

**Systems**

**OS/VS2 Planning Guide for  
Release 2**

Program Number 5742-020

VS2 Release 2

**IBM**

**Second Edition (November, 1973)**

This is a major revision of, and obsoletes, GC28-0667-0. See the Summary of Amendments following the Contents. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This edition applies to Release 2 of OS/VS2 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest **IBM System/360 and System/370 Bibliography**, GA22-6822, for editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Publications Development, Department D58, Building 706-2, PO Box 390, Poughkeepsie, N.Y. 12602. Comments become the property of IBM.

## Preface

This publication contains planning information for OS/VS2 Release 2. It is intended for installation managers and system programmers responsible for assessing the effort required to install an OS/VS2 Release 2 system. Readers should have a background in programming and maintaining computer operating systems such as OS/MVT or OS/VS2 Release 1.

This publication is for planning purposes only. The functions and capabilities described reflect the information that is currently available. This information is subject to change before the availability of OS/VS2 Release 2.

Throughout this publication the term MVS is used interchangeably with VS2 Release 2. MVS refers to the multiple virtual storage concept of this release.

This manual contains the following independent chapters:

- Introduction -- The Introduction highlights the major points that should be considered in planning for the installation and implementation of VS2 Release 2.
- Defining the System -- This chapter provides preliminary planning information for defining the type of system that an installation needs. For example, the chapter describes the VS2 Release 2:
  - procedures and macro instructions for system generation.
  - procedures and parameters for system initialization.
  - system libraries and data sets.

For each topic, a summary of the major changes from both MVT and VS2 Release 1 is provided.

- Directing the Use of System Resources -- This chapter describes some of the new VS2 Release 2 facilities that allow an installation to control the distribution of system resources among system users. It provides planning information that helps assess the effort involved in implementing these facilities.
- System Integrity -- This chapter discusses integrity in a multi-user computer system. It includes recommendations for maintaining, in any control program extensions or modifications, the same level of system integrity as that of the VS2 Release 2 system.
- Conversion Considerations -- This chapter presents preliminary descriptions of some of the most significant differences between VS2 Release 2 and MVT or VS2 Release 1. It provides the information to assess the extent of the modifications required for existing data sets, cataloged procedures, programs, data reduction routines, and operational procedures, to install VS2 Release 2.
- Appendix A: Virtual Storage Map -- This appendix shows the layout of virtual storage in VS2 Release 2.
- Glossary -- The glossary describes VS2 terms.

The following items are described in this publication strictly for advance planning purposes and will not be available with the initial release of VS2 Release 2:

- JES3 job entry subsystem.
- Virtual telecommunications access method (VTAM).

Additional information about these items will be available later. Availability dates for the support of the items may be obtained from the local IBM branch offices.

### **Prerequisite Publication**

#### **IBM System/370 Introduction to VS2 Release 2, GC28-0661**

This publication describes VS2 Release 2 features and facilities, design concepts, and basic hardware requirements.

### **Related Publications**

#### **Introduction to Virtual Storage in System/370, GR20-4260**

This publication provides an overview of the advantages and concepts of virtual storage systems.

#### **IBM Data Processing Glossary, GC20-1699**

This publication defines terms and concepts used in IBM publications.

#### **OS/VS Virtual Storage Access Method (VSAM) Planning Guide, GC26-3799**

This publication contains a detailed description of the virtual storage access method (VSAM).

#### **IBM System/370 OS/VS1 Planning and Use Guide, GC24-5090**

This publication describes OS/VS1. It can assist installation personnel in selecting and evaluating VS1 and can assist system programmers in implementing, modifying, and extending the capabilities of the VS1 control program.

#### **IBM System/370 OS/VS2 Planning and Use Guide, GC28-0600**

This publication describes OS/VS2 Release 1. It introduces VS2 concepts and provides planning information for users planning to have VS2 Release 1 installed.

#### **Introduction to VTAM, GC27-6987**

This publication describes the functions and capabilities that an installation needs to know to plan for installing the virtual telecommunications access method (VTAM).

# Contents

<b>Summary of Amendments</b> . . . . .	9
<b>Introduction</b> . . . . .	11
<b>VS2 Release 2 Highlights</b> . . . . .	12
Multiprocessing . . . . .	12
Job Entry Subsystems . . . . .	12
Data Set Handling . . . . .	12
System Resources Management . . . . .	13
System Activity Measurement Facility . . . . .	13
Allocation . . . . .	13
Multiple Virtual Address Spaces . . . . .	13
<b>MVS Planning Considerations</b> . . . . .	14
<b>Functions Not Supported</b> . . . . .	15
Unsupported Type I Programs . . . . .	17
<b>Defining the System</b> . . . . .	19
<b>System Generation Procedures</b> . . . . .	20
The VS2 Release 2 Starter System . . . . .	21
Minimum Configuration . . . . .	21
Physical Addresses for Devices . . . . .	21
Stage I Macro Instructions . . . . .	22
Job Entry Subsystem Generation . . . . .	23
JES2 Generation . . . . .	24
JES3 Generation . . . . .	25
RMTGEN . . . . .	25
System Generation Changes . . . . .	26
Initializing the New System Data Sets . . . . .	28
Adding Installation-Written Members to System Libraries . . . . .	28
Unsupported VS2 Release 1 Stage I Macro Instructions . . . . .	29
<b>System Initialization</b> . . . . .	30
<b>MVS System Parameters</b> . . . . .	30
Changes to System Parameters for MVS . . . . .	36
Changes to VS2 Release 1 System Parameters . . . . .	36
Unsupported VS2 Release 1 System Parameters . . . . .	37
Unsupported MVT System Parameters . . . . .	38
System Parameter Specification . . . . .	39
System Parameter Library (SYS1.PARMLIB) . . . . .	39
Changes to SYS1.PARMLIB Lists . . . . .	40
IBM-Created Lists . . . . .	41
Installation-Created Lists . . . . .	45
<b>System Libraries and Data Sets</b> . . . . .	47
Required Libraries and Data Sets . . . . .	47
Optional Libraries and Data Sets . . . . .	48
Changes to VS2 Release 1 System Libraries and Data Sets . . . . .	49
Changes to MVT System Libraries and Data Sets . . . . .	50
Adding Paging Space to the System . . . . .	50
Creating Temporary Data Sets in External Page Storage Using Virtual I/O . . . . .	51
<b>Directing the Use of System Resources</b> . . . . .	53
<b>System Resources Management</b> . . . . .	54
Workload Management Routines . . . . .	54
Resource-Use Routines . . . . .	54
Control Function . . . . .	56
Workload Management Concepts . . . . .	56
Transaction . . . . .	57
Performance Variable . . . . .	57
Service . . . . .	57
Service Rate . . . . .	57
System Workload . . . . .	58
Installation Performance Specification . . . . .	58
Performance Groups . . . . .	58
Performance Objectives . . . . .	59
Associating a Transaction With a Performance Objective . . . . .	61
The Relationship Between Workload Levels and Service Rates . . . . .	61
System Resources Manager Parameters . . . . .	65
IPS Parameters (IEAIPSxx) . . . . .	65

Tuning Parameters (IEAOPTxx) . . . . .	67
System Activity Measurement Facility (MF/1) . . . . .	68
MF/1 Operation . . . . .	68
MF/1 Reports and Tracing Records . . . . .	70
CPU Activity Report . . . . .	70
Paging Activity Report . . . . .	70
Workload Activity Reports . . . . .	71
Performance Group Period Report . . . . .	71
Performance Group Report . . . . .	71
System Summary Report . . . . .	72
Channel Activity Report . . . . .	72
I/O Device Activity Reports . . . . .	72
SMF Recording . . . . .	72
Tuning with MF/1 . . . . .	73
Tuning Example . . . . .	73
<b>System Integrity</b> . . . . .	79
Installation Responsibility . . . . .	80
Areas of Concern . . . . .	81
User-Supplied Addresses for User Storage Areas . . . . .	81
User-Supplied Addresses for Protected Control Blocks . . . . .	81
Resource Serialization . . . . .	82
Resource Identification . . . . .	83
SVC Routines Calling SVC Routines . . . . .	84
Control Program and User Data Accessibility . . . . .	84
Control Program Extensions . . . . .	85
Installation Procedural Responsibilities . . . . .	86
<b>Conversion Considerations</b> . . . . .	89
Job Entry Subsystem Considerations . . . . .	90
The JES2 Job Entry Subsystem . . . . .	90
JES2 Job Flow . . . . .	91
MVS Operational Changes . . . . .	93
JES2/HASP Compatability . . . . .	95
The JES3 Job Entry Subsystem . . . . .	96
JES3 Scheduling Features . . . . .	98
Job Definition . . . . .	98
JES3 Job Flow . . . . .	99
Device Scheduling . . . . .	100
Job Selection and Scheduling . . . . .	101
JES Operational Environment . . . . .	102
The Role of the JES3 Operator . . . . .	103
The Operator's Console . . . . .	103
Initialization, Control Statements, and Commands . . . . .	104
JES3/ASP Compatability . . . . .	104
SMF Considerations . . . . .	106
Differences From VS2 Release 1 System Management Facilities . . . . .	106
Differences From MVT System Management Facilities . . . . .	106
SMF Record Differences for VS2 Release 2 . . . . .	106
New System Resources Manager Data . . . . .	109
SMF Recording Replaced by MF/1 . . . . .	109
New Storage Data . . . . .	109
Job Entry Subsystem Statistics . . . . .	110
Multiprocessing Data . . . . .	110
Virtual Storage Access Method Data . . . . .	110
Time Sharing Record Elimination . . . . .	110
Changes to SMF Exits for VS2 Release 2 . . . . .	110
Job Control Language . . . . .	112
Changes to VS2 Release 1 JCL . . . . .	114
Performance Group Specification . . . . .	114
Assigning Job Selection Priority . . . . .	114
Specifying Data Sets in Paging Space (Virtual I/O Data Sets) . . . . .	114
Requesting Data Set Freeing at Data Set Close Time . . . . .	115
Specifying the Number of Resources That May Be Held for Reuse . . . . .	115
Requesting Multiple Copies of an Output Data Set . . . . .	115
Directing the Destination of an Output Data Set . . . . .	115
Support for Tape and Direct Access Devices . . . . .	116
Format Record Identification for the 3886 OCR Device . . . . .	116
Output Limit Specification . . . . .	116
Increased Data Set Definition Capability . . . . .	116
Increased Volume Sharing Capability . . . . .	116
Interpretation of the REGION Parameter . . . . .	116

Job Step Dispatching Priority Default Value . . . . .	117
Changes to MVT JCL . . . . .	117
Requesting Storage for Program Execution . . . . .	117
Specifying the BTAM Online Terminal Test Option . . . . .	117
JES2 JCL Considerations . . . . .	117
JES2 JCL for non-HASP Users . . . . .	117
JES2 Control Cards . . . . .	117
Assigning Job Selection Priority . . . . .	118
JES2 JCL for HASP Users . . . . .	118
/*JOBPARM Control Card . . . . .	118
/*OUTPUT Control Card . . . . .	118
JES3 JCL Considerations . . . . .	119
Communication Between Address Spaces . . . . .	120
Allocation Considerations . . . . .	121
Allocation Serialization . . . . .	121
Mount Characteristics . . . . .	121
User-Assigned Names . . . . .	121
Device Precedence List . . . . .	122
MVS Allocation Philosophy . . . . .	122
Reducing Allocation Contention Among Job Steps . . . . .	123
Operator Command Language . . . . .	124
Changes to Operator Commands . . . . .	124
Multiprocessing Support . . . . .	127
Diagnostic Support . . . . .	127
VTAM Support . . . . .	127
Time Sharing (TSO) Support . . . . .	128
System Resources Management Support . . . . .	128
Miscellaneous Support . . . . .	128
Job Entry Subsystem Support . . . . .	128
Time Sharing Considerations . . . . .	134
System Resources Management Support . . . . .	134
VSAM Support . . . . .	134
Allocation Support . . . . .	134
Job Entry Subsystem Support . . . . .	138
Edit Support . . . . .	138
Accounting Support . . . . .	138
Time Sharing Integration Support . . . . .	138
Catalog Support . . . . .	139
The VSAM Catalog . . . . .	140
VSAM Order of Search . . . . .	141
Creating or Updating the VSAM Catalog . . . . .	142
Updating an OS Catalog From a VSAM Catalog . . . . .	143
Catalog Utility and Macro Changes . . . . .	143
OS CVOL Support . . . . .	144
Defining an OS CVOL for MVS . . . . .	144
Accessing CVOL Catalogs . . . . .	145
Compatability for OS CVOL Requests . . . . .	145
Data Set Conversion . . . . .	146
Reformatting the UADS . . . . .	146
Formatting the Broadcast Data Set . . . . .	146
Programming Considerations With Virtual=Real Addresses . . . . .	146
Program Conversion . . . . .	148
TCAM Conversion . . . . .	148
Multiprocessing Considerations . . . . .	149
Hardware Configuration Considerations . . . . .	149
Operator Control of Configuration . . . . .	149
Recovery Considerations . . . . .	150
Alternate Path Retry . . . . .	150
Dynamic Device Reconfiguration . . . . .	150
Alternate CPU Recovery . . . . .	151
Manual Switching . . . . .	151
Programmer Considerations . . . . .	151
Loosely-Coupled Configurations . . . . .	152
<b>Appendix A: Virtual Storage Map . . . . .</b>	<b>153</b>
<b>Glossary . . . . .</b>	<b>155</b>
<b>Index . . . . .</b>	<b>161</b>

## Figures

Figure 1.	Unsupported MVT Functions (Part 1 of 2)	15
Figure 1.	Unsupported MVT Functions (Part 2 of 2)	16
Figure 2.	Unsupported VS2 Release 1 Functions	16
Figure 3.	Required Device Physical Addresses for the Starter System	22
Figure 4.	Generating a VS2 Release 2 System	24
Figure 5.	Changes in Stage I Macro Instructions	27
Figure 6.	System Libraries to Which Installation-Written Members May Be Added	29
Figure 7.	System Initialization Parameter Summary	31
Figure 8.	Changes to SYS1.PARMLIB Lists	40
Figure 9.	Performance Objective Definition	60
Figure 10.	Associating a Transaction With a Performance Objective	61
Figure 11.	Performance Objectives	62
Figure 12.	System Workload Levels	63
Figure 13.	Service Rates for Different Performance Objectives at Various System Workload Levels	64
Figure 14.	Tuning Example	74
Figure 15.	Changing a Performance Objective	75
Figure 16.	Effects of a Performance Objective Change	76
Figure 17.	JES2 Processing Stages	91
Figure 18.	JES2 Differences From MVT HASP and VS2 Release 1 HASP	95
Figure 19.	Sample JES3/ASP Multiprocessing Configuration	97
Figure 20.	JES3 Job Flow	99
Figure 21.	JES3 Differences From MVT ASP and VS2 Release 1 ASP	105
Figure 22.	Changes to SMF Records for VS2 Release 2 (Part 1 of 2)	107
Figure 22.	Changes to SMF Records for VS2 Release 2 (Part 2 of 2)	108
Figure 23.	Changes to SMF Exits for VS2 Release 2	111
Figure 24.	JCL Changes for VS2 Release 2	113
Figure 25.	Features Supported With JES3 Control Cards	119
Figure 26.	Communication Between Address Spaces	121
Figure 27.	Relationship Between User-Assigned and Generic Names	122
Figure 28.	Serialization Characteristics of Suggested Allocation Methods	123
Figure 29.	Changes to Operator Commands for VS2 Release 2 (Part 1 of 3)	125
Figure 29.	Changes to Operator Commands for VS2 Release 2 (Part 2 of 3)	126
Figure 29.	Changes to Operator Commands for VS2 Release 2 (Part 3 of 3)	127
Figure 30.	JES2 Command Functions	129
Figure 31.	JES3 Command Functions	130
Figure 32.	JES3 Control Command Functions	130
Figure 33.	Changes to Time Sharing Commands for VS2 Release 2 (Part 1 of 3)	135
Figure 33.	Changes to Time Sharing Commands for VS2 Release 2 (Part 2 of 3)	136
Figure 33.	Changes to Time Sharing Commands for VS2 Release 2 (Part 3 of 3)	137
Figure 34.	Simplified Catalog Structure - OS vs. VSAM	140
Figure 35.	Functions of the Access Method Services	142
Figure 36.	Method of Processing OS CVOL Requests	144
Figure 37.	Virtual Storage Map	153



## **Summary of Amendments for GC28-0667-1 VS2 Release 2**

### **Job Entry Subsystem**

The JES2 and JES3 job entry subsystems, generally compatible with HASP II and ASP Version 3 respectively, are described with particular reference to compatibility to other systems.

### **Allocation**

Allocation has been redesigned in an attempt to reduce contention for devices.

### **System Resource Management**

The workload management concepts are defined.

### **Virtual I/O**

I/O may be performed logically rather than physically, in a manner transparent to the user.

### **Communication Between Address Spaces**

The interregion communication of previous systems can occur with less movement of data.

### **Catalog Support**

OS control volumes (CVOLs) are supported under VS2 Release 2.

### **Diagnostic Support**

The system trace function is now available by command (TRACE).

### **V=R Programming Considerations**

Those considerations necessary for running with virtual addresses equal to real addresses are discussed.

### **Error Recovery Under Multiprocessing**

Manual switching, alternate path retry, dynamic device reconfiguration, and alternate CPU recovery can occur with multiprocessing.



## **Introduction**

OS/VS2 Release 2 (MVS) is a virtual storage operating system with multiprogramming, multiprocessing, time sharing (TSO), and job entry subsystems. In addition to providing significant new features, it also enhances existing OS/MVT and OS/VS2 facilities. OS/VS2 Release 2 is generally upward compatible with OS/MFT, OS/MVT, OS/VS1 and OS/VS2 Release 1.

## VS2 Release 2 Highlights

Some of the significant new features provided in MVS include multiprocessing, job entry subsystems, new data set handling facilities, a centralized resource management facility, a system activity measurement facility, a virtual I/O paging mechanism for temporary data sets, and multiple virtual address spaces. These are described in the following sections. For a more detailed description of these features and of the MVS system, see the **Introduction to VS2 Release 2**.

### ***Multiprocessing***

The incorporation of multiprocessing into the MVS control program provides for growth to larger and more complex systems. CPUs can be combined in two types of combinations: loosely coupled or tightly coupled. An installation can use combinations of loosely coupled and tightly coupled multiprocessing to tailor configurations to meet individual requirements.

In a tightly coupled multiprocessing system, two CPUs share real storage, operate under a single control program, and communicate with each other by means of an interrupt capability. The system control program treats the CPUs as resources, assigning them to process tasks. Either JES2 or JES3 can be used to provide support for tightly-coupled multiprocessing.

Loosely coupled multiprocessing systems connect VS2 Release 2 systems by means of channel to channel adapters that pass information between the processors. One processor controls job selection for all of the processors in the system. The JES3 job entry subsystem, a successor of ASP, provides the control for loosely coupled multiprocessing systems. One or more of the JES3 processors can, itself, be a tightly coupled multiprocessor.

### ***Job Entry Subsystems***

JES2 and JES3 provide the job processing functions of:

- Reading local and remote jobs into the system.
- Scheduling jobs.
- Maintaining all data submitted with jobs.
- Supporting the system management facilities.
- Handling all output from batch jobs and time-sharing users.

In addition, JES3 provides:

- Workload sharing by processors within the JES3 complex.
- Network job processing for workload sharing between remote VS2 systems.
- Pre-execution fetch and setup of removable I/O volumes.
- Scheduling based on job-completion deadline.
- Scheduling based on completion deadline.
- Scheduling based on completion of one or more jobs within a related set of jobs.

Either JES2 or JES3 may be specified during system generation to run as a component of the system control program.

## ***Data Set Handling***

The virtual storage access method (VSAM) is designed to offer function and flexibility to online and data base environments. In MVS, the key-sequenced VSAM system catalog structure is designed to reduce access time to very large catalogs. It permits the use of multiple private catalogs and the master catalog in one step, job, or session.

Virtual I/O (VIO), is a new way of processing temporary data sets. It is accomplished on a page basis in virtual storage.

The virtual telecommunications access method (VTAM) is a new direct-control teleprocessing access method with facilities available to applications programs, including those using TCAM.

## ***System Resources Management***

A new facility, called the system resources manager, is a centralized decision-maker that monitors a wide range of data about the condition of the system. It attempts to provide system resources to jobs in a manner specified by the installation. This allows an installation to specify that certain jobs get better response time than other jobs.

## ***System Activity Measurement Facility***

The system activity measurement facility (MF/1) is both a new and standard feature in MVS. It collects information about system activities and produces trace records and reports. MF/1 is basically system oriented (as opposed to SMF which is basically job oriented). It is designed to aid in analyzing both system performance and system trends and in anticipating future system requirements.

## ***Allocation***

A new allocation design allows an installation to assign names to subsets of devices and to define a precedence list for allocating device types. The intent of the design is to reduce contention for devices (serialization) and allocation time by tailoring the allocation process.

## ***Multiple Virtual Address Spaces***

In VS2, virtual storage is an address range of 16,777,216 bytes. Release 1 of VS2 provided one virtual address range; jobs were assigned regions from this address range, just as jobs were assigned regions in real storage in MVT. MVS provides multiple virtual address spaces; each user (i.e., batch job, time-sharing job, and certain system components) receives its own copy of virtual storage, minus the space used for certain system functions (e.g., the nucleus, the link pack area, the system queue area).

The new virtual storage design provides the potential for greater multiprogramming, including the concurrent execution of more large jobs and a greater mix of time-sharing and batch jobs. Also, since each job is isolated in a private address space, inter-region virtual storage fragmentation is eliminated and protection features are extended.

