXR DS    0H
         BAKR  R14,0                SAVE CALLER'S REGISTERS
         LR    R10,R15              SETUP SUBROUTINE'S
         USING USMNMJXR,R10            ADDRESSABILITY
         L     R5,0(,R1)            GET ADDRESSABILITY TO THE
 *       USING IXZYIXPE,R5            POST EXIT PARAMETER LIST
*------- VALIDATE THE INPUT PARAMETERS                                 *
         CLC   YIXPEEYE,=CL6'YIXPE'  EXIT P_LIST VALID?
         BC    NE,USMNMJX4          NO, INDICATE ERROR AND RETURN
```

```
         L     R9,YIXPE_POSTDATA   POINT AT POST_DATA - JES3 SSVT
         DROP  R5
*------- POST IATUSMN                                          *
         LA    R15,USMNSECF        POINT AT ECF.
         L     R2,0(0,R15)         ECF.
USMNMJXL LR    R3,R2               POST ECF USING COMPARE
         O     R3,USMNMED2         AND SWAP.
         CS    R2,R3,0(R15)
         BC    NE,USMNMJXL
*------- POST JES3                                             *
         USING SSVT,R9
         MODESET EXTKEY=ZERO,SAVEKEY=(2) MUST BE IN KEY 0
         IATXPJ3 ADDR=SSVT,TYPE=JPOST POST JES3
         MODESET KEYADDR=(2)       RESTORE ORIGINAL KEY
         XR    R15,R15             SET NORMAL RETURN CODE
         XR    R0,R0               SET NORMAL REASON CODE
         B     USMNMJXX            GO RETURN TO CALLER
*------- ERROR RETURN
USMNMJX4 DS    0H
         LA    R15,4               INDICATE INPUT ERROR
         LA    R0,4                INDICATE BAD PARAMETER LIST
*------- O.K. RETURN
USMNMJXX DS    0H
         PR    ,                   RETURN
         DROP  R10,R9
*-------
         LTORG
*-------
USMNMJDA DC    A(0)                MESSAGE DATA ADDRESS
USMNMJDL DC    A(0)                MESSAGE DATA LENGTH
USMNMJMT DC    XL8'0'              MESSAGE TOKEN
*-------
USMNMDMB DC    CL16'SYSJES$DEFAULT' DEFAULT MAILBOX NAME
USMNMJXP DC    A(USMNMJXR)         POST EXIT ADDRESS
*-------
USMNMJGR DC    CL8'J3LOCAL'        XCF GROUP NAME
USMNMJMB DC    CL16'PRIVATEEYE'
USMNMJRR DC    2F'0'
*-------
USMNMSG  EQU   X'80'               ECF - OPERATOR MESSAGE
USMNJSX  EQU   X'40'               ECF - JESXCF MESSAGE
USMNSECF DC    A(0)                ECF
USMNMED1 DC    AL1(USMNMSG),XL3'00' ECF MASK FOR OPERATOR MESSAGE
USMNMED2 DC    AL1(USMNJSX),XL3'00' ECF MASK FOR JESXCF MESSAGE
*-------
USMNMRYA DS    F                   RETRY ADDRESS
*-------
USMNMS00 MESSAGE MF=L,CLASS=JES,TEXT=USMNMS01
USMNMS01 DC    AL1(USMNMS09-USMNMS02)     ALREADY ACTIVE MESSAGE.
USMNMS02 DC    C'IATU110 - LOCAL MONITOR ALREADY ACTIVE'
USMNMS09 DS    0C
*-------
USMNMS10 MESSAGE MF=L,CLASS=JES,TEXT=USMNMS11
USMNMS11 DC    AL1(USMNMS19-USMNMS12)     LOGON MESSAGE.
USMNMS12 DC    C'IATU100 - LOCAL MONITOR ACTIVE'
USMNMS19 DS    0C
*-------
USMNMS20 MESSAGE MF=L,CLASS=JES,TEXT=USMNMS21
USMNMS21 DC    AL1(USMNMS29-USMNMS22)     IXZXIXNN ERROR MESSAGE.
USMNMS22 DC    C'IATU200 - IXZXIX'
USMNMS23 DC    C'XX'
         DC    C' RETURN/REASON CODE = '
USMNMS24 DC    C'XXXX'
```

```
USMNMS25 DC    C'/'
USMNMS26 DC    C'XXXX'
USMNMS27 DC    C' '
USMNMS29 DS    0C
*-------
USMNMS30 MESSAGE MF=L,CLASS=JES,TEXT=USMNMS31
USMNMS31 DC    AL1(USMNMS39-USMNMS32)     XCF SYSTEM GONE MESSAGE.
USMNMS32 DC    C'IATU300 - SYSTEM '
USMNMS33 DC    CL8' '
         DC    C'-'
USMNMS34 DC    CL8' '
USMNMS36 DC    C' RECEIVED FROM XCF'
USMNMS39 DS    0C
*-------
USMNMS80 MESSAGE MF=L,CLASS=JES,TEXT=USMNMS81
USMNMS81 DC    AL1(USMNMS89-USMNMS82)     LOGOUT MESSAGE.
USMNMS82 DC    C'IATU999 - LOCAL MONITOR TERMINATED'
USMNMS89 DS    0C
*-------
USMNMS90 MESSAGE MF=L,CLASS=JES,TEXT=USMNMS91
USMNMS91 DC    AL1(USMNMS99-USMNMS92)     ABEND MESSAGE.
USMNMS92 DC    C'IATU997 - ABEND IN LOCAL MONITOR'
USMNMS99 DS    0C
*-------
USMNMCM0 DC    AL1(USMNMCM3-USMNMCM1)
USMNMCM1 DC    C'*S '
USMNMCM2 DC    CL8' '
         DC    C' FLUSH '
USMNMCM3 DS    0C
*-------
USMNMCD0 DC    AL1(USMNMCD3-USMNMCD1)
USMNMCD1 DC    C'*V '
USMNMCD2 DC    CL8' '
         DC    C' ON '
USMNMCD3 DS    0C
*-------
USMNMH2C DC    C'0123456789ABCDEF'
USMNMMPC DC    32XL8'FFFFFFFFFFFFFFFF'
* ====== If you are on JES3 511, delete the @521 lines           @521
*-------                                                          @521
UCON     IATYCNDB DSECT=NO                                        @521
UCONRTCD DC    XL17'4000000000000000000000000000000000'  DEFINE   @521
*                                ROUTE CODE 2                     @521
         DS    0D
         PUSH  PRINT
         PRINT NOGEN
*        TITLE 'IATYFCT - FUNCTION CONTROL TABLE DSECT.'
         IATYFCT
*        MODULE ENTRY POINT IDENTIFIER
         IATYMOD DSECT=YES,BR=NO
*        JESXCF MESSAGE ENVELOPE
         IXZYIXEN
*        JESXCF POST EXIT PARAMETER LIST
         IXZYIXPE
*        JESXCF SYSTEM EVENT MESSAGE
         IXZYIXSE
*        XCF GROUP EXIT PARAMETER LIST
         IXCYGEPL
         PRINT GEN
         POP   PRINT
         MEND
```

Appendix F. MCS Sysplex Operations

This appendix discusses the many changes to MCS in support of the sysplex environment.

F.1 Consoles in a Sysplex

MCS consoles may be attached to any system in a sysplex. The physical attachment does not limit their span of control. A single MCS console can operate any system in the sysplex. An operator can control activity in the sysplex from any console regardless of the type of the console (DIDOCS, extended MCS or system console) or where it is attached. A single master console can control all systems in a sysplex. Multiple consoles may have master authority.

F.1.1 System Console

The system console is automatically defined during MVS initialization. How the system console receives messages after initialization can be controlled by defining values in the CONSOLxx parmlib member. During normal operations, when the system console is not in problem determination mode, it receives a minimal set of messages. When the system console is in problem determination mode, after a VARY CN,ACTIVATE on the system console, an operator can:

- Enter commands and receive messages to help debug the system problem
- Control console attribute values for the system console

The system console should be used only for initialization of MVS and for backup and recovery purposes. The system console should not be used alone to operate the system because system performance can be seriously affected. For normal operation of the system, you should use MCS consoles or extended MCS consoles, or subsystem consoles (like NetView consoles).