## MVS Planning Considerations

To make use of some of the new or changed MVS facilities, it will be necessary to modify some existing procedures, libraries, data sets, and programs. The following list highlights examples of some of the more significant Release 2 changes that may require modifications to be made by an installation. Neither the list nor the examples are meant to be exhaustive; rather they call attention to areas of change that should be considered when planning to install the MVS system.

The actual details of the changes are described in this publication in succeeding chapters.

- While the selection and specification of Stage I system generation options has been simplified, it is now necessary to select and specify job entry subsystem options.
- MVS system initialization is designed to enable minimum operator intervention by extending the options that can be specified in SYS1.PARMLIB.
- The system catalog structure is now a VSAM data set that can handle both VSAM and other data set types. Existing OS system catalogs (SYSCTLGs) are either supported as CVOLS, or can be converted to the new VSAM structure.
- The new multi-function access method services are used to convert an OS system catalog to a VSAM master catalog. Access method services also replace most of the catalog functions of system utilities such as IEHMOVE, IEHLIST, IEHPROGM.
- Paging space is contained in VSAM data sets in MVS. At least two page data sets with the required space must be defined during system generation.
- System resources management routines allow the specification of relative rates for system resource scheduling for various categories of batch and time sharing jobs. The installation identifies the categories by assigning each of them a performance group number to be specified in JCL. The installation specifies the distribution rates and other tuning options in SYS1.PARMLIB members.
- The new system activity measurement facility (MF/1) collects and records information about system activities in either the SMF data set or on formatted and printed reports. Some SMF data, such as CPU or paging activity is now recorded only by MF/1 on new SMF records.
- System management facilities (SMF) records have been extended to record data provided by new features such as the system resources manager, the virtual storage access method, or job entry subsystems. Data reduction routines may have to be revised to process new records and new or changed record fields.
- Job control language parameters have been added to specify items such as performance groups. Job entry subsystem control cards may be used to specify JES2 or JES3 job processing options.
- The operator commands have been extended to support multiprocessing and changed for communication with the job entry subsystems.
- The installation can name subsets of a generic set of devices in such a manner as to allow simultaneous allocation within the generic set.

## Functions Not Supported

Figure 1 shows significant MVT functions that are not supported in VS2 Release 2. Figure 2 shows unsupported VS2 Release 1 functions.

Unsupported MVT Function	Comparable VS2 Release 2 Function
Automatic SYSIN batching (ASB) reader BACKSPACE, NOTE, POINT, XDAP, and EXCP addressed to SYSIN/SYSOUT DCBs	Job entry subsystem (JES2 or JES3) -----
Conversational remote job entry (CRJE) Dedicated work files Direct system output (DSO)	Time sharing (TSO) Virtual I/O (VIO) -----
Graphic job processor (GJP) IEBUPDAT utility program IEHIOSUP utility program	----- IEBUPDTE utility program -----
IEHMOVE, IEHLIST, IEHPROGM (most catalog functions) IMBMDMAP	Access method services  AMBLIST
IMCJQDMP service aid IMCOSJQD service aid Main storage hierarchy support	Job entry subsystem and scheduler work area (SWA) dump routines ----- -----
OS readers and writers Queued telecommunications access method (QTAM)	Job entry subsystem (JES2 or JES3) and the external writer Telecommunications access method (TCAM)
QTAM message processing programs Remote job entry (RJE) Rollout/rollin Satellite graphic job processor (SGJP)	TCAM application programs JES2 or JES3 remote job entry Paging -----
Scatter load System environment recording routines (SER0 and SER1) OS system catalog (SYSCTLG)	Paging Recovery management support (RMS) VSAM master catalog with CVOL support

Figure 1. Unsupported MVT Functions (Part 1 of 2)

Unsupported MVT Function	Comparable VS2 Release 2 Function
SYS1.ACCT data set	System management facilities (SMF) data sets
SYS1.SYSJOBQE	Job entry subsystem input and output queues and the scheduler work area (SWA)
Tape output for SMF	----
TESTRAN program	----
Time slicing	System resources manager
Transient areas	Pageable link pack area
TSO driver	System resources manager
TSO trace	Generalized trace facility (GTF)

**Figure 1. Unsupported MVT Functions (Part 2 of 2)**

Unsupported VS2 Release 1 Function	Comparable VS2 Release 2 Function
BACKSPACE, NOTE, POINT, XDAP, and EXCP addressed to SYSIN/SYSOUT DCBs	----
Directed swap for TSO	----
IEHMOVE, IEHLIST, IEHPROGM (most catalog functions)	Access methods services
IMCOSJQD service aid	Job entry subsystem and SWA dump routines
Nucleus-only SYSGEN	----
OS readers and writers	Job entry subsystem and the external writer (JES2 or JES3)
OS system catalog (SYSCTLG)	VSAM master catalog With CVOL support
Priority I/O queueing and ordered seek queueing	----
QTAM message processing programs	TCAM application programs
SYS1.SYSJOBQE	Job entry subsystem input and output queues and the scheduler work area (SWA)
Time slicing	System resources manager
TSO driver	System resources manager
TSO link pack area	----
TSO trace	Generalized trace facility (GTF)

**Figure 2. Unsupported VS2 Release 1 Functions**



## ***Unsupported Type I Programs***

The following Type I programs which were supported by MVT and VS2 Release 1 are not supported by MVS (i.e., APARs written against these programs running on MVS will not be accepted). Corresponding program products are supported in their place. (Contact your local IBM branch office for information about these program products).

Even though the following Type I language compilers are not supported under MVS, generated object code for an installation's programs that use any of these higher-level Type I languages, together with their corresponding library modules, will operate on this release of OS/VS2 subject to the constraints outlined in the "Program Conversion" section of this publication.

Program Name	Program Number
COBOL F	360S-CB-524
COBOL F Library	360S-LM-525
Full ANS Cobol V2	360S-CB-545
Full ANS Cobol V2 Library	360S-LM-546
FORTTRAN G	360S-FO-520
FORTTRAN H	360S-FO-500
FORTTRAN G and H Library	360S-LM-501
PL/1 F	360S-NL-511
PL/1 F Library	360S-LM-512
PL/1 F Syntax Checker	360S-PL-552
SORT/MERGE	360S-SM-023



## **Defining the System**

During system generation and system initialization, the installation can define or change the operation of a standard or optional feature of the system control program to meet specific needs. This chapter provides planning information for defining the system. It is divided into the following sections:

- System Generation Procedures
- System Initialization
- System Libraries and Data Sets

## System Generation Procedures

System generation is the process of selecting data from IBM-distributed libraries and tailoring it to create an operating system for an installation.

An I/O device-only system generation can also be performed. It is performed when only I/O devices and channels are to be added, deleted, or modified. The processor-only generation available with MVT and the nucleus-only generation available with both MVT and VS2 Release 1 are not available with MVS.

The MVS system generation process is simplified over the MVT and VS2 Release 1 system generation processes in the following ways:

- Many previously optional facilities are now standard. Therefore, fewer macro instructions and parameters need to be specified.
- Several user-specified macro instructions have been clarified by consolidating parameters for related options under a single macro instruction.
- Additional system initialization lists are available to be preformatted and taken from the SYS1.PARMLIB data set for use at IPL time.

For MVS, IBM will generate (from installation-provided system generation statements) and install one system control program per uniprocessing or multiprocessing configuration. The IBM installation process also includes performing the installation verification procedure (IVP).

There are six major steps in generating a new operating system.

### Planning

The distribution libraries containing the MVS modules must be ordered from IBM. The options must be selected that, with the standard features, will constitute the desired system. The macro instructions needed to specify the options and standard features of the new system must be selected and coded.

### Volume and data set initialization

The direct access volumes that will contain the distribution libraries (and the starter system) must be initialized. The volumes on which the new system will be generated must also be initialized. The data sets on the system volumes must be allocated (unless they will be specified by SYSGEN macros to be assembled in Stage I for allocation in Stage II).

### Stage I

The macro instructions are assembled and expanded into job control language, utility control statements, assembler language statements, and linkage editor control statements.

### Stage II (Jobstream Execution)

The modules from the distribution libraries are assembled, link-edited, and copied to the data sets that have been allocated on the new system volume(s).

### Job entry subsystem generation

The job entry subsystem modules can be assembled (JES2 only) and link-edited to reflect the options and limits specified for them. Job entry subsystem generation may be performed concurrently with Stage II. Additionally,

remote station programs may optionally be generated (RMTGEN). For JES2 (most) and for JES3 (all) these options can be specified during initialization.

### **Testing**

After the new system is generated, the IBM Program Systems Representative uses the installation verification procedure (IVP) to verify that the new system is operational on the hardware configuration.

## ***The VS2 Release 2 Starter System***

The first time that MVS is generated, the IBM starter system is required to drive the system generation processing. The starter system consists of an MVS operating system that uses JES2 as the job entry subsystem. After the first system generation, either the starter system or the new MVS control program can be used to drive subsequent system generations.

### **Minimum Configuration**

Use of the starter system requires the following minimum configuration:

- One IBM System/370 Model 145, 155II, 158, 165II, or 168. (The clock comparator, the CPU timer feature (#2001) and the advanced control program support feature (#1001) are required on the Model 145.)
- 768K bytes of real storage.
- One multiplexer channel.
- One selector or block multiplexer channel.
- Three IBM 3330 spindles, or four IBM 2314/2319 Direct Access Storage Facility devices.
- One card reader.
- One hardcopy console or one hardcopy device with a display console.
- One 9-track magnetic tape.
- One 1403 or 3211 printer.

### **Physical Addresses for Devices**

The starter system is configured to support the devices at the addresses specified in Figure 3.

Function	Device	Multiplexer Channel	Channel 1	Channel 2	Channel 3	Channel 4
Console	158 Console	014, 016				
	3213	015, 017				
	3210/3215	009, 01F		209, 21F		
	3066	040, 060 0C0, 0D0 0E0, 0F0	140	240	340	440
Reader	2540	00C		20C		
	3505	012		20A		
Punch	2540	00D		20D		
	3525	013		20B		
Tape	2400 (7-TR-DC)		180, 181	280, 281	380, 381	480, 481
	2400 (9-TR) 3400		182, 183 184	282, 283 284	382, 383	482, 483
Printer	3211	002, 004				
	1403	00E, 00F		20E		
Direct Access Storage Device	2305-1		1F0	2F0		
	2305-2		1D0	2D0		
	2314/2319		130, 131 132, 133 134, 135	230, 231 232, 233 234, 235	330, 331 332, 333	
	3330		150, 151 152, 153	250, 251 252, 253	350, 351 352, 353	
	3330-1		158, 159 15A, 15B	258, 259 25A, 25B	358, 359 35A, 35B	
	3340		1C0, 1C1 1C2, 1C3	2C0, 2C1 2C2, 2C3	3C0, 3C1 3C2, 3C3	

Figure 3. Required Device Physical Addresses for the Starter System

### Stage I Macro Instructions

The Stage I macro instructions are used to tailor the new system. They describe the machine configuration, the control program, the data management routines, and the user-written routines. They are also used to specify the program options that are to be included in the system.

The macro instructions that can be specified in Stage I of MVS are:

#### AFFINITY

specifies program names (e.g., emulators) and the identification of the CPU on which each program is required to execute.

#### CHANNEL

specifies the channel characteristics. A CHANNEL macro instruction is required for each channel in the installation's computing system.

**CKPTREST**

specifies standard system completion codes not eligible for automatic restart, and user completion codes that are eligible for automatic restart. If the CKPTREST macro instruction is not specified, the standard set of codes will be used.

**CONSOLE**

specifies the console configuration for the system.

**CTRLPROG**

specifies control program options. If the CTRLPROG macro instruction is not specified, the default values are assumed.

**DATAMGT**

specifies optional access methods (BTAM, ISAM, GAM, TCAM and VTAM) to be included in the system.

**DATASET**

specifies system data sets and the volumes on which they reside. It is required for each defined data set that is not located on the system residence volume. It is also used to specify user-written macros, routines, and procedures to be added to system libraries.

**EDIT**

specifies the physical characteristics and processing attributes of the data sets to be processed by the time sharing EDIT command.

**GENERATE**

specifies the data sets, volumes, and I/O devices required for the SYSGEN process, the SYSGEN output options, and the type of generation being performed.

**IODEVICE**

specifies characteristics of an I/O device and its operating system requirements. An IODEVICE macro instruction is required for each uniquely addressable I/O device.

**SCHEDULR**

specifies job and master scheduler options.

**SVCTABLE**

specifies the number, type, APF authorization, and entry status of the user-written SVC routines to be added to the new system.

**TSO**

specifies the inclusion of IBM-supplied time sharing command processors.

**UNITNAME**

specifies a name for a group of I/O devices and optionally makes the name available for VIO use. A UNITNAME macro instruction is required for each named group of I/O devices in the system, except for unique device types.

### ***Job Entry Subsystem Generation***

Job entry subsystems (JES2 or JES3) are added to the system through the process of job entry subsystem generation, which may be executed concurrently with Stage II. Figure 4 summarizes the steps that are involved in performing a job entry subsystem generation and relates them to the steps that are required to perform a system generation (SYSGEN).

Step	SYSGEN	JES2 Generation	JES3 Generation
Planning (selecting options)	x	x	
Volume and data set initialization	x	x	x
Stage I	x		
Stage II (jobstream execution) Stage II and JES2 generation may be executed concurrently.	x	x	x
RMTGEN (optional)		x	x
Testing	x	x	x

Figure 4. Generating a VS2 Release 2 System

## JES2 Generation

The distribution libraries needed to perform a JES2 generation are distributed by IBM along with the MVS distribution libraries. They contain:

- The JES2 source code.
- The utilities used by the JES2 jobstream.

During JES2 generation, the JES2 control program is assembled to reflect the options and limits that the installation has chosen. The installation takes the following steps to perform a JES2 generation.

### Planning

The JES2 and optional RMTGEN parameters must be chosen and coded.

### Volume and data set initialization

The space required for JES2 data sets must be allocated on the new system volumes.

### Jobstream Execution

The jobstream on the distribution library must be executed to apply options, assemble, link-edit, and copy the JES2 modules from the distribution library to the data sets that have been allocated on the new system volumes. The jobstream also invokes RMTGEN if any remote terminals were specified by the JES2 generation parameters.

**JES2 Generation Options:** The following characteristics of the JES2 control program may be specified:

- The maximum number of jobs, started tasks, and time sharing users in the JES2 subsystem at any one time.
- The number of card readers, printers, and card punches to be used by the job control manager.
- The maximum number of initiators that may be started at any one time.
- The default limits for a job's printed or punched output and the actions to be taken when the specified limits are exceeded.



- The type of processing for SYSOUT data sets.
- The number of direct access storage volumes that may be mounted concurrently as spooling devices.
- Many parameters (or their equivalents) formerly specified only during HASPGEN, can now be specified during JES2 initialization.

### JES3 Generation

The initial release of JES3 is distributed as a separately orderable independent component. It contains:

- A source library containing all JES3 programs and macros.
- An object deck library containing assembled JES3 modules suitable for link editing.
- A sample program library containing test decks and such procedures as the method for adding the JES3 ICR to the existing MVS system.
- A sample JES3 initialization deck.

Because JES3 options and limits are selected at initialization time, it is not necessary to specify them at system generation. RMTGEN parameters must be coded.

### RMTGEN

In addition to generating a job entry subsystem, one or more remote station programs may also be generated. The process of generating job entry subsystem multileaving remote terminal programs is called RMTGEN. The parameters used during RMTGEN specify hardware configuration and software options. Remote terminal programs that can be generated by RMTGEN include:

- System/360 and System/370 binary synchronous communication (BSC) remote terminal program.
- System/360 model 20 remote terminal program.
- 1130 remote terminal program.
- 1130 remote terminal program load program.
- System/3 remote terminal program.

**Note:** Remote terminals can be supported only by 2701 and 2703 emulators on the 3704 and 3705.

**RMTGEN Options:** The following remote terminal program characteristics may be specified:

- The type and degree of text compression.
- The amount of main storage available to the program.
- The number of buffers to be used by the program.
- The inclusion or exclusion of unit record devices.
- The unit address of the BSC adapter and devices.
- The speed of the communication line used.

- The number of buffer areas to be used for restoring compressed lines to their original form.
- The type of assembly listing to be produced during RMTGEN.

### ***System Generation Changes***

Some of the major changes to the system generation processes for MVT and VS2 Release 1 include:

- Changes to the process of initializing new system data sets.
- The reduction of the number of macro instructions that need to be specified during Stage I (see Figure 5). This reduction is due both to the fact that previously optional facilities are now standard and to the consolidation of related options under single macro instructions in MVS. For example, it is no longer necessary to specify two different SYSGEN macro instructions (RESMODS and LINKLIB) to add installation-written routines to the nucleus and the link library. The parameters on these macro instructions have been consolidated under the DATASET macro instruction which is now used to specify the addition of installation-written routines to any system library. (See the topic "Adding Installation-Written Members to System Libraries.")
- The addition of a job entry subsystem generation.
- The addition of parameters for specifying authorized appendages.

MVT	VS2 Release 1	VS2 Release 2
—	—	AFFINITY
ALGLIB	***	***
ALGOL	***	***
ASSEMBLR	***	***
CENPROCS	CENPROCS	—
CHANNEL	CHANNEL	CHANNEL
CHECKER	***	***
CKPTREST	CKPTREST	CKPTREST
CMDLIB	DATASET	DATASET
COBLIB	***	***
COBOL	***	***
CTRLPROG	CTRLPROG	CTRLPROG
DATAMGT	DATAMGT	DATAMGT
DCMLIB	DATASET	DATASET
EDIT	EDIT	EDIT
EDITOR	EDITOR	—
EMULATOR	—	—
FORTLIB	***	***
FORTRAN	***	***
GENERATE	GENERATE	GENERATE
GENTSO	—	—
GJOBCTL	—	—
GRAPHICS	GRAPHICS	DATAMGT
HELP	DATASET	DATASET
IMAGELIB	DATASET	DATASET
IOCONTRL	—	—
IODEVICE	IODEVICE	IODEVICE
LINKLIB	LINKLIB	DATASET
LOADER	LOADER	—
MACLIB	MACLIB	DATASET
OUTPUT	TSO	TSO
PARMLIB	PARMLIB	DATASET
PL1	***	***
PL1LIB	***	***
PROCLIB	DATASET	DATASET
PTOP	***	***
RESMODS	RESMODS	DATASET
RPG	***	***
SCHEDULR	SCHEDULR	SCHEDULR
SECMODS	CENPROCS	—
SECONSLE	SECONSLE	CONSOLE
SORTLIB	***	***
SORTMERG	***	***
SUPRVSOR	—	—
SVCLIB	DATASET	DATASET
SVCTABLE	SVCTABLE	SVCTABLE
SYSUTILS	***	***
TELCMLIB	DATASET	DATASET
TSOOPTION	TSO	TSO
UCS	UCS	—
UNITNAME	UNITNAME	UNITNAME
—	PAGE	DATASET
<p>— no longer specified during system generation</p> <p>*** the function specified by this macro instruction may be added to the system after system generation</p>		

Figure 5. Changes in Stage I Macro Instructions

## Initializing the New System Data Sets

New system data sets must be initialized before Stage II is executed. Initialization includes:

- allocating space for system data sets on the new system volumes.
- building the volume index of the system catalog.
- cataloging the data sets in the new system catalog.

In MVS, the new system data sets can be initialized either during system generation by using the DATASET macro instruction or at any other time before Stage II by using the access method services.

The Stage I DATASET macro instruction did not exist in MFT or MVT. Its use for initializing new system data sets during SYSGEN remains unchanged from VS1 Release 1 and VS2 Release 1. A DATASET macro instruction is specified with the SPACE parameter in the Stage I input deck for every system data set that is to be allocated. The new system data sets are then allocated and cataloged during Stage II execution.

In MVS, the IEHPROGM utility program can be used to catalog only non-VSAM data sets without CVOL pointers. In order to preallocate all new system data sets (VSAM and non-VSAM data sets) the access method services should be used. In this case a DATASET macro instruction should be specified without the SPACE parameter for any system data set used during system generation that is not on the system residence volume. This enables the catalog entry to be created during Stage II.

## Adding Installation-Written Members to System Libraries

The DATASET macro instruction is used during SYSGEN to include installation-written members in system libraries. The parameters:

```
PDS=SYS1.name  
MEMBERS=( name, name... )
```

are functionally equivalent to their counterparts on the RESMODS and LINKLIB SYSGEN macro instructions in MVT and VS2 Release 1. A new feature is provided for adding installation-written routines to the nucleus, however. If only the PDS parameter is specified, the members in the data set specified by PDS will be copied into the nucleus as individual members. If both the PDS and MEMBERS parameters are specified, the members in the data set specified by PDS will be copied into the nucleus and link-edited together with the nucleus member IEANUC01. Figure 6 indicates the system libraries to which installation-written members may be added and the form in which they must be added.

System Library	Form
SYS1.COMDLIB	load module
SYS1.IMAGELIB	load module
SYS1.LINKLIB	load module
SYS1.LPALIB	load module
SYS1.MACLIB	system macro
SYS1.NUCLEUS	load module
SYS1.PARMLIB	system parameter
SYS1.PROCLIB	system procedure
SYS1.TELCMLIB	load module
SYS1.VTAMLIB	load module
SYS1.VTAMLST	system parameter

Figure 6. System Libraries to which Installation-Written Members May Be Added

### Unsupported VS2 Release 1 Stage I Macro Instructions

The following VS2 Release 1 macro instructions are not supported in MVS.

#### CENPROCS

Model dependent information is no longer needed during system generation processing.