F.1.2 MCS Consoles

The CONSOLxx parmlib member defines an MCS console configuration (up to 99 consoles in a MCS sysplex) and MCS processing options. Some of the CONSOLxx definitions have sysplex scope and are only processed by the first system entering the sysplex:

- RLIM (on the INIT statement)
- AMRF (on the INIT statement)
- CNGRP (on the INIT statement)
- ROUTTIME (on the INIT statement)
- RMAX (on the DEFAULT statement)

MCS consoles are assigned both 4-byte and 1-byte console identifiers. The console IDs are used by command processors to route command responses back to the issuing console. EMCS consoles normally get only the 4-byte console ID unless at activation time a 1-byte migration ID is also requested. “Old” command processors, for example JES3 releases prior to JES3 5.2.1, were not programmed to use the 4-byte console ID and return command responses to the originating console when only a 4-byte console ID is available.

F.1.3 Extended MCS Consoles

An extended MCS console (EMCS) is defined through MCSOPER service by the program that uses it. TSO/E, NetView, and SDSF are examples of programs that use EMCS consoles.

Programs can issue commands from an extended MCS console through the MGCRE macro service and include a command and response token (CART), which associates a response with a command. Messages that are queued to EMCS consoles may be selectively retrieved through MCSOPMSG macro service by specifying a CART value or CART mask. Once again, "old" command processors, for example JES3 releases prior to JES3 5.2.1, were not programmed to preserve the CART from the command and return the response through WTO macro service with the CART.

Extended MCS consoles provide flexibility in the number of consoles that can be used in an MVS system. Defining such consoles allows you to increase the number of consoles beyond the MCS limit of 99.

An installation can assign an authorized user of an extended MCS console many of the attributes of MCS consoles (for example, the commands that can be issued and the messages that will be received). The attributes for extended consoles are controlled through the OPERPARM segment in the user′s RACF profile.

Currently, extended MCS consoles can be used by:

TSO/E TSO/E users with the appropriate authority can use the CONSOLE command to establish an extended MCS console session. TSO/E extended MCS consoles can be authorized to issue commands requiring master authority.

The GETMSG function allows TSO/E users to write REXX programs to handle console output.

NetView Beginning with NetView Version 2 Release 3, the option of communicating with MVS through extended MCS consoles was introduced.

Specifying the appropriate parameters ensures that NetView uses extended MCS consoles on MVS/ESA systems. In addition, individual NetView operators and autotasks can allocate themselves an extended MCS console by issuing the GETCONID command.

Applications Application programs can be authorized to establish extended MCS console sessions, during which commands can be issued and messages (both command responses and unsolicited messages) can be received.

Three assembler macros are needed to use extended consoles:

MCSOPER The MCSOPER macro activates and deactivates extended console sessions. The console authorization is defined either using OPERPARM data from RACF or the OPERPARM parameter list on the MCSOPER macro.

MCSOPER can specify a parameter called MSGDLVRY. This determines whether messages will be delivered to this extended

MCS console. If MSGDLVRY=NONE is specified, no messages will be delivered to the console.

MGCRE The MGCRE macro enables the user to issue operator commands.

MCSOPMSG Messages are returned to extended MCS consoles in a Message Data Block (MDB). The program that issues the MCSOPMSG macro must include the MDB mapping macro (IEAVM105) in order to gain access to the MDB fields.

F.2 MCS Command and Message Flow

Commands and messages flow through a system in a similar way as shown in Figure 52.

F.2.1 Command Flow

When an operator command is entered through the MGCRE macro service (see Figure 52), it is first processed by the installation command exit routines.

The exits are specified using the .CMD statement in the MPFLSTxx parmlib member. These exits can perform authority checking, modify the command text, or the command processing. A return code set by the exits indicates whether the command was processed by the exit, or MCS should process the command, or the user is not authorized to issue the command.

The command is then broadcast on the Subsystem Interface (SSI) to all active subsystems. Each subsystem, beginning with the primary job entry subsystem, may inspect the command and decide whether to process it. The subsystems make this decision based on the command prefix characters of the command string. For example, by default, NetView looks for a percent sign (%) as the first character. If a subsystem decides that it is to process the command, the command is passed to subsystem processing, and a return code is set to indicate that the command was processed by a subsystem.