#### EDITOR

All functions provided by this macro instruction are available in SYS1.PROCLIB cataloged procedures.

#### GRAPHICS

The DATAMGT macro instruction now performs the function.

#### LINKLIB

The DATASET macro instruction now performs the function.

#### LOADER

Options specified by this macro instruction are now available in SYS1.PROCLIB cataloged procedures.

#### MACLIB

The DATASET macro instruction now performs the function.

#### PAGE

The DATASET macro instruction now performs the function.

#### PARMLIB

The DATASET macro instruction now performs the function.

#### RESMODS

The DATASET macro instruction now performs the function.

#### SECONSLE

The CONSOLE macro instruction now performs the function.

#### UCS

The options specified by this macro instruction are standard in the MVS system.

## **System Initialization**

Before any job can be processed, the control program and its associated control blocks and work areas must be loaded into storage and prepared for operation. Loading these control program modules is the function of the initial program loader (IPL).

After the IPL program completes its loading functions, control passes to the nucleus initialization program (NIP), which performs more functions necessary for the operation of the control program. (For example, NIP defines storage areas and initializes certain tables, work areas, and control blocks.) NIP also loads and initializes routines (such as fixed LPA routines) selected by the user. The nucleus initialization program also initializes real storage. It gives the installation control over virtual storage definition by processing installation-supplied system parameters that specify virtual storage options.

Master scheduler initialization completes the initialization process by establishing system tasks such as the console communications task, the log task, job entry subsystems, system management facilities, the recovery/termination management task and the missing interruption checker task.

In MVS, changes have been made to the system initialization process to provide the installation with greater flexibility in specifying system options as well as the opportunity to simplify the IPL process. These changes include:

- The reduction of informational IPL messages.
- The elimination of the time-of-day (TOD) clock messages and associated replies, unless the installation requests that they be issued or unless there is an error in TOD clock processing.
- The implementation of multiple SYS1.PARMLIB members for SMF, volume attributes, commands, and the link library.
- The creation of a SYS1.PARMLIB member that contains installation-specified operator commands that are issued by the system control program at the completion of system initialization processing. The operator need respond only in case of error conditions.

### ***MVS System Parameters***

The installation uses system parameters to control the system initialization process. System parameters may be specified through the master console or in the IEASYSxx lists in the system parameter library, SYS1.PARMLIB. Each parameter in a SYS1.PARMLIB list is specified in the same format as that required by the "SPECIFY SYSTEM PARAMETERS" message. Figure 7 provides a summary of these parameters. More detailed descriptions of the parameters follow.

Parameter	Value Specified		SYS1.PARMLIB List Read
APF	Authorized library name	⊙	IEAAPFxx
APG	Automatic priority group for system resources manager	•	
BLDL	Pageable directory for SYS1.LINKLIB		IEABLDxx
BLDLF	Nonpageable directory for SYS1.LINKLIB	•	IEABLDxx
CLPA	New link pack area to be created	•	IEALOD00 IEAPAK00
CMD	Command to be issued internally	⊙	COMMNDxx
CSA	Size of the common service area	⊙	
CVIO	Delete all VIO data sets from paging space	⊙	
DUMP	Data sets for SYS1.DUMP		
FIX	Reenterable routines for nonpageable LPA	•	IEAFIXxx
HARDCPY	Hard copy log		
IPS	Installation performance specification	⊙	IEAIPSxx
LNK	Names of data sets concatenated to SYS1.LINKLIB	⊙	LNKLSTxx
LOGCLS	Output class for log data set	⊙	
LOGLMT	WTL limit for log data set	⊙	
MAXUSER	Maximum number of virtual address spaces	⊙	
MLPA	Modifications to pageable LPA	•	IEALPAxx
NUCMAP	New nucleus map for DSS	⊙	
OPI	SYS1.PARMLIB operator intervention restrictions	•	
OPT	System resources manager tuning parameters	⊙	IEAOPTxx
PAGE	Page data set names	•	
REAL	V = R address area size	•	
SMF	SMF parameters	⊙	SMFPRMxx
SQA	Size of the system queue area		
SYSP	System parameter list to be merged with IEASYS00	•	IEASYSxx
VAL	Volume characteristics	⊙	VATLSTxx
VRREGN	Default region size for a V=R request	⊙	
WTOBFRS	Number of buffers for WTO routine use	⊙	
WTORPLY	Number of operator reply elements for WTOR routine use	⊙	
<ul style="list-style-type: none"> <li>• — Not Included in MVT Systems</li> <li>⊙ — Not Included in VS2 Release 1 Systems and MVT Systems</li> </ul>			

Figure 7. MVS System Initialization Parameter Summary

**APF**

specifies the value that is appended to IEAAPF to form the name of the SYS1.PARMLIB list that contains data set names with their corresponding volume identifications of authorized program libraries.

The SYS1.LINKLIB and SYS1.SVCLIB libraries are automatically included as authorized libraries.

**APG**

specifies an automatic priority group for use by the system resource management routines. If the APG parameter is not specified, a default value of 7 will be used.

**BLDL**

specifies a value that is appended to IEABLD to form the name of a SYS1.PARMLIB list that contains the names of modules for which entries are to be built in the pageable BLDL table.

The directory created in response to the BLDL parameter will reside in pageable storage.

If the BLDL and BLDLF parameters are both specified, the BLDL parameter will be ignored. A warning message will be issued to identify this condition.

The IEABLD00 list is the default list.

**BLDLF**

specifies a value that is appended to IEABLD to form the name of a SYS1.PARMLIB list that contains the names of modules for which entries are to be built in the resident BLDL table.

The directory created in response to the BLDLF parameter will reside in fixed storage.

If the BLDLF and BLDL parameters are both specified, the BLDLF parameter will be accepted. A warning message will be issued to identify this condition.

The IEABLD00 list is the default list.

**CLPA**

specifies that a new pageable link pack area is to be created.

If a previously created link pack area exists, it is deleted and the space it occupied is made available for system paging use.

If CLPA is specified, CVIO is automatically specified.

**CMD**

specifies a value that is appended to COMMND to form the name of the SYS1.PARMLIB list that contains the commands that the installation selects to be issued internally as soon as the system initialization process is complete.

If not specified in a system parameter list of by the operator, the COMMND00 list is used.

**CSA**

specifies the common service area.

The value specifies the size, in 1K byte multiples, of the common service area.

If CSA is not specified, the default value of 100K bytes is used.



### CVIO

for system restart, specifies that all virtual I/O (VIO) data sets are to be deleted from the paging space.

CVIO is automatic if CLPA is specified.

### DUMP

specifies tape volumes and/or the direct access storage devices to be used for the SYS1.DUMP data sets.

### FIX

specifies a value that is appended to IEAFIX to form the name of the SYS1.PARMLIB list that contains the names of reenterable routines from the SYS1.LINKLIB, SYS1.LPALIB, and SYS1.SVCLIB data sets to be made resident as a fixed extension to the link pack area (fixed link pack area).

If not specified, a fixed link pack area is not created.

### HARDCPY

specifies that the hard copy log is desired, whether or not the system log is to be used as the hardcopy log, and whether or not messages are to be recorded on the hardcopy log.

If the console configuration contains an active graphic console or more than one active console, a hardcopy log is required. In this case, if routing codes 1,2,3,4,7,8, or 10 are desired, they need not be specified since all of these codes are automatically assigned by the system control program.

### IPS

specifies a value that is appended to IEAIPS to form the name of the SYS1.PARMLIB member that contains the installation performance specification (IPS).

The IEAIPS00 is the default list.

### LNK

specifies a value that is appended to LNKLST to form the name of the SYS1.PARMLIB list that contains the names of the data sets to be concatenated to SYS1.LINKLIB.

Any number of SYS1.PARMLIB lists may be specified in this manner until the number of data sets contained reaches 15. (The maximum number of data set names other than SYS1.LINKLIB is 15. This allows 16 data sets including SYS1.LINKLIB to be concatenated.) SYS1.LINKLIB is opened first. Then each data set that is specified is opened and concatenated in the order in which it appears (i.e., left to right).

The LNKLST00 list is the default list.

### LOGCLS

specifies the JES2 output class for the log data sets.

If the LOGCLS parameter is not specified, a default class of A is used.

### LOGLMT

specifies the number of write-to-log (WTL) instructions that may be issued before the log data set is scheduled for output processing.

If the LOGLMT parameter is not specified, a default value of 500 WTL instructions is used.

### **MAXUSER**

specifies the maximum number of independent virtual address spaces available for use by the operating system.

If the MAXUSER parameter is not specified, a default value of 256 is used. The maximum number that may be specified is 1,536 minus the number of active VIO data sets.

### **MLPA**

specifies a value that is appended to IEALPA to form the name of the SYS1.PARMLIB list that contains the reenterable routines from SYS1.LINKLIB, SYS1.LPALIB, and SYS1.SVCLIB data sets to be made resident as a pageable extension to the link pack area (modified link pack area).

If the MLPA parameter is not specified, a modified link pack area is not created.

### **NUCMAP**

specifies the creation of a new map of the nucleus in the SYS1.DSSVM data set and the deletion of the existing nucleus map.

The parameter must be specified if the nucleus for which a map exists is modified and an IPL is performed using the modified nucleus. The NUCMAP parameter need not be specified again until another modification is made to the nucleus.

A new nucleus map will be created without the specification of the NUCMAP parameter, if the name of the nucleus load module changes between IPL procedures or if a nucleus map does not exist for the first IPL after system generation.

### **OPI**

specifies operator intervention restrictions of the system parameters in the SYS1.PARMLIB lists.

The OPI parameter can only be included in SYS1.PARMLIB lists. It may be specified for individual system parameters or for the entire set of system parameters.

### **OPT**

specifies a value that is appended to IEAOPT to form the name of the SYS1.PARMLIB list that contains tuning parameters for the system resources management routines.

If not specified in a system parameter list or by the operator, default values provided by the system resources management routines are used.

### **PAGE**

specifies one or more page data set names to be used for a particular system IPL, in addition to the page data set names in the PAGE parameter specification of PARMLIB members IEASYS00 or IEASYSxx. The first data set name is used to contain as much of the PLPA as space permits.

The values specified include identification of the VSAM data sets for which space was previously allocated and formatted (and, optionally, the unit on which the volume containing the specified data sets should reside).

## **REAL**

specifies nonpageable storage (virtual equals real address area).

The value specified is the number of 1K byte blocks to be reserved for nonpageable (V=R) storage.

If the value specified is 0, there will be no V=R storage in the system.

If the REAL parameter is not specified, a default of 76K will be used.

## **SMF**

specifies a value that is appended to SMFPRM to form the name of the SYS1.PARMLIB list that contains system management facility (SMF) values.

If not specified in a system parameter list or by the operator, the SMFPRM00 list is used.

## **SQA**

specifies the system queue area.

The value specified is the number of 64K byte segments to be added to the minimum size of the system queue area.

If not specified in a system parameter list or by the operator, a default value of one (64K byte segment) is added to the minimum size of the system queue area.

If the specified amount of virtual storage is not available, a warning message will be issued. The SQA parameter may be respecified at that time.

The size of the system queue area cannot be changed during a restart that reuses the previously initialized link pack area. If the SQA parameter is specified on the restart and the size is different, the system uses the previous size and issues a warning message.

## **SYSP**

specifies the SYS1.PARMLIB list of system parameters that is to be merged with the default list of system parameters, IEASYS00.

The value specified is appended to IEASYS to form the name of a SYS1.PARMLIB list that contains the system parameters to be used for the current IPL. Any number of IEASYSxx lists may be specified.

If not modified by this parameter, only the IEASYS00 list is used. The IEASYS00 list contains the values selected during system generation for various system parameters unless the contents of the list have been modified by the system programmer at the installation.

When multiple IEASYSxx lists are specified by the SYSP parameter, they are treated in ascending priority sequence (i.e., parameters defined in a member specified near the beginning of the list will be replaced by the same parameters defined in a member specified near the end of the list). In the process of merging parameters with the IEASYS00 list, IEASYS00 is always read, and always assumes the lowest priority.

For example, if the operator responds:

```
R 00, 'SYSP=(01,02)'
```

and these two IEASYSxx lists contain the following system parameters:

```
IEASYS01: MLPA=(00,01),BLDL=00
```

```
IEASYS02: MLPA=02,SQA=10
```

then the effective system parameters will be:

```
MLPA=02, BLDL=00, and SQA=10
```

#### VAL

specifies a value that is appended to VATLST to form the name of the SYS1.PARMLIB list that contains mount and allocation characteristics for any or all of the direct access volumes used by the installation.

If not specified in a system parameter list or by the operator, the VATLST00 list is used.

#### VRREGN

specifies nonpageable storage (virtual equals real address area) for a specific job.

The value specified becomes the default value for the real main storage size (in 1K multiples) to be allocated to a job requiring V=R address space.

If the VRREGN parameter is not specified, a default value of 64K is used.

The VRREGN system parameter should not be confused with the REAL system parameter. REAL specifies the total amount of nonpageable (V=R) storage available to the system. VRREGN specifies the subset of the total space that is available for an individual job.

#### WTOBFRS

specifies WTO buffers.

The value specifies the number of buffers to be used by the write-to-operator (WTO) routines.

If the WTOBFRS parameter is not specified, a default value of 20 is used.

#### WTORPLY

specifies WTOR elements.

The value specifies the number of operator reply elements to be used by the write-to-operator-with-reply (WTOR) routines.

If the WTORPLY parameter is not specified, a default value of 5 is used.

### Changes to System Parameters for MVS

Figure 7 shows new system parameters for MVS. Some VS2 Release 1 and some MVT system parameters are not supported. Of these, some have been replaced by new parameters and others have been eliminated, either because their functions are not supported in MVS or because their functions are now performed by the system control program.

**Changes to VS2 Release 1 System Parameters:** The following VS2 Release 1 system parameters have changed for MVS.

## APG

The only value that can be specified for the APG parameter in MVS is the priority of the automatic priority group.

## DUMP

Ten SYS1.DUMP data sets can be specified in MVS.

## PAGE

Values specified by the PAGE parameter have changed for MVS because paging space is made up of preformatted VSAM data sets, two of which must be defined and formatted during system generation, (See the topic "Adding Paging Space to the System" in this chapter.) The PAGE parameter also no longer specifies a particular volume to contain the link pack area.

## REAL

In MVS, the value specified by the REAL parameter represents the total amount of V=R storage in 1K byte multiples, as opposed to the number of 4K byte blocks added to a 64K byte minimum value in VS2 Release 1.

**Unsupported VS2 Release 1 System Parameters:** The following VS2 Release 1 system parameters are not supported in MVS but are provided for by comparable functions.

## LSQACEL

This parameter is not needed in MVS because quickcells for the local system queue area are controlled by virtual storage management routines.

## PAL

The values specified by this parameter are supplied by the address space supervision routines in MVS.

## SQACEL

This parameter is not needed because quickcells for the system queue area are controlled by virtual storage management routines.

## TMSL

This parameter is not needed because the system resources management routines determine the acceptable time slices.

The following VS2 Release 1 system parameters are not supported in MVS and have no comparable MVS functions.

## AUXLIST

This parameter is not needed because in MVS it is not meaningful to distinguish between the auxiliary storage available for background and time-sharing jobs.

## CPQE

The number of channel programs to be used by the address space supervision routines is no longer an installation option.

## MPA

It is not necessary to specify the size of the master scheduler region because in MVS, the master scheduler task has its own independent virtual address space.

## TSOAUX

It is not necessary to specify back-up auxiliary storage space for time sharing jobs in MVS, because each time sharing user is assigned to an individual virtual address space.

**Unsupported MVT System Parameters:** The following MVT system parameters are not supported in MVS, but are provided for by comparable MVS functions.

**MIN**

This parameter is not needed in VS2 because the initiator modules are contained in the pageable link pack area.

**MOD**

The CPU model identification is determined in VS2 by the STIDP instruction.

**RAM**

Routines from SYS1.SVCLIB and SYS1.LINKLIB can be added to the VS2 link pack area by means of the SYS1.LPALIB data set or the MLPA or FIX system parameters.

**RERP**

In VS2, error recovery procedure (ERP) routines are always resident in the link pack area.

**RSVC**

In VS2, SVC routines are always resident in the link pack area.

**SQS**

This parameter is replaced by the SQA system parameter.

**TMSL**

This parameter is not needed because the system resource management routines determine the acceptable time slices.

The following MVT system parameters are not supported in VS2 Release 2 and have no comparable MVS functions.

**ALTSYS**

This parameter is not needed because the system residence devices are treated by dynamic device reconfiguration (DDR) as any other swappable device.

**HRAM**

MVT hierarchy support is not needed in VS2.

**HSVC**

MVT hierarchy support is not needed in VS2.

**MPS**

It is not necessary to specify the size of the master scheduler region because in MVS, the master scheduler task has its own independent virtual address space.

**QBF**

This parameter is not needed in MVS because the job queue is replaced by the scheduler work area (SWA) and job entry subsystem data sets.

### **System Parameter Specification**

An installation can tailor a system through the specification of system parameters, either during system generation (SYSGEN) processing or in the IEASYS00 or IEASYSxx lists of SYS1.PARMLIB. The system operator specifies

system parameters through the master console in response to the "SPECIFY SYSTEM PARAMETERS" message. Proper use of IEASYS00 or IEASYSxx can result in a significant saving of time.

The IEASYS00 (system parameter) list is always created during the SYSGEN process, and the system parameters HARDCPY, PAGE and either BLDL or BLDLF are included in it. The installation may also specify that the system parameters CSA, REAL, SQA, VRREGN, APF=00, and FIX=01 should be included in the IEASYS00 list during SYSGEN processing.

The remainder of the system parameters are specified in two ways:

- By using a system utility program to add them to the IEASYS00 or any IEASYSxx list.
- By entering them through the operator's console.

The IEASYS00 list is read for every IPL. The installation is not only able to add parameters to this list, but may change parameter values that were included in it during SYSGEN processing. Operator interaction will be minimized during system initialization processing if the installation ensures that the IEASYS00 list contains all of the system parameters required to establish the desired operating environment. The system operator will then be able to respond to the "SPECIFY SYSTEM PARAMETERS" message with end-of-block (EOB), and will not be required to respond further unless error conditions are detected and prompting occurs.

The IEASYS00 member should generally contain installation default values, or parameters for which values can be defined that do not change from IPL to IPL. The IEASYSxx members, on the other hand, should be used to contain parameters that are subject to change from IPL to IPL, depending on the user applications. (See the SYSP parameter description for additional information.)

### ***System Parameter Library (SYS1.PARMLIB)***

The SYS1.PARMLIB data set contains both IBM-created and installation-created lists of parameter values. The formats for the lists have been revised from MVT to allow better space utilization within parameter records, and to afford more flexibility in their definition. All MVT and VS2 Release 1 parameter lists that are still valid lists in MVS need not have their formats changed.

## Changes to SYS1.PARMLIB Lists

Figure 8 shows the new or changed MVS SYS1.PARMLIB lists.

List	Change from MVT	Change from VS2 Release 1
COMMNDxx	New	New
IEAABD00	New	New
IEAAPFxx	New	New
IEAAPP00	New	New
IEABLDxx	New	None
IEADMP00	New	New
IEAFIXxx	New	None
IEAIGE00	No longer supported	No longer supported
IEAIGG00	No longer supported	No longer supported
IEAIPSxx	New	New
IEALOD00	New	None
IEALPAxx	New	None
IEAOPTxx	New	New
IEAPAK00	New	None
IEARSV00	No longer supported	No longer supported
IEASYSxx	New	None
IKJPRM00	Now only TIOC parameters	Now only TIOC parameters
IQAORDER	Not applicable	No longer supported
IRBMF1xx	New	New
LNKLSTxx	No longer a single member	No longer a single member
PARMTZ	New	None
SMFPRMxx	No longer a single member named SMFDEFLT	No longer a single member named SMFDEFLT
VATLSTxx	No longer a single member named PRESRES	No longer a single member named PRESRES

Figure 8. MVS Changes to SYS1.PARMLIB Lists



## IBM-Created Lists

The IBM-created lists that are named in column 1 are standard in SYS1.PARMLIB for MVS.

Standard List	Alternate List	Contents
IEAABD00		Address of areas to be dumped by SYSABEND
IEABLD00	IEABLDxx	Names of SYS1.LINKLIB load modules whose directory entries are to be made in the BLDL table
IEADMP00		Addresses of areas to be dumped by SYSUDUMP
IEAIPS00	IEAIPSxx	Installation performance specification
IEALOD00		Names of SYS1.LPALIB modules whose directory control block information is to be maintained in nonpageable storage
IEAPAK00		Groups of names of SYS1.LPALIB modules that are to be packed together in the link pack area
IEASYS00	IEASYSxx	Lists of system parameters
LNKLST00	LNKLSTxx	Names of additional data sets that may be concatenated with SYS1.LINKLIB data sets
SMFPRM00	SMFPRMxx	SMF parameters

The following lists are optionally created during the system generation process if specified by the installation.

Standard List	Alternate List	Contents
IEAAPF00	IEAAPFxx	Names of authorized libraries
IEAAPP00		Names of installation-written EXCP appendages
IEAFIX01	IEAFIXxx	Names of SYS1.SVCLIB, SYS1.LPALIB, and SYS1.LINKLIB modules that are to be made resident as a nonpageable extension to the link pack area

Of the IBM-created lists, multiple lists can be created and maintained for those that have alternate names. These lists may therefore be respecified by the installation to continue the system tailoring begun with system generation processing. The desired list may then be selected from the multiple lists by system parameters specified from the operator's console or in the IEASYS00 or IEASYSxx lists.