Once the command has been examined by all active subsystems, it is logged to the hardcopy log (SYSLOG and/or OPERLOG).

If none of the subsystems has marked the command as having been processed, it is assumed to be an MVS command and is passed to the appropriate MVS command processor. If one does not exist, an error message is issued.

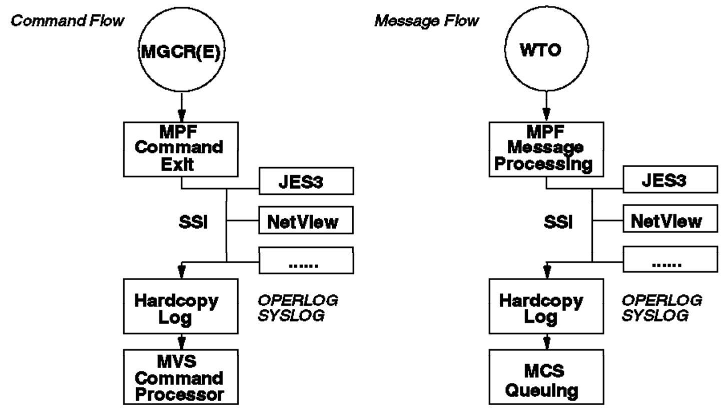


Figure 52. MCS Command and Message Flow

F.2.2 Message Flow

A message is entered to the system either through a WTO (Write to Operator) or a WTOR (Write to Operator with Reply) macro service. The message can be:

Unsolicited An unsolicited message is routed by a routing code; that is, the message is issued by the system and it is not a response to a command.

Solicited Solicited messages are responses to commands issued by an operator.

The essential difference between solicited and unsolicited messages is that solicited messages are (normally) routed to the console that issued the command, while unsolicited messages go to consoles that are receiving the routing codes used for the message.

Figure 52 shows a typical message flow within a single system. First, the message is processed by MPF (Message Processing Facility). This processing is based on entries in the MPFLSTxx parmlib member.

MPF allows an installation to influence how WTO and WTOR messages are to be processed. Through the MPFLSTxx member, you can specify four processing options for a message:

- Suppress message display

  If a message is suppressed, it is logged to the hardcopy log, and does not appear on any console. WTOR messages and command responses cannot be suppressed.

- Retain action message

  Retention means that action messages are saved by the Action Message Retention Facility (ARMF) so that operators can view them later.

- Automation eligibility specifies that the message is eligible for automation. An automation subsystem, specifically NetView, can look at the message and can perform predefined operator actions. (Note that TSO/E and other EMCS consoles may be activated to receive automation messages.)

- Invoke an installation-written exit

You can write exit routines to process WTO and WTOR messages. MPFLSTxx can specify which messages are to be processed by which exit routine. The exit can alter the message text and the way in which the message is to be processed.

For full details of MPFLSTxx options and parameters, see *MVS/ESA SP V5 Initialization and Tuning Reference*.

Following MPF processing, the message is broadcast to all active subsystems. The message is presented to each subsystem in turn. Each subsystem may inspect the message and process it as appropriate. A subsystem can alter WQE fields, in which case later subsystems on the SSI will see the changed WQE. A WQE (Write-to-Operator Queue Element) is an internal control block that contains the message text and all related information for that message.

After the message has been inspected by all active subsystems, it is written to the hardcopy log (SYSLOG and/or OPERLOG) unless hardcopy logging is suppressed by an exit.

Finally the message is routed for display on the appropriate MCS and extended MCS consoles. The routing may require message transportation (using XCF services) to other systems in the sysplex because the receiving consoles may not be physically attached to the system where the message was issued.