### Authorized Appendage List (IEAAPP00)

This list contains the names of installation-written I/O appendage routines and a description of the function of each routine. Any I/O appendage that is to be used by an unauthorized routine must be named in the IEAAPP00 member of SYS1.PARMLIB. (Authorized routines are able to use any installation-written I/O appendage.) This system integrity requirement prevents unauthorized routines from incorrectly using I/O appendages. To restrict an I/O appendage to use by authorized programs, omit it from this list.

It is not necessary to include the names of control program (e.g., system access methods) appendages in an authorized appendage list.

If the installation chooses to specify the creation of IEAAPP00 during system generation, appendage names may be entered, even if the appendages have not yet been written. This capability allows the installation to "prime" the list with the names of appendages that will be added to the system later.

See the chapter "System Integrity" for a description of authorized and unauthorized routines.

#### **Authorized Program Library List (IEAAPFxx)**

This list contains the data set names and corresponding volume identifications for libraries that may contain programs requiring APF authorization.

The ability for installation-specified libraries to contain APF-authorized programs extends the VS2 Release 1 APF support that allowed authorized programs to reside only in SYS1.LINKLIB, SYS1.SVCLIB, and SYS1.LPALIB. The specification of additional libraries for authorized programs in the IEAAPF00 or IEAAPFxx list in MVS eliminates the need to concatenate libraries containing programs requiring APF-authorization.

It is not necessary to include SYS1.LINKLIB or SYS1.SVCLIB in an authorized program library list.

#### **Dump Default Lists (IEAABD00 and IEADMP00)**

The dump default lists, IEAABD00 and IEADMP00, contain the IBM-supplied defaults for the storage areas to be dumped to the data sets defined by SYSABEND and SYSUDUMP DD statements, respectively. The installation may change the defaults in these lists to satisfy individual requirements. Some of the defaults that may be specified include the system queue area, the local system queue area, the scheduler work area for the task, the task's control blocks, the GTF or supervisor trace table, the PSW at entry to ABEND, general register contents at entry to ABEND, the contents of the task's link pack area, the contents of the task's job pack area.

#### **Fixed LPA List (IEAFIXxx)**

The fixed LPA list indicates those modules from SYS1.SVCLIB, SYS1.LPALIB, and/or SYS1.LINKLIB that should be included in the nonpageable extension to the link pack area. The list should contain modules that significantly increase system performance when they are fixed rather than paged.

The fixed LPA list is created during system generation processing, if so specified by the installation.

#### **Installation Performance Specification (IEAIPSxx)**

The installation performance specification (IPS) consists of control information used by the workload management routines that are part of the system resources management routines. It includes performance group definitions, performance objectives, and service definition coefficients that are used to establish service rates.

For a more detailed description of the contents of the IPS, see the chapter: "Directing the Use of System Resources".

### **Link Library List (LNKLSTxx)**

The link library list enables concatenating up to 15 data sets, on multiple volumes, with SYS1.LINKLIB. It is generated during the system generation process but may be modified via the IEBUPDTE utility program. Each data set that is concatenated with SYS1.LINKLIB may have up to 16 extents. (Note: After additions or changes are made that extend data sets concatenated with SYS1.LINKLIB, the system should be re-IPLed to update the description of SYS1.LINKLIB to the system.)

In MVS, multiple link library lists may be maintained in SYS1.PARMLIB. The list to be used is selected at IPL time by the LNK system parameter.

### **LPA Directory Load List (IEALOD00)**

The link pack area (LPA) directory load list indicates those modules residing on SYS1.LPALIB or in the pageable LPA that are to have the contents supervision directory control block information initialized by NIP in nonpageable storage. When this list is used, contents supervision routines do not have to search to find the module in the LPA directory. The number of page faults that may occur is therefore reduced. Candidates for this list are modules that are frequently used.

### **LPA Packing List (IEAPAK00)**

The link pack area (LPA) packing list contains the names of modules residing on the SYS1.LPALIB data set that may be grouped together by packing the LPA. Grouping these modules may sharply reduce page faults as well as eliminate wasted space within the LPA.

Modules that frequently pass control to one another should be grouped together to reduce page faults.

The total size of all modules in a group should not exceed 4K (page size).

The proper loading of modules into the LPA is done at NIP time, based upon the data in IEAPAK00. Modules that are not listed in IEAPAK00 are then loaded into the LPA in the order that they appear in the SYS1.LPALIB directory.

There are no alternate lists for IEAPAK00. If IEAPAK00 is not found as a member of SYS1.PARMLIB, the modules will be loaded in the order that they appear in the SYS1.LPALIB directory.

### **Resident BLDL List (IEABLDxx)**

The resident BLDL list specifies the names of modules from SYS1.LINKLIB, or any data set concatenated to SYS1.LINKLIB, that are to have directory entries built by NIP. This eliminates the directory search required when a load module is requested from SYS1.LINKLIB.

The directory will reside either in the fixed or pageable area of the system, depending upon whether the BLDLF or BLDL reply is specified by the operator in response to the "SPECIFY SYSTEM PARAMETERS" message. The BLDL and BLDLF parameters are mutually exclusive.

The member name for the standard list is IEABLD00. The load module names must be listed in the same order as they appear in the directory; that is, they must appear in ascending collating sequence. (Note: Directory entries in the resident BLDL table are not updated as a result of updating the load module in

the library. The old version of the load module is used until an IPL process updates the resident BLDL table.)

The IEABLD00 list indicated by BLDL=00 or BLDLF=00 is created during system generation processing. The list will be null if no entries are specified for it at SYSGEN.

If specified during system generation and not modified through operator communication or a list of system parameters (IEASYSxx) the default IEABLD00 is used.

#### **SMF Parameter List (SMFPRMxx)**

The SMF parameter list contains the parameters that control SMF operations. Multiple SMF parameter lists may be maintained in SYS1.PARMLIB. The list to be used is selected at IPL time by the SMF system parameter.

The single SYS1.PARMLIB member (SMFDEFLT) supplied by IBM in MVT and VS2 Release 1 is not valid. In MVS, the IBM-supplied SYS1.PARMLIB member that contains the default SMF parameters is called SMFPRM00. The format of the SMFPRM00 and SMFPRMxx lists is different from the MVT and VS2 Release 1 SMFDEFLT list.

The ALT and PRM keywords are not supported because the SMF data sets, SYS1.MANX and SYS1.MANY must be cataloged in MVS.

The MDL keyword has also been deleted.

The .SID keyword has been expanded to allow the specification of a System/370 model number.

#### **System Parameter List (IEASYSxx)**

The system parameter list may include all system parameters that are valid input to the "SPECIFY SYSTEM PARAMETERS" message. The standard list, IEASYS00, is created during Stage II of the system generation process. This parameter list enables an installation to specify all parameters needed during a particular IPL process without requiring that they be specified from an operator's console.

Operator intervention for an individual parameter list may be restricted by use of the 'OPI=YES/NO' parameter.

## Installation-Created Lists

The installation is also able to create certain lists, that are not supplied by IBM, to further tailor the system to installation needs. Most of these may also be maintained as multiple lists. The following lists may be created by the installation in MVS:

Default List	Alternate List	Contents
COMMND00	COMMNDxx	Commands to be issued by the system control program at the completion of system initialization processing.
IEALPA00	IEALPAxx	Names of SYS1.SVCLIB, SYS1.LPALIB, and SYS1.LINKLIB modules that are to be made resident as a pageable extension to the link pack area.
IEAOPT00	IEAOPTxx	System resource management tuning parameters.
IKJPRM00		TIOC buffer parameters
IRBMF100		MF/1 parameters
PARMTZ		Local time constant
VATLST00	VATLSTxx	Volume attributes

### Command List (COMMND00)

This list contains commands (e.g., START) that are to be issued by the system control program at the completion of system initialization processing. It may also be used to specify whether or not the time-of-day (TOD) clock initialization messages and associated responses should be issued.

The COMMNDxx list is specified by the CMD system parameter.

### Local Time Constant (PARMTZ)

This SYS1.PARMLIB member may be used to contain the value (in hours, minutes, and seconds) that the local time differs from Greenwich Mean Time (GMT), and the direction (east or west) from the GMT.

This constant is used to set the time-of-day clock and overrides the local time value that is specified at system generation for inclusion in the CVT.

### Modified Link Pack Area List (IEALPAxx)

This list may be used to indicate reenterable routines from the SYS1.LINKLIB, SYS1.LPALIB, and SYS1.SVCLIB data sets that are to be contained in the pageable extension to the link pack area. The routines in this area (called the modified link pack area) are resident only for the current IPL. If a routine is searched for, the modified link pack area is searched first, followed by the pageable link pack area. If a system restart is initiated and a previously created link pack area is used, the routines in the modified link pack area must be respecified if desired.

The IEALPAxx list is specified by the MLPA system parameter.

### **System Resources Management Tuning Parameter List (IEAOPTxx)**

This parameter list contains the values that are used by the system resources management algorithms to influence decisions about swapping jobs in or out of real storage. The values are described in the chapter: "Directing the Use of System Resources".

### **MF/1 Parameter List (IRBMF1xx)**

This list contains the parameters that control MF/1 operation. The options that may be specified are described in the topic: "MF/1 Operation" in the chapter: "Directing the Use of System Resources".

### **Volume Attribute List (VATLSTxx)**

The volume attribute list may be used to define the mount characteristics (permanently resident, reserved) and allocation characteristics (storage, public, private) for any, or all, direct access volumes used by an installation.

This list was a single SYS1.PARMLIB member called PRESRES in MVT and VS2 Release 1. Multiple members of the volume attribute list may be maintained in MVS.

The VATLSTxx list is specified by the VAL system parameter.

## System Libraries and Data Sets

Some system data sets are basic to MVS and are required for every operating system. Other system data sets are optional and are required only when selected options are included in the system.

### *Required Libraries and Data Sets*

The following libraries and data sets are required for every MVS operating system and must be on the system residence volume.

- **SYS1.LOGREC** (error recording data set) -- The machine error recording data set is used by recovery management routines to record statistical data about machine errors and software errors.
- **SYS1.NUCLEUS** (nucleus library) -- The nucleus library contains the resident portion (nucleus) of the control program and may contain modules selected and link-edited during system generation.
- **SYS1.SVCLIB** (SVC library) -- The SVC library contains some OLTEP modules and some appendage modules.

The following libraries and data sets are required for every MVS operating system and must be on direct access volumes. It is not necessary that they be on the system residence volume.

- **Job Entry Subsystem Data Set(s)** -- These data sets contain spooled and checkpointed record used by either JES2 or JES3. Either may be a multivolume data set.
- **Master Catalog** -- The master catalog is a key-sequenced VSAM data set that contains pointers to cataloged data sets and user catalogs. The name can be specified at system generation time on the PDS= parameter of the Stage I DATASET macro instruction. This catalog replaces the MVT and VS2 Release 1 system catalog (SYSCTLG).
- **Page Data Sets** -- The page data sets are known collectively as paging space. Paging space contains the “pagedout” portions of address spaces and the common system area, as well as the data written to virtual I/O data sets.
- **SYS1.DSSVM** -- This data set contains the command language modules used by the dynamic support system (DSS).
- **SYS1.LINKLIB** (link library) -- The link library contains programs and routines that are not in the link pack area and are referred to by ATTACH, LINK, LOAD, or XCTL macro instructions.
- **SYS1.LPALIB** (link pack area library) -- The link pack area library contains all modules that are loaded into the pageable link pack area. In MVS, the pageable link pack area is a read-only area. Therefore, modules contained in it must be refreshable. Modules that are modified during execution (non-refreshable) should be included in the modified link pack area. Because all of the modules in this library are brought into the LPA at IPL time, this library may be placed on a demountable volume that may be removed after IPL.
- **SYS1.MACLIB** (macro library) -- This library contains macro definitions for the system macro instructions used by the assembler-language processor.

- **SYS1.PARMLIB** (parameter library) -- The parameter library contains default parameters for the initialization of system functions.
- **SYS1.PROCLIB** (procedure library) -- The procedure library contains cataloged procedures used for job definition.
- **SYS1.SAMPLIB** -- This library contains independent utility programs, the IPL text, and the SMF exit routines.
- **SYS1.STGINDEX** -- This VSAM data set is used as an internal catalog for the pageable link pack area and virtual I/O data sets that must be saved across job steps and IPLs.
- **SYS1.BROADCAST** (broadcast data set) -- the broadcast data set contains two types of time-sharing messages: notices and mail. This data set must be cataloged but requires space allocation only if time sharing messages are to be written.
- **SYS1.CMDLIB** (command library) -- This library contains time-sharing command processor programs, service routines, and utility programs. This library must be cataloged. It requires space allocation unless the TSO stage 1 system generation macro instruction is coded such that the time sharing command system is not included.
- **SYS1.MANX/SYS1.MANY** (SMF data sets) -- The system management facility (SMF) data sets contain the information collected by the SMF routines. They may also contain measurement information collected by the system activity measurement facility (MF/1). The SMF data sets must be cataloged but require space allocation only if SMF or MF/1 recording is planned.
- **SYS1.UADS** (user attribute data set) -- This data set contains a list of terminal users who are authorized to use time-sharing and information about the users. This data set must be cataloged but requires space allocation only if the installation plans to initiate terminal sessions.

### ***Optional Libraries and Data Sets***

The following data set is optional and, if selected, must be on the system residence volume.

- **PASSWORD** -- This data set contains records that associate the names of protected non-VSAM data sets with the passwords assigned to the data sets.

The following libraries and data sets are optional and, if selected, must be on direct access volumes unless otherwise stated.

- **SYS1.DCMLIB** -- This library contains program function key (PFK) definitions for display consoles and is required if a display console with the optional PFK feature is included in the system.
- **SYS1.DUMP** (dump data set) -- This data set contains core image dumps recorded in the event of abnormal termination of a critical task. The dump data set may reside on tape.
- **SYS1.HELP** -- This data set is required if the time-sharing HELP command is used. Each member of this data set contains information regarding the syntax, subcommands, and functions for each time-sharing command.



- **SYS1.IMAGELIB** -- This data set contains the 1403 and 3211 universal character set (UCS) and forms control buffers (FCB) image modules. It is required if a 1403 printer with the UCS feature, 3211 printer, 3890 document processor device, or 3886 optical character recognition device is included in the system.
- **SYS1.PAGEDUMP** -- This data set contains the stand-alone dump program, if this program is on a direct access device. If the stand-alone dump program is on tape, this data set will not exist.
- **SYS1.TELCMLIB** (telecommunications library) -- This library contains the telecommunications subroutines in load module format and is required if any of the TP access methods other than the virtual telecommunications access method (VTAM) are selected.
- **SYS1.VTAMLST** -- This data set contains lists of internal parameters in card image format used during VTAM initialization.
- **SYS1.VTAMLIB** -- This library contains the VTAM executable code in load module form.
- **SYS1.VTAMOBJ** -- This data set contains the 3705 network control program (NCP) that is loaded by VTAM during VTAM initialization.

### ***Changes to VS2 Release 1 System Libraries and Data Sets***

System data sets and libraries in MVS are the same as their counterparts in VS2 Release 1 with the following exceptions:

- The page data set, **SYS1.PAGE**, has been replaced by two or more uniquely named VSAM data sets. This paging space must be defined before the first IPL. (See the topic "Adding Paging Space to the System.")
- The system catalog, **SYSCTLG**, has been replaced by a key-sequenced VSAM data set. Conversion aids are discussed in the topic "Catalog Conversion" in the chapter "Conversion Considerations".
- The data sets that support job entry subsystems are new.
- The format of the **SYS1.BROADCAST** data set has been modified to include a change level indicator in the header record and a new field for each user identification in the mail directory records. This field contains the address of the last message in the chain of messages associated with each user identification. An MVS format for the broadcast data set is obtained by using a new subcommand (SYNC) of the time-sharing **ACCOUNT** command. (See the topic "Formatting the Broadcast Data Set" in the "Conversion Considerations" chapter.)
- The load module size of modules in the **SYS1.DSSVM** data set has been increased from 1K bytes to 2K bytes.
- The **SYS1.SYSJOBQE** data set does not exist. It has been replaced by a scheduler work area (SWA) in virtual storage for initiator records and by a job entry subsystem data set for input, queued jobs, and output.
- The content of the **SYS1.LOGREC** data set has been expanded to include records for software error recording, missing interrupts, and dynamic device reconfiguration(DDR) routines.

- The SMF data sets, SYS1.MANX and SYS1.MANY, may now be used for system activity measurement facility (MF/1) recording. They must be cataloged in MVS.
- The log data sets are no longer named SYS1.SYSVLOGX and SYS1.SYSVLOGY by IBM. They are dynamically allocated by the job entry subsystem.
- The SYS1.PARMLIB library contains new members to control the initialization of the system. See the topic "SYS1.PARMLIB" in the section "Defining the System" for descriptions of these members.
- The SYS1.STGINDEX data set is new.
- The format of the SYS1.UADS data set has been modified to contain information for new keywords on ACCOUNT subcommands. An existing user attributes data set (UADS) is converted to the new MVS format by using an Account Facility utility program (UADSREFM). (See the topic "Reformatting the UADS" in the "Conversion Considerations" chapter.)
- The SYS1.VTAMLIB, SYS1.VTAMLST and SYS1.VTAMOBJ data sets are new.

### ***Changes to MVT System Libraries and Data Sets***

System data sets and libraries in MVS are also the same as their counterparts in MVT, with the exceptions previously noted and the following:

- The SYS1.ACCT data set no longer exists. The SMF data sets SYS1.MANX and SYS1.MANY should be used in its place.
- The SYS1.DSSVM data set is new.
- The SYS1.LPALIB data set is new. It contains all of the SVCs and most of the modules that, in MVT were contained in the SYS1.LINKLIB and SYS1.SVCLIB libraries. All of the modules contained in this new data set appear in the fixed or pageable link pack area in VS2.
- The page data sets are new.
- The SMF data sets, SYS1.MANX and SYS1.MANY, may now be used for VSAM recording.

### ***Adding Paging Space to the System***

Before the first system IPL, the installation must define and preformat paging space. In MVS, paging space contains system pages, the pageable link pack area, and virtual I/O data sets. Paging space consists of two or more VSAM data sets. The use of the define statement with this type of VSAM data set causes a paging data set to be formatted with track overflow for maximum performance of paging I/O.

At least two such data sets are required for paging space because two copies (primary and secondary) of certain pages are maintained on external storage. These data sets must contain enough space for a minimum number of secondary pages, corresponding primary pages, and additional pages that have no associated secondary copies.

The auxiliary storage management initialization routines enforce these minimum values by internally separating paging data sets for exclusive storage of

secondary page copies, from paging data sets used exclusively for primary copies. These routines attempt to select for secondary page copies, enough (entire) data sets on the slowest paging device type to satisfy the minimum number of pages. They classify paging devices by average access time. For best separation, the installation should direct candidate page data sets for secondary page storage onto the slowest device available for paging; preferably directing the other paging data sets to separate devices.

The installation can include more paging space than the minimum by specifying additional page data sets during system generation. If it is necessary to add paging space to the system after system generation, the installation must first use access method services to define and format the data sets, then cause their names to be entered in a PARMLIB member PAGE list, and finally re-IPL the system. The operator can also specify additional names via the PAGE system parameter.

No more than 25 page data sets can be specified during system generation. The number of page data sets must never exceed 63.

Page data sets can be added to the system in even or odd numbers; there is no need to pair them.

### ***Creating Temporary Data Sets in External Page Storage Using Virtual I/O***

Conventional access methods (BSAM, QSAM, BDAM, and BPAM) and XDAP and EXCP macro instructions, can create and access temporary data sets in external page storage or system paging space by utilizing the virtual input/output (VIO) facility.

Specific unitnames can be defined during system generation (UNITNAME macro) as eligible for VIO processing. The VIO unitname defined is coded in a problem program JCL statement which defines a system-named temporary data set without specifying the volume serial number. VIO then interprets and simulates the execution of the channel programs associated with an EXCP issued by an access method or a problem program.

The channel programs are not scheduled and translated in the conventional manner by the I/O supervisor. VIO simulates I/O activity using a buffer that is the size of a track for the first device in the list defined by the UNITNAME macro instruction. To the access method or problem program, a VIO device is a real direct access storage device; actually the virtual device is simulated on the volumes assigned to external page storage.

VIO processing relies heavily on the services of the paging supervisor. The paging supervisor causes the real I/O that is requested by VIO, and maintains for virtual, auxiliary, and real storage, the page tables that make data accessible in a virtual storage environment. The paging supervisor maintains the contents of the VIO buffer based on requests issued by VIO.

VIO offers the following enhancements to conventional processing of temporary data sets:

- Channel program translation and page fixing by the I/O supervisor for each EXCP request, are eliminated.
- The paging supervisor dynamically allocates page-size physical blocks (auxiliary storage page slots) to a VIO data set as it is being created. DADSM overhead for the allocation of space on a real device is thereby eliminated.

- Many I/O operations may be satisfied by moving data between a VIO buffer and an access method or user buffer, thus eliminating the channel and device activity associated with normal channel program request processing. The data set is automatically blocked in page size (4K) records, irrespective of access method or user-buffer size.
- There is generally more efficient utilization and easier management of direct access space, since temporary data sets use auxiliary space within the paging data sets.
- When a data set page is to be read, the paging supervisor determines if it is still in real storage and available for reclamation (the reclaim function). If it can be reclaimed, the read is accomplished without physical I/O.

## **Directing the Use of System Resources**

This chapter describes some of the MVS facilities that allow installations to direct the use of system resources.

System resources management routines allow an installation to specify, in measureable terms, the relative rate at which system resources should be provided to various categories of background and terminal jobs.

The system activity measurement facility (MF/1) provides reports on selected areas of system activity. The installation uses the information in these reports to determine how to specify performance objectives for the system resources management routines.

## System Resources Management

System resources management routines are new in MVS. They provide:

- The capability for an installation to specify, in measurable terms, the rate at which system resources should be provided to various users of the system.
- A means by which the system control program is able to monitor the use of system resources and attempt to maintain their most efficient use at all system workload levels.

These routines monitor the use of system resources to determine when background and time sharing jobs should be swapped in or out to meet the installation-specified objectives.

The resources management routines are divided into three primary categories:

- Workload management routines.
- Resource-use routines.
- A control function.

### *Workload Management Routines*

The function of the workload management routines is to control the relative rate at which processing resources are provided to individual jobs or time sharing users so as to conform to installation performance specifications. Their design enables installations to specify different processing rates for a job or time sharing command depending on the system workload level and on the age of, or amount of processing done by, a job or command. Workload management routines accomplish this processing rate control by monitoring the processing resources used by each address space and scheduling swaps (to or from main storage) to achieve an intended processing rate. A more detailed discussion of the installation interface with the workload management routines is found under "Workload Management Concept."

### *Resource-Use Routines*

The resource-use routines consist of a set of algorithms that make scheduling recommendations intended to optimize the utilization of the system CPU, I/O, main storage, and paging resources. These algorithms include:

#### **I/O Load Balancing**

This algorithm attempts to maintain a dispatchable job mix that has a balanced use of logical channels. The algorithm monitors the fraction of I/O requests delayed on each logical channel and ascertains whether this measure of load falls within an acceptable range. When an imbalance -- either an overload or underload situation -- occurs the algorithm seeks out an address space known to be a heavy user of the imbalanced resource, and may then recommend a swap. To reduce the overhead of monitoring the I/O profile of each individual address space, the I/O request counts gathered already on behalf of SMF are analyzed for this purpose.

#### **CPU Load Adjustment**

This algorithm continuously monitors CPU utilization and when this value either goes above or below an acceptable target range, the CPU Load Adjustment algorithm will attempt to remedy the situation by recommending the swapping of one or more heavy CPU users into main storage in the case

of CPU underload, or out of main storage in the case of overload. Because MVS installations are advised to initiate more jobs than will normally fit in main storage, ingredients for a well balanced job mix normally exist on the queue of swapped-out but ready to execute address spaces.

A special case occurs when the CPU is 100% utilized. When this occurs the algorithm looks for address spaces near the bottom of the dispatching queue that are not being regularly dispatched. This situation is considered a serious CPU overload because it means that address spaces are tying up real storage, but are either not executing at all or are executing too slowly to warrant the expenditure of real storage. The solution is to recommend the replacement of heavy CPU users in main storage with I/O users who make better use of the system resources.

#### Main Storage Occupancy

This algorithm attempts to ensure that the sum of the working sets of all swapped-in address spaces (those address space pages that are regularly referenced) nearly fills, but does not exceed, the capacity of main storage. The sum is judged too large if the size of the available frame count drops below a low threshold. When this occurs, an address space is promptly swapped-out to free frames and thereby increase the available frame count. To prevent this corrective action from happening too often, the main storage occupancy algorithm evaluates the consequences of all anticipated swaps in an attempt to keep the available frame count near a target which is above the low threshold. If the low threshold is crossed too frequently, this target is dynamically adjusted away from the low threshold. If it is not crossed at all, the algorithm is too conservative and the target is dynamically adjusted closer to the low threshold.

#### Page Replacement

Page stealing makes frames occupied by unreferenced pages, available for reuse by active pages. The page replacement algorithm takes advantage of data maintained on an address space basis, to determine candidates for page stealing.

This represents a marked departure from Release 1 which employs a system-wide page stealing algorithm. Perhaps more important however, is the fact that in Release 2 page stealing is not used to perform the system-wide control function of ensuring the availability of sufficient deallocated real pages. The problem of thrashing -- excessive demand paging -- is largely eliminated in Release 2 because page stealing is not used to remedy real frame shortages. Instead, when such a shortage occurs, the SRM main storage occupancy algorithm uses swapping to reduce the number of address spaces executing in main storage.

To ensure an acceptable percentage of "good steals," the page replacement algorithm monitors the page referencing activity of individual address spaces, and selects a page to be stolen if and only if it has gone unreferenced for a predefined amount of that address space's CPU execution time. This criterion prevents the stealing of active working set pages that go unreferenced for periods of elapsed time simply because that address space temporarily did not get dispatched.

For pages in the Common Area this approach is not used, since these pages are not associated with any particular address space. Unreferenced elapsed time is used as the prime criterion for stealing these pages.

### Device Allocation

This algorithm determines the unit to be used for data sets for which an allocation choice must be made from a list of more than one device candidate. The goal is to allocate from lightly-utilized channels, and to collect allocations for the same address space on the same logical channel without collecting them on the same device, so that swapping that address space will have a predictable effect on the system I/O load.

### Automatic Priority Group (APG)

This algorithm reorders dispatchable address spaces within the automatic priority group based on the degree to which they are I/O bound; within the APG, I/O-bound jobs are given higher dispatching priorities than CPU-bound jobs.

To fully realize the benefits of this algorithm, the installation should place all possible users within the APG. In MVS, this is facilitated because the APG priority is used as the default step dispatching priority. (See the topic "Changes to VS2 Release 1 JCL" in the chapter "Conversion Considerations.")

### ENQ/DEQ Algorithm

This algorithm determines when a batch or terminal job is in control of a system resource and is delaying other jobs that are enqueued on the same resource. When such a determination is made, the job that is enqueued upon the system resource is made non-swappable for an installation-specified time interval. The interval is specified by the ERV parameter in the IEAOPTxx member of SYS1.PARMLIB.

## ***Control Function***

The control function gathers the results (recommendation values for potential swap decisions) produced by the workload management routines and the resource-use algorithms. It uses a control algorithm to combine and evaluate the individual recommendation values to produce a composite recommendation value for each potential swap decision. The control function then uses these values to determine the amount of swapping that should be performed in an attempt to satisfy the installation-specified performance objectives and efficiently use system resources.

The control algorithm uses optional installation-specified parameters called resource factor coefficients to weight the recommendation values produced by the I/O and CPU load balancing algorithms. The use of these coefficients (which are contained in the IEAOPTxx member of SYS1.PARMLIB) provides one method of system tuning. For example, if the installation increases the CPU coefficient, the recommendation value produced by the CPU load balancing algorithm will be more heavily weighted when it is evaluated by the control algorithm in conjunction with other recommendation values.

## ***Workload Management Concepts***

One of the SRM design objectives is to assure maximum control over user performance by an installation, without requiring the installation to supply its own scheduling algorithm. The solution adopted is to permit an installation to specify in measureable terms, the response and turnaround objectives wanted, and allow the workload management routines to determine how to achieve these objectives. The following concepts support such an approach by providing



measurable terms for the response objectives of individual batch and time sharing users.

### Transaction

The basic entity to which response and turnaround objectives are associated is called a transaction. Since both batch and time sharing are supported by the workload management routines, a concept of transaction is needed which supports both environments. A job is the batch entity about which turnaround is usually measured. It is therefore selected as the definition of a batch transaction. Provision also exists to treat individual job-steps as separate transactions.

The time sharing entity about which response is measured, normally begins when the user strikes end-of-block and ends when he receives his terminal output. This sequence may be encompassed in a time sharing command or subcommand -- in either case it qualifies as a separate transaction.

### Performance Variable

Having defined what a transaction is, the need now exists for a variable capable of expressing response and turnaround objectives for transactions. Response time itself is not satisfactory because it is too dependent on the individual processing requirements of each transaction. What is needed is a variable that measures the rate at which processing takes place, rather than the amount of processing required. An objective expressed in terms of processing rate can reasonably be used to assure a time sharing user a consistent quality of responsiveness, regardless of the variety of commands he chooses to enter.

### Service

The measure of processing chosen is a composite variable called service -- a combination of an individual address space's CPU, I/O, and main storage usage. The formula for service also includes three installation-supplied coefficients.

$$\text{SERVICE} = A * (\text{SMF Task CPU Time} * K) + \\ B * (\text{SMF I/O Request Count}) + \\ C * (\text{Working Set Size} * \text{SMF Task CPU Time} * K)$$

where: A, B, and C are installation-supplied constants, and K is a model-dependent MIPS adjusting factor.

### Service Rate

Service rate, defined as the service accumulated by a transaction divided by the elapsed time, is the variable used by the installation to express response and turnaround objectives. It is used by the workload management routines to monitor the progress of each executing transaction.

The SMF job termination record, the time sharing TIME command, and the MF/1 workload activity report, return to the installation the data necessary to correlate service rate with response and turnaround times.

Thus, for example, a user regularly submitting the same batch job should know that if it requires approximately 72K service units, and through arrangement with the installation manager the job is assured 25 su/sec, then it will complete in 48 minutes, while if the arrangement is for 50 su/sec it will complete in 24 minutes.

There is a practical limit to the service rate a job can absorb. The workload management routines can do no better than provide the job with the highest service rate it can absorb, even if a higher rate is requested.

## System Workload

One important new facility in MVS, is the ability to temporarily swap most batch and time sharing address spaces out of main storage. This facility makes it feasible -- and usually desirable -- to start more batch initiators than the real storage capacity can efficiently support. The SRM keeps track of all active address spaces, both batch and time sharing, and schedules swaps in and out so that a number of conditions are satisfied. For the purposes of this discussion, it is sufficient to mention two of these conditions:

- The demand for real storage should not cause excessive paging.
- The service rate afforded each address space should correspond closely to the installation performance objectives.

The combined demand of all address spaces for system resources constitutes the system workload. The more active address spaces that exist -- whether batch or time sharing -- the heavier will be the system workload. MVS is designed to efficiently handle a very wide range of system workloads. An installation should anticipate, however, that under heavy system workload conditions, some address spaces must be temporarily swapped out of main storage. This results in lower service rates for the delayed address space. Later it will be shown that an installation can specify the service rate reductions that individual batch and time sharing address spaces should experience under different system workload levels.

## Installation Performance Specification

It is the job of the installation to translate the various performance requirements of its users into one or more installation performance specifications. This control data, which is included in the SYS1.PARMLIB data set, permits the workload manager to schedule the availability of processing resources among individual users in accordance with the dictates of the installation.

An installation performance specification (IPS) is a mapping that associates a particular transaction with an intended service rate. It is important to note that the IPS mappings take into account other considerations, specifically the system workload, the amount of service already accumulated by a transaction, and the age of the transaction.

## Performance Groups

The purpose of performance groups is to group together user transactions that the installation considers to have similar performance requirements. A teleprocessing applications program that has multiple users and executes in a single virtual address space is considered to be a single transaction, and its terminal users are not distinguished from each other by the system resources management routines.

The installation can define as many as 255 performance groups, each identified by a distinct performance group number. Each performance group definition includes as many as eight performance group periods. By dividing a performance group into periods, an installation can associate different performance objectives with different periods in the life of a transaction. The duration of a period can be specified either as a number of real-time seconds or

as a number of accumulated service units. For example, one performance group might be defined for short compile-load-go jobs that should have a very rapid turnaround time. The installation divides the performance group into two periods. The first period is associated with a high service rate and lasts until N number of service units are accumulated, where N has previously been determined to be a sufficient number of service units to complete a short compile-load-go job. If a job in the group is not complete after N service units are accumulated, it enters the second period, for which a lower service rate has been specified, and remains in this period for the duration of the job.

## **Performance Objectives**

A performance objective specifies service rates; how many service units per second an associated transaction should receive under different system workload conditions.

The motive for specifying different performance objectives is to give certain users better service at the expense of other users. However, the relative importance of this motive is dependent on the size of the system workload (i.e., the degree to which the demand for system resources exceeds the supply).

The installation specifies the treatment of different subsets of jobs by fixing arbitrary points, called workload levels, at which it defines target service rates for each subset of jobs. Conceptually, increasing workload levels can be thought of as increasing competition or demand for system resources. Thus, in order to preserve a high service rate for high priority jobs as demand (workload level) increases, an installation will reduce the target service rate for lower priority jobs.

Typically, an installation supplies an IPS such that if the system workload is moderate, all jobs should receive satisfactory service rates. However, if the system workload is heavy, the installation will want to provide an acceptable service rate to high priority jobs, and lower the service rate for lower priority jobs. Under very heavy workload conditions the installation might completely sacrifice the service rate of lower priority jobs (i.e., assign a service rate of 0) in order for the system to continue to provide acceptable service to the higher priority jobs. By defining workload levels, the installation can express the varying service relationships between groups of users at different system workload levels.

For example, an installation defines three performance objectives, numbered 2, 5, and 8. When the system workload is light, each performance objective should specify a better than satisfactory service rate for the transactions associated with it; the installation assigns the number 2 to this workload level and assigns corresponding service rates to each performance objective, as illustrated in the chart at the top of Figure 9. Under very heavy workload conditions, the installation wants to completely sacrifice the service rates for objective 5 and to lower the service rate for objective 8 but still provide acceptable service to jobs associated with objective 2. This workload level is assigned the number 8; the corresponding service rates associated with each objective are illustrated in Figure 9. The two intermediate workload levels shown in Figure 9 are established to reflect the changing relationships between the performance objectives as the workload level increases from 2 to 8.

## Performance Objectives

Performance Objective Number	Service Rate for Workload Level 2	Service Rate for Workload Level 4	Service Rate for Workload Level 6	Service Rate for Workload Level 8
2	44	40	36	32
5	60	10	0	0
8	100	20	18	6

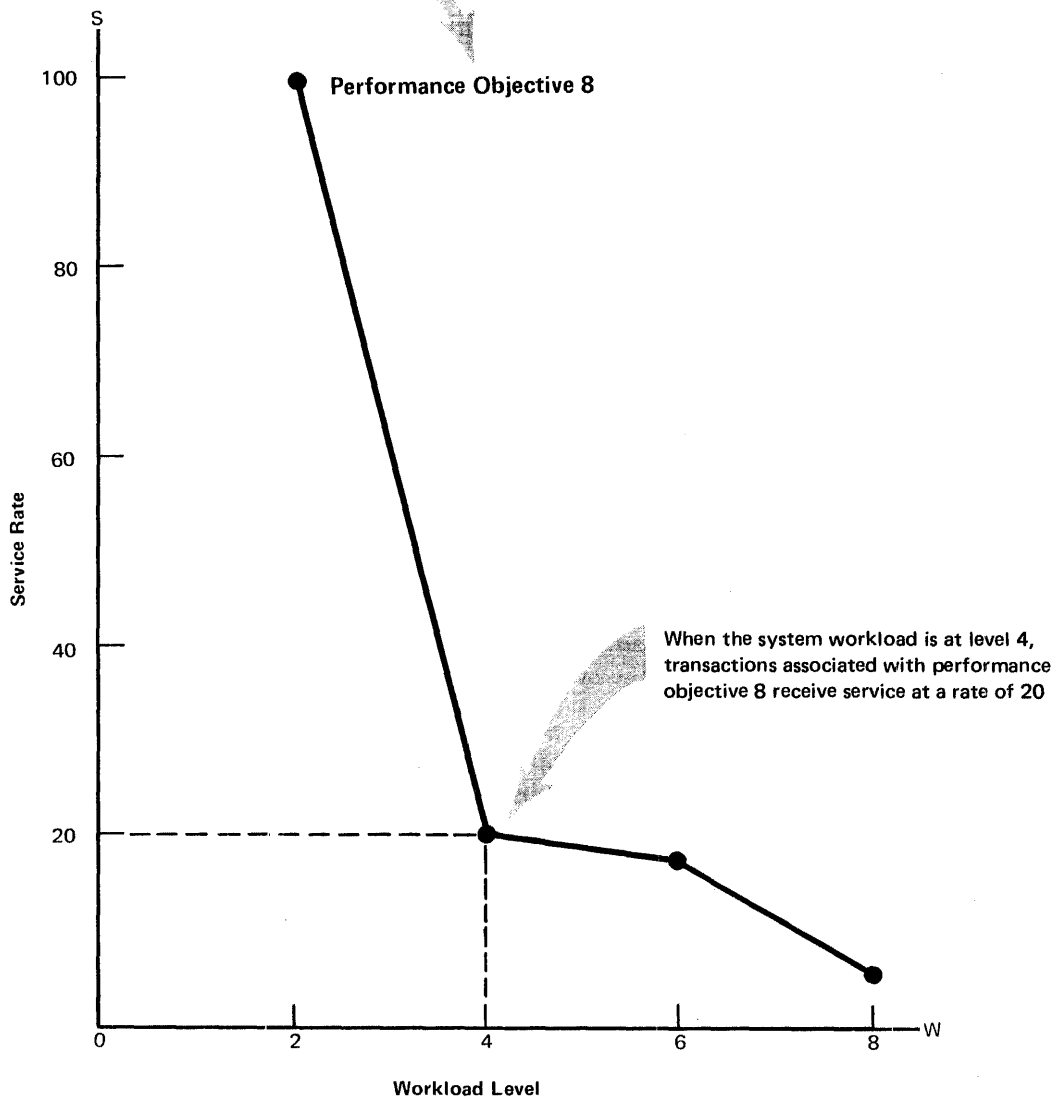


Figure 9. Performance Objective Definition

The system resources manager tracks the service rates provided to all users. As the system workload level increases or decreases, it adjusts the service rates in an attempt to maintain the relationships between the performance objectives (and therefore between the transactions associated with the objectives) that the installation has defined.

Figure 9 also contains a graphic representation of one of the performance objectives (performance objective 8). This graph illustrates the service rates that the installation has defined for the system workload levels of 2, 4, 6, and 8. The workload management routines attempt to provide the service rate that applies to the performance objective for the associated job at the current system workload level. In Figure 9, when the current system workload is 4, the workload management routines recommend swapping in order to maintain a service rate of 20 for the jobs that are associated with performance objective 8. Jobs associated with other performance objectives will be targeted at service rates indicated in their performance objectives for a workload level of 4.

### Associating a Transaction with a Performance Objective

Every transaction that enters the system has a performance group number. In Figure 10, transaction A is associated with performance group 9, transaction B is associated with performance group 8. For its first performance group period, transaction B will be associated with performance objective 5. This means that, for this period, the workload management routines will attempt to maintain a service rate for transaction B that equals the service rate specified in performance objective 5 for the workload level that equals the actual workload level of the system at a given time.

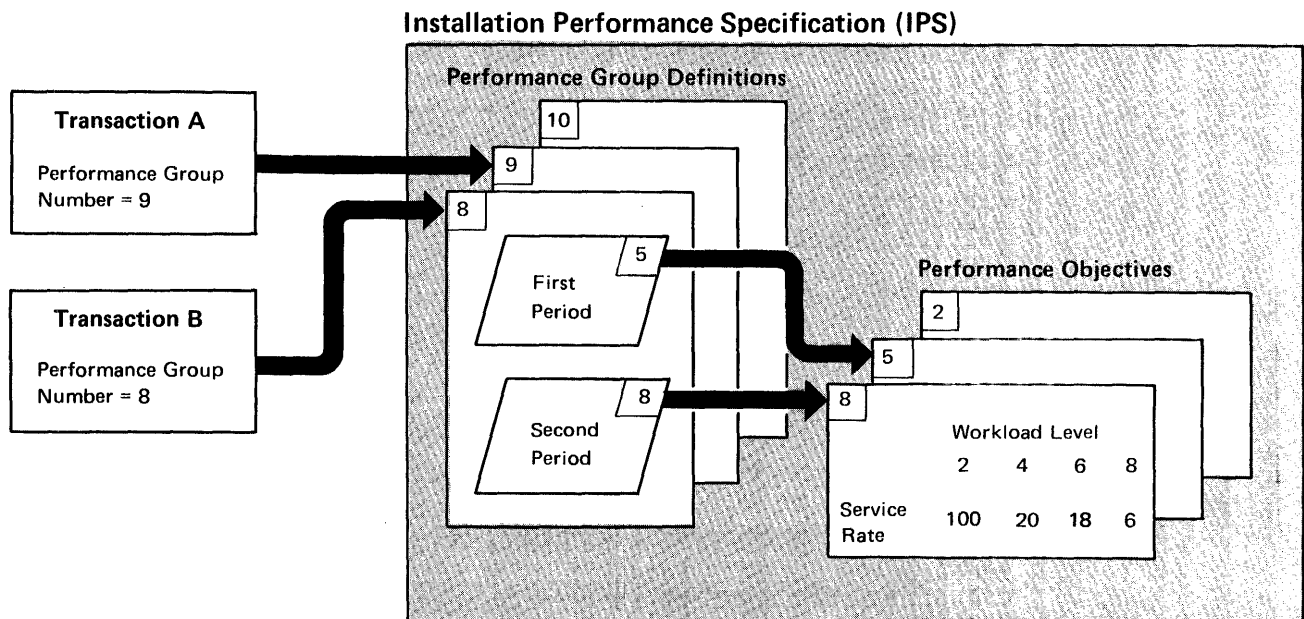


Figure 10. Associating a Transaction With a Performance Objective

### The Relationship Between Workload Levels and Service Rates

To summarize the relationship between workload levels and service rates, consider a hypothetical installation whose users fall into one of three distinct subsets. Each subset can be considered to be a performance group. They are:

- Performance Group A -- A long running batch job that is run only once a day is the only job in this class and is extremely important.
- Performance Group B -- Moderate length batch jobs, of intermediate importance make up this class.
- Performance Group C -- Relatively short batch jobs, least in importance, make up this class.

Figure 11 illustrates the performance objectives that have been defined for each of the three performance groups. (Three distinct performance groups, each with only one defined period and one associated performance objective, are assumed for simplicity of illustration.)