F.3 Command Routing in a Sysplex

Several console definitions affect command routing in a sysplex. Command routing with JES3 5.2.1 is an important consideration due to the fact that the **T* command no longer can be used to send commands to a main processor. There are four options that can be used to send commands to another processor; they are:

- CMDSYS parameter on the CONSOLE statement in the CONSOLxx member of parmlib or the K V operator command
- Using the command prefix facility (CPF)
- The IEECMDPF program in SYS1.SAMPLIB
- The ROUTE command

Using XCF services, MCS transports the command directly to the command processing system, where the command is passed through MPF command user exits, the SSI loop, hardcopy logging, and finally through MVS command processing. This is the same process as would be taken for commands issued locally on that system.

Figure 53 shows command routing in a sysplex.

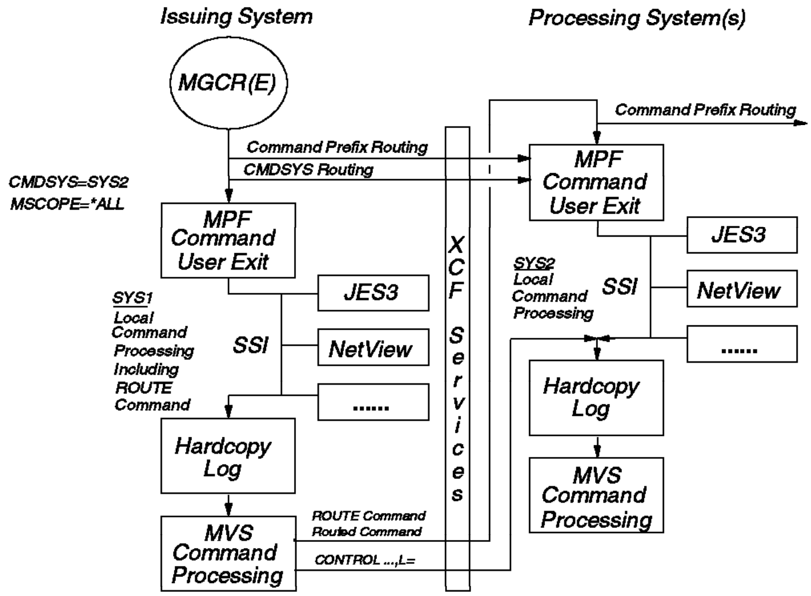


Figure 53. Command Routing in a Sysplex

F.3.1 CMDSYS Parameter and Routing

Consoles, both MCS and extended MCS, can have an associated CMDSYS value, which specifies the system to which commands issued on the console are to be sent. This provides an implicit method of allowing a console that is physically attached to one system to be logically associated with another system.

For MCS consoles, CMDSYS is specified in CONSOLxx. For EMCS consoles, it is part of the RACF OPERPARM segment.

Implicit command routing is done when:

- A console physically attached to one system, say SYS1, as shown in Figure 53, is logically associated with another system, say SYS2. That is, CMDSYS(SYS2) is in effect. This means that commands issued from this console are to be executed on another system.

Note that CPF processing is independent of CMDSYS processing and is taken before the CMDSYS routing.

Note also that there are some commands (LOGON/LOGOFF, TRACK/STOPTR, ROUTE and some variations of CONTROL) that are not affected by the CMDSYS values. These commands are always processed on the issuing system.

F.3.2 Command Prefix Routing

The command prefix facility (CPF) allows subsystems, such as JES3, to register command prefixes with MCS. These prefixes are defined through the CPF macro service and are held in the CPF table, which is propagated to all systems in a sysplex.

CPF allows any operator or any authorized application to enter a command from any system in a sysplex and route that command to the appropriate system for execution. The command responses will come back to the originating console. The application can be an installation exit, a subsystem, or an installation-written program.

The scope of a command prefix can be either SYSPLEX or SYSTEM:

SYSPLEX The command issued is routed to the system for which the prefix is defined.

SYSTEM The command issued is executed on the system on which the command is entered.

Note: To avoid conflicts with WTOR replies, the JES3 default SYSTEM scope prefix (8) is not registered with CPF and therefore is not displayed by the operator command *D OPDATA*. This is shown in Figure 54.

A console physically attached to one system, for example SYS1 as shown in Figure 52 on page 140, issues a command using a designated command prefix, indicating that the command is to be executed on another system, for example SYS2.