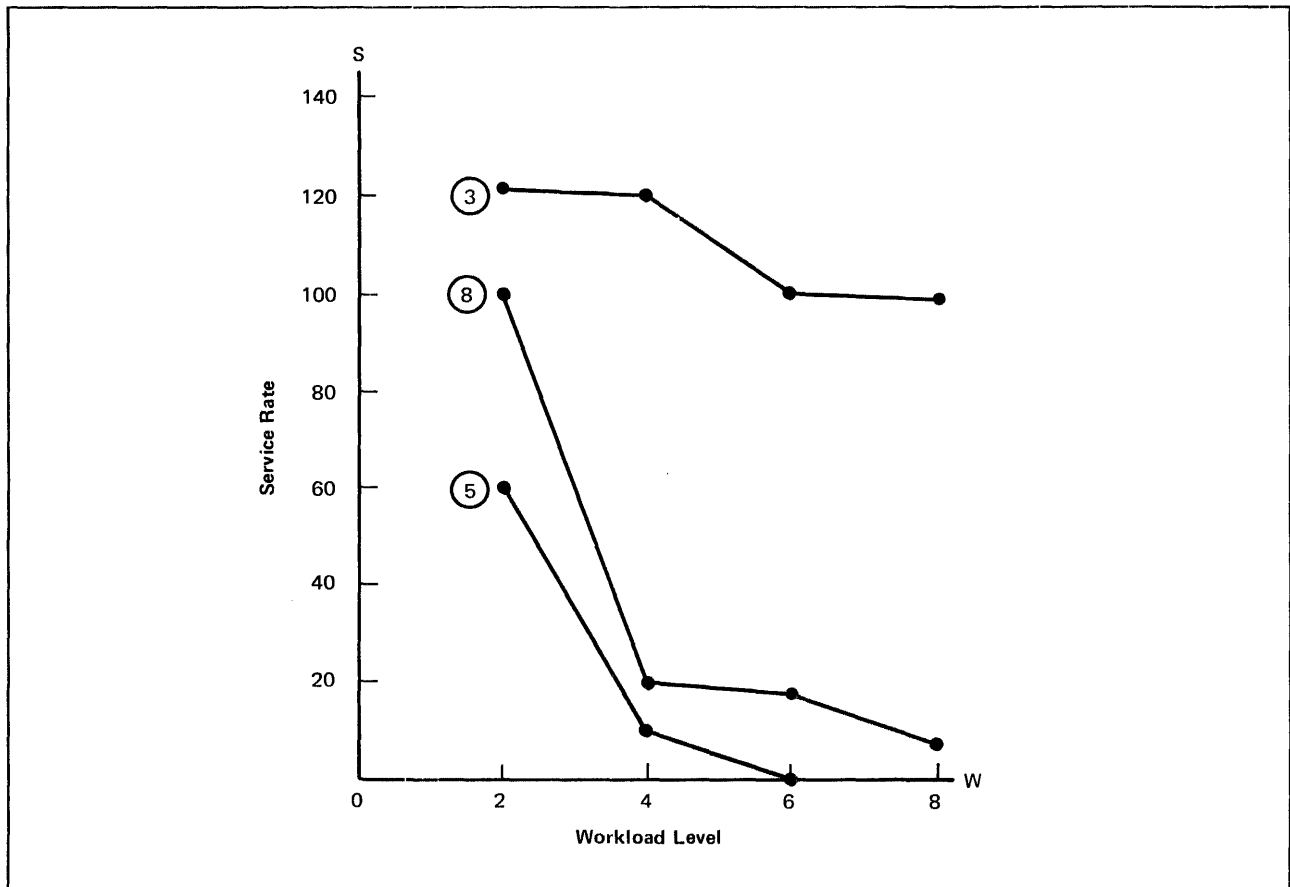


Figure 11. Performance Objectives

Group A jobs are associated with performance objective 3, group B jobs with performance objective 8, and group C jobs with performance objective 5. Jobs associated with performance objective 3 demand a service rate of approximately 120 at workload levels 2 and 4. At workload level 6, performance objective 3 jobs demand a service rate of 100 while performance objective 5 jobs have stopped receiving service, and performance objective 8 jobs have reduced their demands to 18 service units per unit time. At workload level 8, even performance objective 8 jobs nearly stop receiving service in deference to A jobs. As long as the A job receives a service rate of approximately 100, the installation knows that the job will complete processing in a satisfactory amount of time.

The workload level at which the system operates is a function of the number of jobs in each performance objective actually initiated at any time. Figure 12 illustrates system workload levels that result from three different possibilities:

- One job from each performance objective.
- One job from performance objective 3, four from performance objective 8, and two from performance objective 5.
- One job from performance objective 3, seven from performance objective 8, and one from performance objective 5.

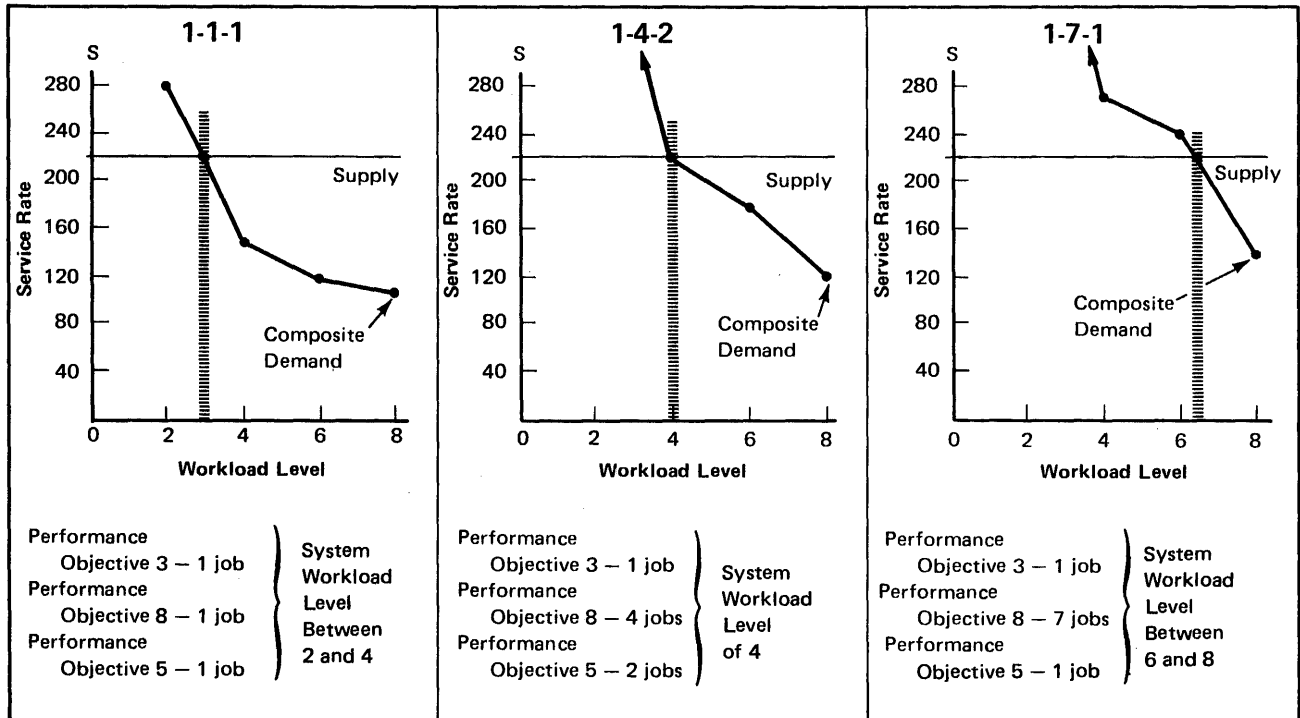
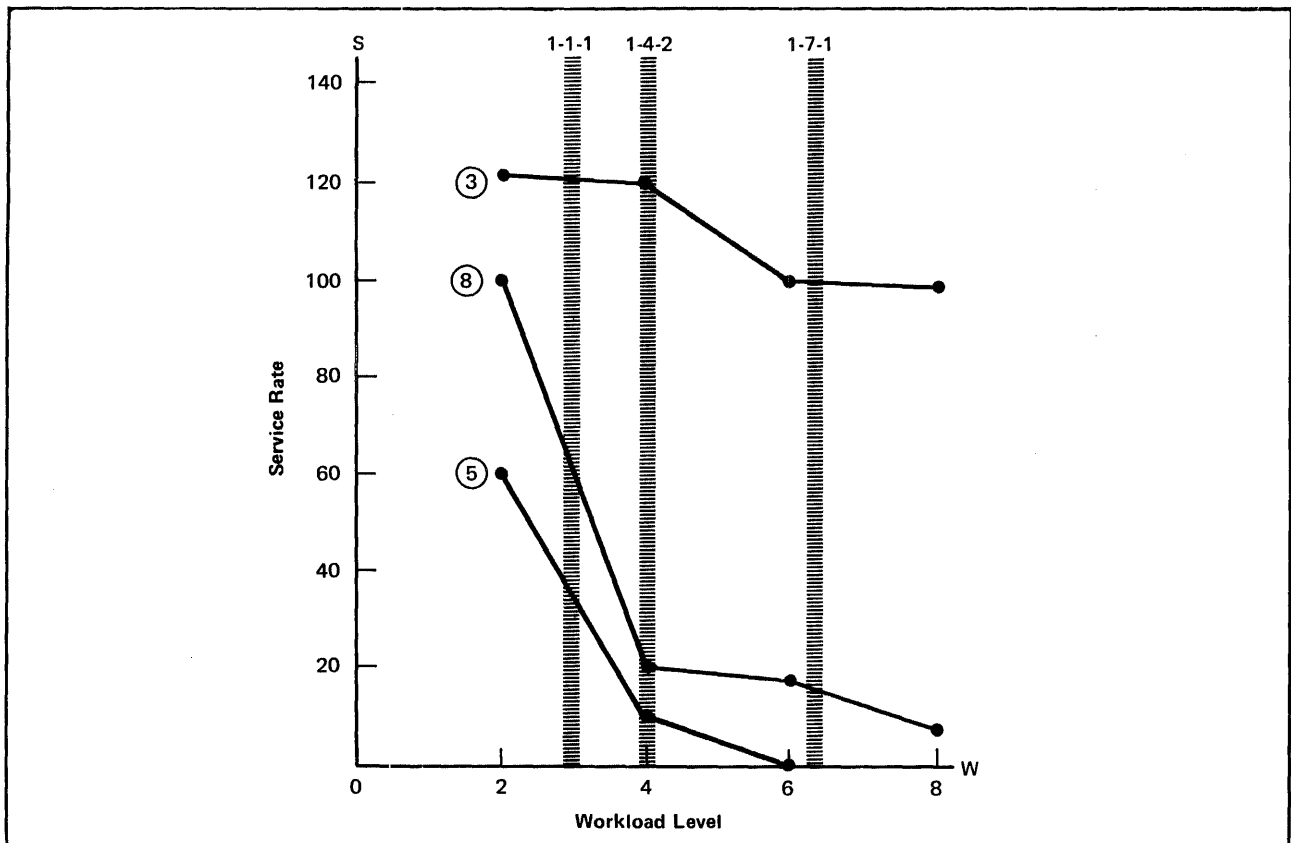


Figure 12. System Workload Levels

In these illustrations a simplified assumption is made that the system provides a composite service rate of 220 service units per unit of time to all active jobs; this is the "supply" line. The "demand" line represents the composite service demands from all jobs at each workload level. It is derived by multiplying the number of jobs in each performance group by the service rate specified for that performance objective at the particular workload level. In other words, with a job mix of 1-4-2, (the center graph on Figure 12), the composite service rate demanded by all jobs at workload level 4 is  $1 \times 120 + 4 \times 20 + 2 \times 10 = 220$ . (At workload level 4, a job associated with performance objective 3 demands a service rate of 120, a performance objective 8 job demands 20 and a performance objective 5 job demands 10.) The intersection of the supply and demand lines produces the resulting system workload - that point of equilibrium at which the system can meet the composite demands of all active users.

In Figure 13, the different system workload levels that result from the three different job mixes are superimposed over the three performance objectives to illustrate the service rates that apply as the system workload level increases.



**Figure 13. Service Rates for Different Performance Objectives at Various System Workload Levels**

As shown in Figure 13, the following conclusions about the installation's performance objective specifications may be drawn from this hypothetical analysis:

With 1 job initiated in each performance objective:

- The system will be operating at a workload level between 2 and 4.
- The performance objective 3 job will receive approximately 121 service units per unit time.
- The performance objective 8 job will receive approximately 64 service units per unit time.
- The performance objective 5 job will receive approximately 35 service units per unit time.

With 1 performance objective 3 job, 4 performance objective 8 jobs, 2 performance objective 5 jobs:

- The system will be operating at a workload level of 4.
- The performance objective 3 job will receive approximately 120 service units per unit time.
- The performance objective 8 jobs will each receive approximately 20 service units per unit time.
- The performance objective 5 jobs will each receive approximately 10 service units per unit time.



With 1 performance objective 3 job, 7 performance objective 8 jobs, 1 performance objective 5 job:

- The system will be operating at a workload level between 6 and 8.
- The performance objective 3 job will receive approximately 100 service units per unit time.
- The performance objective 17 jobs will each receive approximately 15 service units per unit time.
- The performance objective 5 job will not be receiving service.

In each of the preceding examples, the sum of the service rates provided to all users will equal approximately 220 service units per unit time (the composite system “supply”).

### ***System Resources Manager Parameters***

The system resources manager parameters fall into two classes. Those that affect the recommendation values produced by workload management routines are contained in the SYS1.PARMLIB member IEAIPS00 (or IEAIPSxx); those that otherwise affect swapping decisions are contained in the SYS1.PARMLIB member IEAOPT00 (or IEAOPTxx).

These parameters provide the installation with a great degree of flexibility in stating performance requirements.

#### **IPS Parameters (IEAIPSxx)**

An installation performance specification contains four categories of information:

- Service definition coefficients -- The workload management routines use these values as multipliers, in calculating the service that a transaction has received. (See the topic “Service” earlier in this chapter.)
- Up to 32 workload level numbers -- Workload level numbers are used to specify the relative service rates that jobs of different priorities are to receive at varying levels of system activity. Each number is related to a specific service rate in each performance objective, and the numbers increase in magnitude as contention for system resources increases.
- Up to 64 performance objectives -- Each performance objective is specified in terms of the service rates to be provided at different workload levels. The performance objective number that is specified for each performance group period, identifies the performance objective to be used for that period.
- Up to 255 performance group definitions -- A performance group definition is identified by a performance group number. It may be subdivided into as many as eight different periods. Each period has the following parameters associated with it:
  - period duration
  - units code
  - performance objective number
  - response/throughput bias
  - interval service value

These categories of information are specified by the following parameters:

#### Performance Group Number

The performance group number is an integer between 1 and 255 that associates a transaction with a performance group definition. All transactions have performance group numbers. 1 is the default for batch jobs and 2 is the default for terminal jobs.

#### Period Duration

The period duration is a positive integer that indicates the length of the interval during which a specified performance objective is to be associated with a transaction.

#### Units Code

The units code is specified by the letter "S" or the letter "R". "S" indicates that the indicated period duration is to be measured in terms of accumulated service; "R" indicates that it is to be measured in terms of real-time seconds.

#### Performance Objective Number

The performance objective number is an integer between 1 and 64, that indicates the performance objective with which the transaction is to be associated for the duration of the indicated period.

#### Response/Throughput Bias

The response/throughput bias (RTB) is an integer that indicates, for each performance group period, that deviations from the service rate specified in the performance objective can be tolerated in favor of higher system throughput. "1" indicates that for jobs in this performance group period, the resource-use algorithms will produce evaluations that will influence swap decisions; "0" indicates that they will not.

#### Interval Service Value

The interval service value (ISV) specifies a minimum number of service units that a transaction must receive during each interval of real storage occupancy, before a swap evaluation is made by the workload management routines.

#### Workload Level

The workload level is an integer between 1 and 128. Up to 32 workload level numbers can be specified (in increasing magnitude representing heavier workload levels of the system). Workload levels are used to differentiate between light and heavy demands for the use of system resources. The same workload level numbers must be specified for each performance objective in an IPS.

#### Performance Objective

Each performance objective is identified by an integer from 1 to 64 followed by a definition of the service rates that are associated with the workload level numbers specified by the workload level parameter. A service rate must be specified for each workload level number defined in the IPS.

#### Service Definition Coefficients

The service definition coefficients (CPU, IOC, MSO) are used by the system resources management routines to calculate the service units that are provided to a particular job. Each coefficient is a multiplier on one of the three basic processing units in these calculations. For example, if an installation specifies

an IOC coefficient of 3.0, whenever a job issues an I/O request, the system resources management routines multiply the I/O resource (one processing resource) by the service definition coefficient (3.0), and that job is considered to have used three service units.

Each of the service definition coefficients may be specified only once in an IPS. If a service definition coefficient is not specified, an IBM-supplied default value will be used.

#### **Service Rates**

Service rates are integers that define a target rate that system resources are to be supplied to an associated transaction. When grouped in sets called performance objectives, these define an intended service rate for any system workload condition.

### **Tuning Parameters (IEAOPTxx)**

The parameters in the SYS1.PARMLIB member IEAOPTxx include:

#### **Resource Factor Coefficients**

Resource factor coefficients are values in the range 0.0 to 9.9. They specify values that are used as multipliers in the control algorithm to weight the recommendation values produced by the I/O and CPU load balancing algorithms. The two coefficients are IOC and CPU. Each coefficient has a default value of 1.0. The coefficients are used to make the recommendation values produced by the corresponding algorithms either more or less important than the recommendation values supplied by the workload management routines, for which an implied coefficient of 1.0 applies.

#### **Resource Manager Constants**

This category contains the enqueue residence value (ERV). The ERV is an integer in the range 0-999999. It is used to calculate the amount of time that an address space will not be swapped because the address space is enqueued upon a system resource required by another address space. The time interval is determined by multiplying the ERV by the (model dependent) time required to execute 10,000 machine instructions. (This procedure is intended to keep the execution interval relatively constant on all System/370 models.)

## **System Activity Measurement Facility (MF/1)**

The system activity measurement facility (MF/1) is new in MVS. Installations may use it to monitor selected areas of system activity and obtain feedback in the form of trace records and/or formatted reports. Measurements may be gathered independently on:

- CPU activity
- Channel activity and channel-CPU overlap activity
- I/O device activity and contention
  - Unit record devices
  - Graphics devices
  - Direct access storage devices
  - Communication equipment
  - Magnetic tape devices
  - Character reader equipment
- Paging activity
- Workload activity

MF/1 activity is divided into three categories:

- SAMG (system activity measurement gathering) obtains the measurements of system activity requested by the installation.
- SARG (system activity report generation) produces formatted reports of desired system measurements.
- MFC (measurement facility control) controls the collection, tracing and reporting of system activity.

SAMG consists of distinct sets of measurement gathering (MG) routines associated with each class of measurement data. Only those sets associated with the measurement data requested by the system operator are loaded and active during MF/1 reporting intervals.

SARG generates formatted summary reports of information requested by the system operator, collected by SAMG, and routed to it by MFC. Printing of these reports may be done as they are generated, or may be deferred until MF/1 termination.

MFC is the system task that controls MF/1 operation. It causes the loading of the MG routines that will be needed to obtain the desired information, passes control to these routines at the desired intervals, routes collected information to SARG as required, and terminates MF/1 execution at the end of its specified duration.

### ***MF/1 Operation***

The operation of MF/1 is controlled by a set of parameters that may be contained in:

- The PARM field of the START command that is issued to start MF/1 processing.

- The PARM field of the EXEC statement in the MF/1 cataloged procedure.
- The MF/1 partitioned data set member (typically a SYS1.PARMLIB member).

The IBM-supplied cataloged procedure for MF/1 names the MF/1 partitioned data set member that contains card image MF/1 control input. Member names are of the form IRBMF1xx, where xx is a decimal digit field. In the absence of any explicit specification, the SYS1.PARMLIB member IRBMF100 is used.

The system operator issues the START command to begin MF/1 monitoring. The parameters controlling the operation of MF/1 are gathered from the above sources in the order stated. Explicit specification of an option in the PARM field of the START command will therefore override any alternative specifications of that option in any other source. If a parameter is not specified in any of the three sources the system default value is used.

The MF/1 options are summarized below. The underlined values are the system default values, as well as the values that appear in the SYS1.PARMLIB member IRBMF100.

CPU/NOCPU  
PAGING/NOPAGING  
CHAN/NOCHAN  
WKLD(PERIOD/GROUP/SYSTEM)/NOWKLD  
DEVICE(device list)/NODEVICE

These parameters are used to specify which of the classes of MG routines (CPU activity, paging activity, channel activity, workload activity, I/O device activity) will be loaded and executing during the MF/1 reporting period.

UNTR/NOUNTR  
TAPE/NOTAPE  
DASD/NODASD  
COMM/NOCOMM  
GRAPH/NOGRAPH  
CHRDR/NOCHRDR

These parameters are the elements of the device list above. They specify the device types (unit record, tape, direct access, communications, graphics, character reader) about which information will be collected.

REPORT(REALTIME/DEFER)/NOREPORT

This parameter specifies whether or not formatted reports of the gathered data will be generated, and when the reports will be generated (i.e., as the data is gathered, or after MF/1 processing terminates).

SYSOUT(class)

This parameter specifies the SYSOUT class to which formatted reports will be directed. Class A is the default.

RECORD/NORECORD

This parameter specifies whether or not trace records will be written to the SMF data set.

INTERVAL(value/valueM)

This parameter specifies the interval at which data will be gathered for report formatting and/or trace record writing. The range is from 1 to sixty minutes. The default is 15M.