F.3.3 IEECMDPF Program

Another use for the CPF could be through the IEECMDPF program provided in SYS1.SAMPLIB. This program is intended to be executed on each system in a sysplex (for example, through a START command in a common COMMNDxx PARMLIB member) to create a command prefix for each system equal to its system name. This allows an installation to direct a single-system command to a given system by simply preceding the command with the system name.

The command prefixes in use can be displayed by issuing the MVS operator command DISPLAY OPDATA,PREFIX.

```
D OPDATA,PREFIX
 IEE603I 11.41.01 OPDATA DISPLAY 288
  PREFIX     OWNER      SYSTEM     SCOPE     REMOVE    FAILDSP
  *          JES3       SC50       SYSPLEX   NO        PURGE
  SC47       IEECMDPF   SC47       SYSPLEX   YES       SYSPURGE
  SC49       IEECMDPF   SC49       SYSPLEX   YES       SYSPURGE
  SC50       IEECMDPF   SC50       SYSPLEX   YES       SYSPURGE
```

Figure 54. D OPDATA Operator Command

F.3.4 ROUTE Command

The MVS ROUTE operator command explicitly routes another operator command for execution on another system in a sysplex. It can be issued from both MCS and EMCS consoles. The response to the command is returned to the issuing console (unless redirected by an L= parameter).

The format of the ROUTE command is:

```
ROUTE sysname,text
```

Where:

sysname The name of the system that is to process the command

text The command itself

Regardless of the CMDSYS value in effect for a console that issues the ROUTE command, the ROUTE command itself is processed on the system on which it was issued.

F.3.4.1 L= Command Operand

Using the L= operand on certain MVS commands, like CONTROL or DISPLAY, allows you to specify a target console name on any system in the sysplex. For example, an operator can enter the CONTROL command with L= on one console to change the console characteristics of another console on a different system.

Such commands go through all the normal processing stages on the issuing system: installation command exit processing, SSI broadcasting, logging and execution by an MVS command processor before being transported to the receiving system.

A flag is set to indicate that installation command exit processing and broadcasting on the SSI are to be bypassed on the receiving system. The command is, however, logged on the receiving system and is then processed by the appropriate MVS command processor.

Note that in this case the command is logged on both the issuing and the receiving system.

If the console identified by the L= operand is on a different system than the one that issued the command, the command has to be transported to that system for processing.

The ROUTE command, as shown in Figure 53 on page 142, is processed in two stages; the ROUTE command is processed on the issuing system, and the routed command, which is transported to the receiving system, is processed on the receiving system as a locally issued command.

The installation command exit and the active subsystems on the issuing system see the ROUTE command and its operands. The ROUTE command is logged in the hardcopy log on the issuing system. The ROUTE command processing on the issuing system then causes the routed command to be transported to the receiving system.

The routed command is then treated as though it originated on the receiving system for execution there.

In summary, the full ROUTE command is processed on the issuing system, and the routed command is processed on the receiving system.

F.4 Message Routing in a Sysplex

Figure 55 shows the message flow across systems in a sysplex. The message goes through all the steps of message processing on the issuing system. These steps were shown in Figure 52 on page 140 and F.2.2, “Message Flow” on page 140.

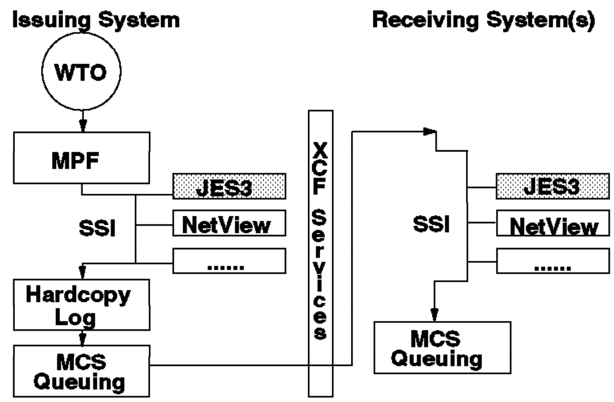


Figure 55. Message Flow in a Sysplex

If there are active consoles receiving this message or active subsystem allocatable consoles on other systems in the sysplex, the message is transported to these systems. On the receiving systems the message goes through the SSI loop, but it is not logged, and finally the message is processed by the message queuing tasks.