**CYCLE(value)**

This parameter specifies the frequency at which observations are made of sampled data. The range is from 50 to 999 milliseconds. The default is 250 milliseconds.

**STOP(value/valueM/valueH)/NOSTOP**

This parameter specifies the desired time interval for MF/1 activity, in minutes or hours (the default is minutes). The operator STOP command will terminate execution at any time, regardless of the value for this parameter. The default value is 15M.

**MEMBER(nn)**

The value specified by this parameter is added to IRBMF1 to form the name of the partitioned data set member that contains the MF/1 options. The default is 00.

**OPTIONS/NOOPTIONS**

This parameter specifies whether or not a list of the keyword options to be used will be printed at the operator's console. If the list is printed, the operator will be able to respond with any desired changes to the option list.

***MF/1 Reports and Tracing Records***

Measurement output is produced in two forms:

- Printed Reports -- The data is formatted and printed for each class of system activity data collected.
- Trace Records -- The data is written as a trace record to the SMF data set. One or more trace records are produced for each class of system activity data collected.

The printed reports and trace records contain the same type of information, but in different forms. The installation can specify desired output as:

- Printed reports only.
- Trace records only.
- Both printed reports and trace records.

The following sections describe the contents of each type of printed report and trace record.

**CPU Activity Report**

This report provides information about the use of the central processing unit(s). The total CPU wait time that has taken place during an MF/1 reporting interval is generated for each CPU. The CPU wait time is also expressed as a percentage of the reporting interval time.

**Paging Activity Report**

This report provides detailed information about the demands made on the system paging facilities during the reporting interval. It also distinguishes between the use of real main storage and external page storage. The following categories of information are included:

- Page reclamations -- Page reclamation rates and percentages are produced, for system pageable areas and the user's private address spaces, for both VIO and non-VIO accesses.
- Page-ins -- Page-in rates and percentages are produced for all pages transferred from external page storage to real storage. This information is provided for system pageable areas and the user's private address spaces.
- Page-outs -- Page-out rates and percentages are produced for pages transferred from real storage to external page storage. This information is provided for system pageable areas and the user's private address spaces.
- External page storage counts -- Total page slots are produced, as well as the number of unused slots, unavailable slots, and slots containing either data set pages or virtual address space pages.
- Pageable real storage counts -- The total number of pageable real storage frames is produced, as well as breakdowns into counts and percentages of used and unused page frames.
- Swap counts -- The total number of swaps are produced, as well as average pages per swap-out, and average pages per swap-in.

### **Workload Activity Reports**

These reports provide information about the system workload activity and its relationship to the installation-specified performance objective for each performance group period. This information can be used to evaluate and modify the IPS. The three different workload activity reports that are available provide three different levels of detail. The types of reports desired are specified by input parameters indicated below from the most detailed report to the least detailed report.

- Performance group period report -- (WKLD(PERIOD))
- Performance group report -- (WKLD(GROUP))
- System summary report -- (WKLD(SYSTEM))

**Performance Group Period Report:** This report provides, for each performance group period within each performance group:

- The total number of service units used by all transactions in that performance group period.
- The average service rate for all transactions within that performance group period.
- The average workload level of all associated transactions.
- The average number of transactions that were concurrently active.
- The total number of transactions that terminated.
- The average elapsed time of a transaction that terminated.

**Performance Group Report:** This report provides, for each performance group, for one MF/1 reporting interval:

- The total number of service units used by all transactions in that performance group.
- The average number of associated transactions that were concurrently active.

- The total number of associated transactions that terminated during the reporting interval.

**System Summary Report:** This report provides, for one MF/1 reporting interval:

- The total number of service units used by all transactions.
- The mean workload level of the system.
- The average number of transactions that were concurrently active.
- The total number of transactions that terminated during the reporting interval.

### **Channel Activity Report**

This report provides information about channel loading for all channels in the system. For each active channel, a "channel activity" count provides the total number of successful SIO instructions (excluding the sense SIO instructions) issued to that channel during the reporting interval. A "percent channel active" indication provides the percentage of the total reporting interval during which each channel was busy. A "percent channel busy and CPU waiting" indication provides the percentage of the reporting interval during which each channel was busy while the associated CPU was in a wait state. An installation may use this information, in conjunction with the information contained in the I/O device activity reports, to determine possible bottlenecks associated with channels.

### **I/O Device Activity Reports**

These reports provide information about I/O device loading for all devices in the chosen device class(es). The classes desired are specified in the input parameter string (UNITR, TAPE, DASD, COMM, GRAPH, CHRDR). For each device report, a "device activity count" indicates the number of successful SIO instructions (excluding the sense SIO instructions) issued to the device during the reporting interval. A "percent busy" indication provides the percentage of the reporting interval during which the device was busy. A "mean queue length" value provides an average number of requests that were enqueued and waiting to be serviced on the device during the reporting interval. An installation may use this information, in conjunction with the information contained in the channel activity reports, to determine possible bottlenecks in the I/O subsystem.

### **SMF Recording**

If RECORD is specified among the MF/1 input parameters, a record will be written to the SMF data set, at the end of each reporting interval, for each report requested by the parameters. Each record contains information similar to the contents of the corresponding formatted report. Totals, averages, and percentages are not explicitly contained in the record, but may be calculated from the explicit data. The SMF records, and their corresponding printed reports are:

SMF70 -- CPU Activity Report  
 SMF71 -- Paging Activity Report  
 SMF72 -- Workload Activity Report  
 SMF73 -- Channel Activity Report  
 SMF74 -- I/O Device Activity Report



## Tuning With MF/1

Tuning a system is the process of modifying its hardware or software characteristics to bring performance closer to installation defined objectives. Once an installation's performance objectives are clearly defined, MF/1 can be used to help an installation both specify and meet them. For example, the channel activity report and I/O device activity report can provide information about the efficiency of the system-wide I/O load balance and how it affects system throughput. Analysis of these reports could indicate possible system configuration changes or data set residence modifications.

MF/1 can also be used to determine if the IPS reflects the installation's turnaround/response requirements and, if so, whether the IPS objectives are being met. It can further be used to analyze the specified performance objectives for possible modification.

### *Tuning Example*

Figure 14 shows the division of the installation's jobs into three performance groups. Assume that users associated with performance group 7 -- the time sharing users -- are experiencing poor response time at their terminals. The installation decides to activate MF/1 for a three hour period during a typical day, to obtain six reports on the workload activity for analysis. To initiate this reporting the operator enters:

```
START MF1.MF1,,(NOCPU,NOPAGING,NOCHAN,  
NODEVICE,INTERVAL(30M),STOP(3H))
```

(Assume that there are no EXEC statement parameters in the MF/1 procedure, and that the default SYS1.PARMLIB list IRBMF100 is used.)

With the above command and the standard defaults, the following options will control MF/1 execution in this case:

```
NOCPU  
NOPAGING  
NOCHAN  
WKLD(PERIOD)  
NODEVICE  
REPORT(DEFER)  
SYSOUT(A)  
NOTRACE  
INTERVAL(30M)  
CYCLE(250)  
STOP(3H)  
OPI
```

### IPS Values

Performance Group Number	Significance of Performance Group to Installation	FIRST PERIOD					SECOND PERIOD				
		Duration	Units	Performance Objective	RTB	ISV	Duration	Units	Performance Objective	RTB	ISV
7	Terminal Jobs	Interaction	-	2	-	-	-	-	-	-	-
8	High Priority Batch	3000	S	8	-	-	Rest of Job	-	8	-	-
9	Low Priority Batch	Entire Job	-	5	-	-	-	-	-	-	-

### Performance Objectives

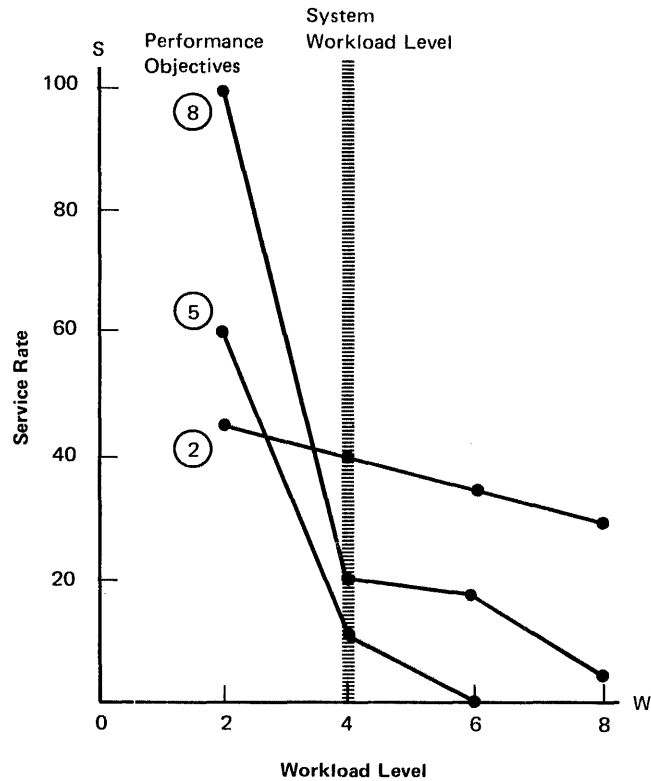


Figure 14. Tuning Example

Assuming that the operator does not prematurely terminate MF/1 execution with the STOP command, MF/1 will terminate at the end of three hours, and the six reports that were collected at half-hours intervals will then be printed. The installation can now begin its analysis.

Two figures of interest in this case are the average system workload level, and the average active time for all performance group 7 transactions. Suppose that these reports indicate that, during the time that MF/1 monitoring took place, the system was operating approximately at workload level 4 and that the average response time for time sharing users was 30 seconds (considered by the installation to be poor response time).

An analysis of the performance objective specification shows that, at workload level 4, performance objective 2 specifies that performance group 7 transactions receive 40 service units per unit time. The installation might reasonably wish to increase the service rate for its time sharing users in order to improve the response time. This increase would raise the performance objective curve for performance objective 2, as shown in Figure 15.

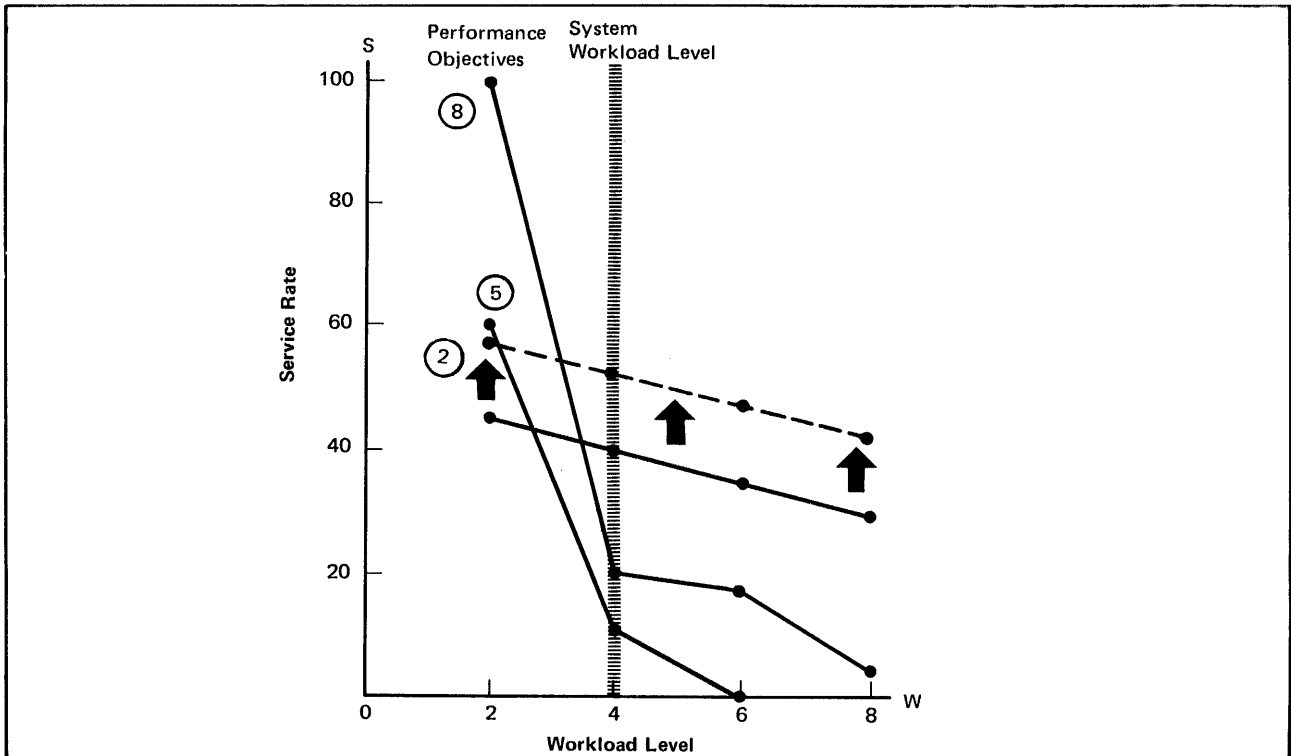


Figure 15. Changing a Performance Objective

Figure 16 shows that raising the service rate for performance objective 2 raises the system workload level, all other factors being equal. Transactions associated with performance objectives 5 and 8 would, therefore, be affected. Transactions associated with performance objective 5 would be most severely affected because their demand curve slopes most severely to the right of workload level 4.

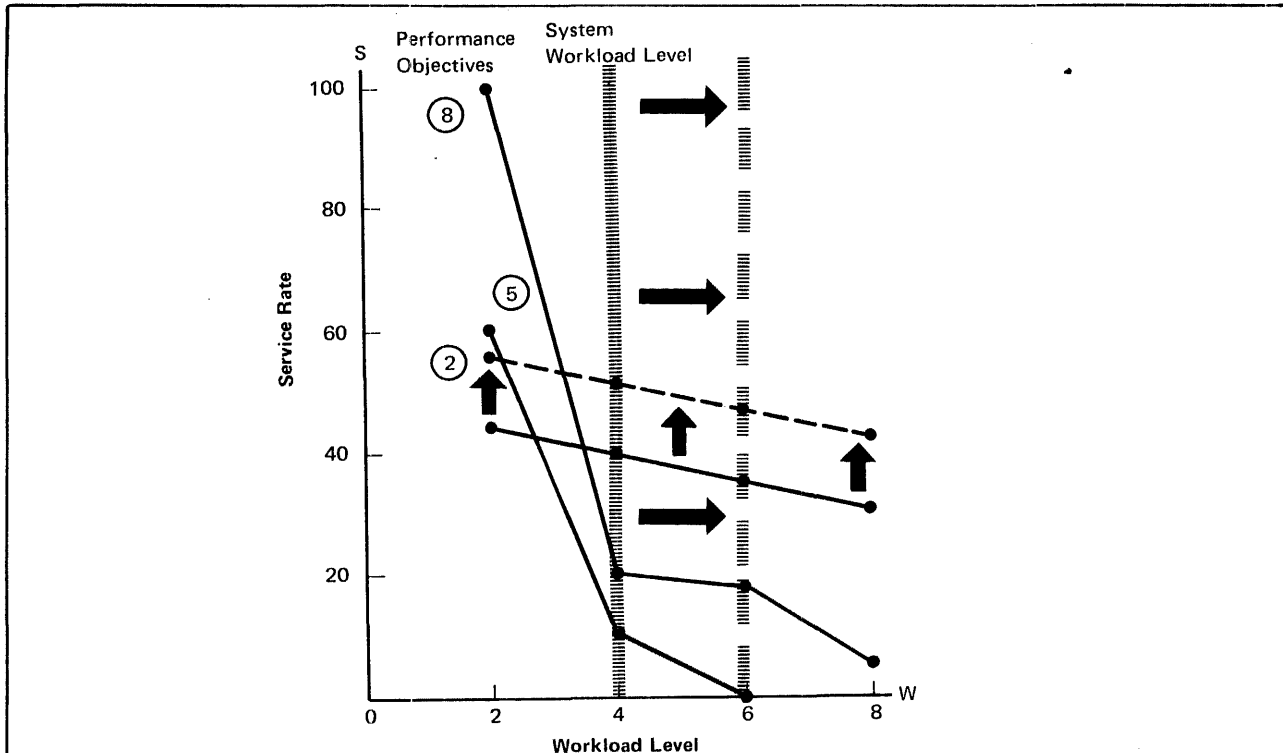


Figure 16. Effects of a Performance Objective Change

Before the installation modifies performance objective 2, it is able to perform further analysis to determine the quantitative effects of such a change. Suppose the MF/1 reports showed an average of 20 jobs associated with performance objective 2, 8 jobs with performance objective 5, and 4 jobs with performance objective 8. Then, some measure of the average system rate of service supplied to all jobs can be obtained by multiplying the number of jobs at each performance objective by the service rate for that performance objective at workload level 4 (i.e.,  $20 \times 40 + 8 \times 20 + 4 \times 10 = 800 + 160 + 40 = 1000$  total service units per unit time).

In this case, increasing the performance objective 2 curve by 10 service units, results in a total increase in demand of 200 service units (10 times the 20 jobs associated with performance objective 2). Thus, the workload level of the system shifts to the right (increases) to compensate for this increase in demand, while maintaining the system supply of 1000 total service units approximately constant. (See Figure 16.) If the installation considers the service rates provided for all performance objectives at the projected new workload level to be acceptable, it may modify the IPS to contain the new performance objective 2.

The installation may then use MF/1 monitoring to verify the accuracy of the projections; repeating the entire procedure, if desired, until satisfactory results are achieved.

The preceding example is not meant to reflect actual figures or relationships, but rather to indicate the possibilities for tuning that MF/1 can provide. The illustrated method of improving response time is neither the only one, nor necessarily the best -- other alternatives include varying the interval service value parameter, dividing the time sharing performance group into more than one period, and increasing the number of performance groups provided for time sharing users. Any method chosen to improve this or other system performance characteristics will suggest other uses and variations for MF/1.



## System Integrity

A highly desirable property of an operating system is its ability to insure that one program cannot interfere with or modify another program's (system or user) execution unless it is authorized to do so. For reliability and system availability, this is extremely important; for data security, it is essential. MVS provides this capability in the form of system integrity support.

System integrity is defined as the ability of the system to protect itself against unauthorized user access to the extent that security controls cannot be compromised. That is, there is no way for an unauthorized\* program using any system interface to:

- Bypass store or fetch protection, i.e. read or write from or to another user's areas.
- Bypass password checking, i.e. access password protected data for which a password has not been supplied.
- Obtain control in an authorized state.

In MVS all known integrity exposures have been removed. IBM will accept as valid, any APAR that describes an unauthorized program's use of any system interface (defined or undefined) to bypass store or fetch protection, to bypass password checking, or to obtain control in an authorized state.

---

\*An authorized program in MVS is one that executes in a system key (keys 0-7), in supervisor state, or is authorized via the Authorized Program Facility (APF).

## **Installation Responsibility**

To ensure that system integrity is effective and to avoid compromising any of the integrity controls provided in the system, the installation must assume the responsibility for the following items:

The installation must be responsible for the physical environment of the computing system. Operations personnel and system programmers have, in effect, uncontrolled access to certain portions of the operating system. Problems of this nature are discussed in **The Considerations of Physical Security in a Computer Environment**, G520-2700.

The installation must be responsible for the adoption of certain procedures (e.g. the password protection of appropriate system data sets) that are a necessary complement to the integrity support within the operating system itself. The most significant of these procedures are described briefly in the topic "Installation Procedural Responsibilities."

The installation must ensure that its own modifications and additions to the control program do not introduce any integrity exposures. That is, all installation-written authorized code (e.g. an installation SVC) must perform the same or an equivalent type of validity checking and control that the MVS control program employs to maintain system integrity.



## Areas of Concern

Several areas with the potential for integrity exposures have been identified. These areas, and what has been done to eliminate them as potential sources of integrity exposures, are described below. The description is a guideline to aid the installation in preserving system integrity in any modification or addition to the control program. It is also intended to aid in determining whether an impact on existing installation code might occur, where that code is dependent on the use of non-standard interfaces to the control program. There should be no impact on installation-written routines that use standard interfaces (problem program/system interface described in an SRL), because no standard interface has been removed from the system control program in connection with system integrity support. MVS system integrity support is concerned only with restricting the unauthorized problem program; there is no attempt or intention to prevent the installation manager or any authorized system programmer from modifying the system control program.

### ***User-Supplied Addresses for User Storage Areas***

A potential integrity exposure exists whenever a routine having a system protection key (key 0-7), accepts a user-supplied address of an area to which a store or fetch is to be done. If the system routine does not adequately validate the user-supplied address to ensure that it is the address of an area accessible to the user for storing and fetching data, an integrity violation can occur when the system key program routine:

- Stores into (overlays) system code or data (e.g. in the nucleus or the system queue area), or into another user's code or data.
- Moves data from a fetch-protected area that is not accessible to the user (e.g. the fetch-protected portion of the common service areas), to an area that is accessible to him.

The elimination of this problem requires that system key routines always verify that the entire area to be stored into, or fetched from, is accessible (for storing or fetching) to the user in question. The primary validation technique is the generally established MVS convention that system key routines obtain the protection key of the user before accessing the user-specified area of storage. For example, MVS data management SVC routines (which generally run in key 5) assume the user's key before modifying a data control block (DCB) or an I/O block (IOB).