If a message is destined for a specific console that is not active in the sysplex, it is logged and discarded unless it is an action message or WTOR message, in which case it is displayed on consoles with the UD attribute (by default, the master console of the sysplex).

Messages already “delivered” (queued) to an extended MCS console operator but not yet displayed are purged from MCS queues when the console is deactivated; that is, unprocessed queued messages are not rerouted.

Appendix G. Special Notices

This publication is intended to help IBM support personnel and customers understand JES3 functions in a Parallel Sysplex. The information in this publication is not intended as the specification of any programming interfaces that are provided by MVS/ESA JES3 Version 5 Release 2.1, Program Number 5655-069. See the PUBLICATIONS section of the IBM Programming Announcement for MVS/ESA JES3 Version 5 Release 2.1, Program Number 5655-069 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.

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:

ACF/VTAM
CICS
DFSMS
DFSMS/MVS
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
ESCON
IBM
IMS
IMS/ESA
MVS
MVS/ESA
NetView
Parallel Sysplex
PR/SM
RACF
VTAM

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 H. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

H.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How To Get ITSO Redbooks" on page 153.

Short Title	Title	Order Number
JES3 Implementation Guide	*MVS/ESA SP-JES3 Implementation Guide*	SG24-4582
Version 5 Implementation Guide	*MVS/ESA Version 5 Implementation Guide*	SG24-4584
MVS 5.1 Presentation Guide	*MVS/ESA 5.1.0 Technical Presentation Guide*	GG24-4137
Parallel Sysplex Perf.	*S/390 Parallel Sysplex Performance*	SG24-4356
RACF V2.1 Inst. and Impl.	*RACF V2R1 Installation and Implementation Guide*	GG24-4405
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

A complete list of International Technical Support Organization publications, known as redbooks, with a brief description of each, may be found in:

International Technical Support Organization Bibliography of Redbooks, GG24-3070.

H.2 Other Publications

These publications are also relevant as further information sources.

- JES3 Publications for Version 5 Release 2.1.

Short Title	Title	Order Number
MVS/ESA SP V5 JES3 Commands	*MVS/ESA JES3 Commands*	GC28-1444
MVS/ESA SP V5 JES3 Conversion Notebook	*MVS/ESA JES3 Conversion Notebook*	GC28-1438
MVS/ESA SP V5 JES3 Initialization and Tuning Guide	*MVS/ESA JES3 Initialization and Tuning Guide*	SC28-1455
MVS/ESA SP V5 JES3 Initialization and Tuning Reference	*MVS/ESA JES3 Initialization and Tuning Reference*	SC28-1456
MVS/ESA SP V5 JES3 Messages	*MVS/ESA JES3 Messages*	GC28-1489
MVS/ESA SP V5 JES Common Coupling Services	*MVS/ESA Programming: JES Common Coupling Services*	GC28-1497

The following publications are available to IBM-licensed customers only.

Short Title	Title	Order Number
MVS/ESA SP V5 JES3 Customization	*MVS/ESA JES3 Customization*	LY28-1853
MVS/ESA SP V5 JES3 Diagnosis	*MVS/ESA JES3 Diagnosis*	LY28-1855

- Conversion

Short Title	Title	Order Number
MVS/ESA SP V5 Conversion Notebook	*MVS/ESA Conversion Notebook*	GC28-1436

- Operations

Short Title	Title	Order Number
MVS/ESA SP V5 Planning: Operations	*MVS/ESA Planning: Operations*	GC28-1441
MVS/ESA SP V5 System Commands	*MVS/ESA System Commands*	GC28-1442

- I/O Configuration Management

Short Title	Title	Order Number
MVS/ESA SP V5 HCD: Planning	*MVS/ESA Hardware Configuration Definition: Planning*	GC28-1445
HCD: User′s Guide	*MVS/ESA Hardware Configuration Definition: User′s Guide*	SC33-6468

- Multi-System Configuration Management

Short Title	Title	Order Number
Sysplex Overview	*System/390 MVS Sysplex Overview: An Introduction to Data Sharing and Parallelism*	GC28-1208
Sysplex Systems Management	*System/390 MVS Sysplex Systems Management*	GC28-1209
Sysplex Hardware and Software Migration	*System/390 MVS Sysplex Hardware and Software Migration*	GC28-1210
Sysplex Application Migration	*System/390 MVS Sysplex Application Migration*	GC28-1211
MVS/ESA SP V5 Setting Up a Sysplex	*MVS/ESA Setting Up a Sysplex*	GC28-1449
MVS/ESA SP V5 Sysplex Services Guide	*MVS/ESA Programming: Sysplex Services Guide*	GC28-1495
MVS/ESA SP V5 Sysplex Services Reference	*MVS/ESA Programming: Sysplex Services Reference*	GC28-1496
MVS/ESA SP V5 Planning: Global Resource Serialization	*MVS/ESA Planning: Global Resource Serialization*	GC28-1450
ESCON Manager Planning Guide	*Planning for the Enterprise Systems Connection Manager Version 1 Release 3*	GC23-0423

Short Title	Title	Order Number
ESCON Director Planning	*Planning for the 9032 Enterprise Systems Connection Director*	GA23-0364
ESCON Migration Planning	*ESCON: Planning for Migration*	GG66-3181
Sysplex Timer Planning	*Planning for the 9037 Sysplex Timer*	GA23-0365

- Initialization and Tuning

Short Title	Title	Order Number
MVS Initialization and Tuning Guide	*MVS/ESA Initialization and Tuning Guide*	SC28-1451
MVS Initialization and Tuning Reference	*MVS/ESA Initialization and Tuning Reference*	SC28-1452

- Problem Determination and Recovery

Short Title	Title	Order Number
Recovery and Reconfiguration Guide	*MVS/ESA Recovery and Reconfiguration Guide*	GC28-1458

- Application Development

Short Title	Title	Order Number
Authorized Assembler Services Guide	*MVS/ESA Programming: Authorized Assembler Services Guide*	GC28-1467
Authorized Assembler Services Library	*MVS/ESA SP V5 Authorized Assembler Services Reference, Volumes 1-4*	GC28-1475, GC28-1476, GC28-1477, GC28-1478

- Customization

Short Title	Title	Order Number
MVS/ESA SP V5 Installation Exits	*MVS/ESA Installation Exits*	SC28-1459
TSO Programming	*MVS TSO Programming*	GC28-1460

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. Ordering information 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/redbooks.

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:

- **IBMLINK**

  Registered customers have access to PUBORDER to order hardcopy, to REDPRINT to obtain BookManager BOOKs

- **IBM Bookshop** — send orders to:

  usib6fpl@ibmmail.com (United States)
  bookshop@dk.ibm.com (Outside United States)

- **Telephone orders**

1-800-879-2755	Toll free, United States only
(45) 4810-1500	Long-distance charge to Denmark, answered in English
(45) 4810-1200	Long-distance charge to Denmark, answered in French
(45) 4810-1000	Long-distance charge to Denmark, answered in German
(45) 4810-1600	Long-distance charge to Denmark, answered in Italian
(45) 4810-1100	Long-distance charge to Denmark, answered in Spanish

- **Mail Orders** — send orders to:

  IBM Publications
  P.O. Box 9046
  Boulder, CO 80301-9191
  USA

  IBM Direct Services
  Sortemosevej 21
  DK-3450 Allerød
  Denmark

- **Fax** — send orders to:

1-800-445-9269	Toll-free, United States only
45-4814-2207	Long distance to Denmark

- **1-800-IBM-4FAX (United States only)** — 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

- **Redbooks Home Page on the World Wide Web**

  http://www.redbooks.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  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). A category form and detailed instructions will be sent to you.

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 WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  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/redbooks.html
- **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.
- **ITSO4USA category on INEWS**
- **IBM Bookshop** — 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.

Index

Special Characters

A

B

C

D

E

F

G

I

J

Printed in U.S.A.

SG24-4776-00