### ***User-Supplied Addresses for Protected Control Blocks***

A potential integrity exposure exists whenever the control program (system key/privileged mode) accepts the address of a protected system control block from the user. For most system control blocks this situation should not be permitted to exist. However, in certain cases it is necessary to allow the user to provide the address of a system control block that describes his allocation/access to a particular resource (e.g. a data set), in order to identify that resource from a group of similar resources, (e.g. a user may have many data sets allocated). Inadequate validity checking in this situation can create an integrity exposure, since an unauthorized problem program could provide its own (counterfeit) control block in place of the system block and thereby gain the ability to:

- Access a resource in an uncontrolled manner (since the control block in this case would normally define the restrictions, such as read-only for a data set, on the user's allocation to the resource).
- Gain control in a privileged state (because such control blocks may contain the addresses of routines that run in privileged mode or with a system (0-7) key).
- Cause various other problems depending on exactly what data the system may be keeping in the control block involved.

To avoid this type of exposure, the control program must verify, for every address of this type that is accepted from a problem program, that the address is that of:

1. A protected control block created by the control program.
2. The correct type of control program block (e.g. a TCB versus a DEB, or a QSAM DEB versus an ISAM DEB).
3. A control block created for use in connection with the user (job step) that supplied the address.

In MVS, verification is generally accomplished by establishing a chain or table of the particular type of control block to be validated. This chain or table is located via a protected and job step-related control block that is known to be valid (i.e. whose address is not allowed to be supplied by the user, and is located via a chain of protected control blocks that begins with a control block known to be valid or fixed at a known location at IPL time, such as the CVT\*). Thus a control block can only be entered in the chain/table by an authorized program, which satisfies (1) above; by definition the chain/table establishes the type of control block, which satisfies (2) above; and also by definition each chain/table is located only through a job step-related control block, satisfying point (3).

## ***Resource Serialization***

Resource serialization is another potential source of integrity problems. Here an integrity exposure can result when serialization controls on sensitive control program resources are nonexistent, or inadequate to the extent that an unauthorized problem program can directly, or indirectly through a part of the control program (e.g. by issuing an SVC), change the content or status of a system resource (e.g. control block, work area), while another portion of the control program (e.g. another SVC) is using that resource or is in some way dependent on its content or status remaining unchanged for a given period of time.

Adequate control of serial resources becomes even more significant in a multiprocessing environment since some uniprocessor serialization techniques (e.g., disablement), are no longer effective because of the possibility of multiple tasks, even multiple tasks in the same job step, using the same serial resource and running concurrently on different CPUs.

To eliminate this as a potential exposure, a locking technique is used to serialize access to the resources in question. However, a locking technique is not

---

\*This does not imply that a system routine must go back to the CVT or similar control block every time it wants to establish a valid chain. Typically, a control block address not too far down on such a chain is available already validated in a register. For example, the first load of an SVC may receive control with a valid TCB address in a register.

effective unless all routines that have the potential for changing the resource, or that depend on its status remaining unchanged for a given period, make use of the (same) locking mechanism. Effective use of a locking technique therefore requires considerable investigation and effort, to determine and keep track of system resources that must be serialized and routines that access such resources.

In MVS, a combination of ENQ/DEQ and a new hierarchical locking structure with multiple types of locks, is the primary method used to synchronize the use of serial resources. The locking mechanism is restricted to key 0 programs, which prevents unauthorized problem programs from interfering with the serialization of system resources. For the same reason, ENQ/DEQ is restricted to authorized programs for all major names of the form SYSZxxxx, and for the following explicit major names: SYSDSN, SYSIEECT, SYSIEFSD, SYSIEA01, SYSVTOC, SYSPSWRD, SYSCTLG, SYSIGGV1, SYSIGGV2, and SYSVSAM.

Another integrity consideration, related to the above problem, but generally applicable to all validity checking for integrity purposes, is the time-of-check-to-time-of-use problem. This refers to the fact that from the time of a validity check until the completion of the operation associated with that validity check, the variables on which the outcome of the validity check is based must not be allowed to change to the extent that the result of the validity check would be changed. While this is a relatively obvious statement, the requirement it imposes has considerably influenced the direction of the integrity support in MVS.

In some cases, this requirement can be met "automatically" through use of hardware serialization mechanisms, such as checking the validity of fetch/store operations by assuming the user's protection key. In other cases, techniques such as saving validated data in protected (storage keys 0-7) core and use of the above-mentioned locking mechanisms are required.

### ***Resource Identification***

Resource identification is another area that can be subject to integrity exposures. Exposures can result if the control program does not maintain and use sufficient data to uniquely distinguish one resource from other similar resources. For example, a program must be identified by both name and library to distinguish it from other programs. The consequences of inadequate resource identification are problems such as the ability of an unauthorized problem program to create counterfeit control program code or data, or to cause varying types of integrity problems by intermixing incompatible pieces of control program code and/or data.

The general solution can only be stated as the reverse of the problem; that is, the control program must maintain and use sufficient (protected) data on any control program resource, to distinguish between that resource and other control program or user resources. The following are examples of the controls that MVS employs to comply with the requirement:

- In general, authorized program requests to load other authorized programs are satisfied only from authorized system libraries (see the topic "Control Program Extensions").
- MVS takes explicit steps to ensure that routines loaded from authorized system libraries are used only for their intended purpose. This includes expanded validity checking to remove any potential for the unauthorized

program to specify explicitly which of the authorized library routines are to gain control in any given situation.

- Sensitive system control blocks are validated as being the "correct" blocks to be used in any given control program operation. (See the topic "User-Supplied Addresses of Protected Control Blocks").

### ***SVC Routines Calling SVC Routines***

A potential problem area exists whenever a problem program is allowed to use one SVC routine (routine A) to invoke a second SVC routine (routine B) that the problem program could have invoked directly. An integrity exposure occurs if SVC routine B bypasses some or all validity checking based on the fact that it was called by SVC routine A (an authorized program), and in addition, user-supplied data passed to routine B by routine A either is not validity checked by routine A, or is exposed to user modification after it was validated by routine A. This problem will not exist if the user calls SVC routine B directly, because the validity checking will be performed on the basis of the caller being an unauthorized program.

SVC routine A, which is aware that it has been called by an unauthorized program, must ensure that the proper validity checking, etc., is accomplished. However, it is usually not practical for SVC routine A to do the validity checking itself, because of the potential for user modification of the data prior to or during its use by SVC routine B. The general solution should be for SVC routine A to provide an interface to SVC routine B, informing routine B that the operation is being requested with user-supplied data in behalf of an unauthorized problem program (implying that normal validity checking should be performed).

In practice, in MVS, most SVC B type routines that could be subject to this problem, use the key of their caller as a basis for determining whether or not to perform validity checking. Therefore most SVC A type MVS routines have simply adopted the convention of assuming the key of their caller before calling the SVC B routine.

### ***Control Program and User Data Accessibility***

Important in maintaining system integrity, is the consideration of what system data is sensitive and must be protected from the user, and what data can be exposed to user manipulation. The implications of the exposure of the wrong type of data are obvious.

In general, it is necessary to protect the following types of data:

- Code, and the location of code, that is to receive control in an authorized state.
- Work areas for such code, including areas where it saves the contents of registers.
- Control blocks that represent the allocation or use of system resources.

MVS maintains items such as the above in system storage, or in a separate region in the case of some APF-authorized (see the following section) programs. It may also be necessary to protect, for a limited period, certain data that is normally under the control of the user (e.g. to prevent its modification during a critical operation). In this case MVS provides fetch protection for such data if:

- The data consists of proprietary information (e.g. passwords).
- The control program cannot determine the nature of the contents of the data area.

### ***Control Program Extensions***

The final potential problem area involves the fact that there exists a somewhat hazy distinction between the control program and certain types of problem programs. In most installations, there are problem state/user key (keys 8-15) programs that are actually extensions to the control program in that they are allowed (by means of various special SVCs, etc.) to bypass normal system controls over access to system resources. For example, a special utility program that scans all the data on a pack might be able to avoid the normal system extent checking on a direct access volume. The Authorized Program Facility (APF) was introduced in VS2 Release 1 as a means of identifying these control program extensions as "authorized" programs, and restricting the use of the various "special" SVCs used by such programs to bypass normal system controls. This use of APF is continued in MVS with the restriction of SVCs under APF extended to include certain new SVCs introduced in MVS.

If an installation has its own control program extensions and special SVCs that allow the bypass of normal system security or integrity checks (e.g., an SVC that returned control in key 0), and if such SVCs are not currently restricted from use by an unauthorized program, then the APF facility should be used to restrict these SVCs and to authorize the control program extensions that use them. A description of how to use the APF facility can be found in appropriate VS2 publications.

An important MVS addition is that APF has been extended to recognize and accept authorized programs from "authorized" libraries other than SYS1.SVCLIB, SYS1.LINKLIB, and SYS1.LPALIB. Any library may be designated as an authorized library by inclusion of its name in the SYS1.PARMLIB member, IEAPF00 or IEAPFxx, prior to IPL. Such libraries are the equivalent of the above system libraries in that nonspecific system requests (library not specified) for routines that will execute with system key, privileged mode, or APF authorization, can be satisfied by a module loaded from any authorized library.

APF supports a facility (TESTAUTH) used to control the execution of certain paths through SVCs, such as the path that opens (OPEN) the VTOC. This facility may not be used to control the use of I/O appendages in MVS. Appendages are controlled through SYS1.PARMLIB as described above.

**Note:** The user of APF should also be aware that APF authorization is defined such that any SVC that is restricted by the APF mechanism, can be executed by any system key (0-7) or (in most cases) privileged mode routine. However, this is not true in reverse. An APF authorized program will not automatically be allowed system services that are restricted by system key or privileged mode tests.

## Installation Procedural Responsibilities

As noted previously, the installation is responsible for the implementation of procedures that are necessary in certain instances to complement the MVS control program integrity/security support. These responsibilities are discussed below. Note that in some instances, the concern is for basic system integrity, while other items are concerned with a particular type of security.

1. The installation must password protect appropriate system libraries. System integrity obviously cannot be maintained if system code and data is exposed to arbitrary modification by any user on the system. For integrity purposes, it is generally sufficient to protect appropriate libraries from write access; no password is required for read access but a password is required for write access. However, for security purposes, it is necessary to protect certain system data sets (e.g. the PASSWORD data set itself) from read as well as write accesses. To facilitate the protection of such data sets, during IPL and system task initialization password requests are suppressed for data sets being opened by the system.

Certain auxiliary storage manager (ASM) initialization procedures make password protection of the SYS1.STGINDEX data set ineffective. However, ASM provides protection via an ENQ on the data set name, effectively preventing another user from allocating the data set.

2. Because the checkpoint data set produced by the Checkpoint/Restart facility contains sensitive system data normally protected from the user, maintaining system integrity requires that such data sets be protected from modification (or from being counterfeited) prior to their use by Restart. MVS implements a facility whereby the installation can adopt a set of special procedures/controls over checkpoint data sets that will eliminate their potential for compromising system integrity. The control mechanism involves a combination of:
  - System/operator validation of checkpoint data sets.
  - External labeling procedures for "checkpoint volumes."
  - Offline control of "checkpoint volumes."
  - Prohibiting I/O to checkpoint data sets, except through the checkpoint SVC (APF-authorized programs excepted).
3. The installation should be aware of the following password protection requirements, which may necessitate some changes in installation procedure.
  - Because of the characteristics of magnetic tape devices, once a program has access to one file on a tape, it can not be ensured that the program is not able to access other files on that tape, even though it may not possess passwords for those files. For this reason, MVS has implemented a change in the handling of password-protected tape data sets to attempt to ensure that if a user is allowed access to one data set on a tape volume, he can be presumed also authorized to other data sets on the volume. To this end, whenever a user attempts to create a tape data set whose file sequence number is greater than 1, the protection attributes of the new file will be compared against the protection attributes of the immediately preceding file on the tape (no protection, protected for write, protected for read and write, etc.). If the protection attributes do not match, the new data set will not be created. In addition, if the previous data set is password protected, its password must be supplied before the new data set will be created.

Note that these changes do not eliminate problems caused by an installation library containing already-existing tape files with mixed protection attributes.

- The security-conscious installation should ensure that password-protected data sets are created only on tapes which are unused or completely erased. This is necessary because in certain cases residual information left on the tape after the password data set was created, could result in a user being able to open one of the previous data sets on the tape, thus gaining access to the whole tape.
4. In the case of the 2840/2250, the installation may want to use dynamic allocation and JCL validation exits to achieve additional control over device allocation and user identification. Fetch/store protect is not available for the 2840 buffer, which can be shared by up to four 2250 graphic devices. The above-mentioned exits can be used to establish a control mechanism that restricts to users of the same security classification, 2250's that are connected to the same 2840.
  5. Although APF-authorized programs bypass the usual system control validity checks, in any user interfaces they must provide validity check mechanisms the same as or equivalent to the system validity checks. The effectiveness of APF is dependent on this. However, it may be impractical to provide these checks for some APF-authorized programs, and in these cases their use should be restricted by placing them in APF-authorized password protected libraries. The passwords for these libraries should then be provided only to individuals authorized to use the programs. Installations might consider using this control mechanism for the IEHINIT, OLTEP, and IEHDASDR routines.
  6. If there are two modules with the same name on different authorized libraries and an authorized program attempts to load one of these, it could get the other (either accidentally or deliberately with JOBLIB, STEPLIB, or concatenation). This type of exposure requires the additional restriction that duplicate module names not be permitted across authorized libraries.
  7. The optional update password for VSAM catalogs provides protection against unauthorized delete or alter operations on non-VSAM data set entries; operations on VSAM data set entries are protected under the password for the data set. Catalogs (e.g., the master catalog) containing sensitive non-VSAM data sets should be protected under the update password.
  8. OS CVOLs should be password protected (one password for all CVOLs under the name SYSCTLG) to avoid concurrent access between OS catalog routines and an unauthorized user. However, this does not prevent unauthorized delete or alter operations on the catalog. This protection is not offered for OS CVOLs.
  9. Installation-written programs (e.g., I/O appendages) that execute in supervisor state or system key (0-7) in the user's address space, must be link edited reentrant to ensure that they are loaded in a protected subpool.





## Conversion Considerations

This chapter describes significant differences and incompatibilities between OS/MVS and:

- OS/MVT
- OS/VS2 Release 1.

It is divided into the following sections:

- Job entry subsystem considerations.
- SMF considerations.
- Job control language.
- Communication between address spaces.
- Allocation considerations.
- Operator commands.
- Time sharing considerations.
- Catalog support.
- Data set conversion.
- V=R programming considerations.
- Program conversion.
- Multiprocessing considerations.

## **Job Entry Subsystem Considerations**

The job entry subsystems available with MVS are:

JES2 -- generally compatible with HASP II.

JES3 -- generally compatible with ASP Version 3.

With the first release of MVS, only JES2 will be available; therefore JES3 information reflects current design and is for advanced planning purposes only.

All jobs, started tasks, and LOGONs processed by the system must be entered through a job entry subsystem, which must therefore be started before any processing is done. The primary job entry subsystem is started automatically at system initialization. Any other is started by the operator.

### ***The JES2 Job Entry Subsystem***

JES2 functions as a job entry and system output processor. It also starts and stops system initiators, selects jobs to be processed by initiators, and frees resources when the jobs are complete. JES2 controls a job prior to execution and following job termination.

## JES2 Job Flow

Figure 17 is an overview of the JES2 processing stages. Input, execution, output, and purge services execute in the JES2 address space. Other functions depicted execute in system or user address spaces.

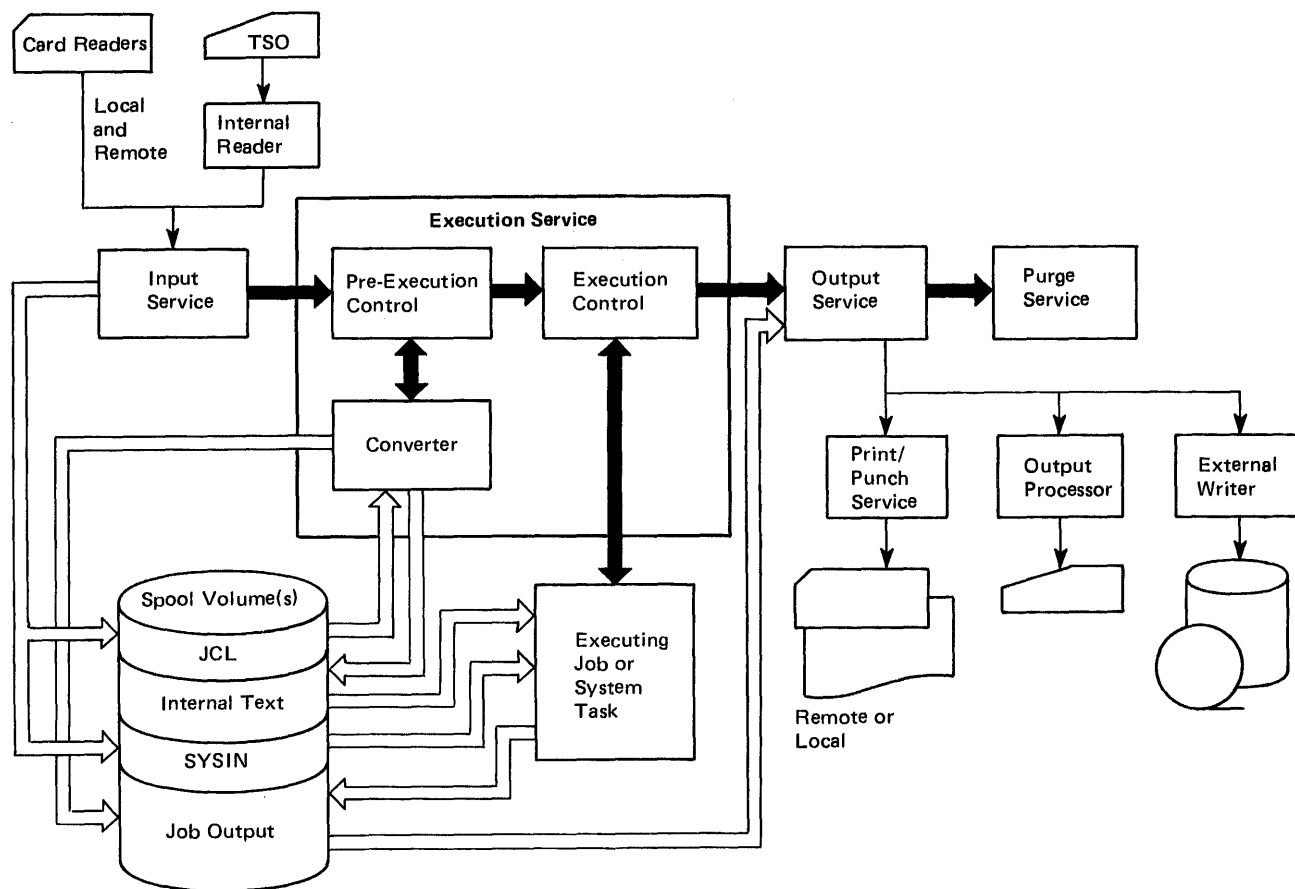


Figure 17. JES2 Processing Stages

**Input Service:** Jobs are read into the system from various types of card readers and remote terminals, and through the internal reader interface. An internal reader is an output (SYSOUT) data set allocated to JES2, which is recognized and placed in the input stream. Jobs and system tasks submit jobstreams in this manner.

Each job or user is given a unique job identifier; then the JCL and input data are stored on direct access volumes for retrieval during Execution Service.

**Compatibility:** Input service functions as a complete replacement for the OS reader; however no catalog procedures are associated with this function. Procedure libraries and interpretation parameters are associated with jobs by JES2.

Prior to execution, jobs are maintained on JES2 execution queues rather than on SYS1.SYSJOBQE. Pre-execution manipulation must occur through the use of JES2 commands rather than system commands.

Jobs with JCL syntax errors fail during conversion. The job is sent directly to output service, thus the JCL is not interpreted.

Support of the DLM parameter on DD \* or DD DATA JCL statements is compatible with OS support.

JES2 ignores the "//" (null) statement. The only effect of this statement is a possible higher input record count than MVT.

**Execution Service:** Through this service, jobs are submitted to the system for execution. Conversion takes place immediately after input service. The JCL is converted into internal text, unless syntax errors are discovered. Job selection takes place as soon as an initiator eligible to process the job is available. Jobs are selected in priority sequence within their class.

The subsystem interfaces for the following functions are supported through execution service:

- Subsystem job selection (including presenting jobs for warmstart).
- Allocation/unallocation of JES2 data sets.
- Sysout output interfaces (TSO output TMP).
- Job status.
- Job cancel.
- End of storage.
- End of task.
- WTO/WTOR/DOM messages.
- Command processor.
- Verify remote user id.
- Job termination.
- Job re-enqueue.
- OPEN/CLOSE of JES2 data sets.

**Compatibility:** JES2 does not support the Execution Task Monitor of OS/MVT HASP systems. The System Resources Manager of MVS accomplishes the same functions, with no defined interface to JES2.

All sysin/sysout data sets are maintained by JES2.

JES2 provides an access method for reading and writing the spool data sets in response to user requests. This method of accessing spool data is different from that provided by OS/MVT HASP.

JES2 is responsible for presenting jobs for warmstart or restart.

JES2 supports 36 job classes rather than 15. (A-Z, 0-9)

EXCP to a JES2 data set is not supported.

NOTE/POINT to DCBs processing JES2 data sets is not supported.

JES2 supports the IEFUSO SMF exit.

**Output Service:** The print data sets and system messages created during execution are printed; the punch data sets are punched.

JES2 operator commands are used to control JES2 printer/punch devices. Operators at remote stations can control only those JES2 devices attached to their particular station.

When each group of job-related data sets and each spinoff data set are processed, output service writes one SMF type 6 record to the SMF data set.

**Compatibility:** JES2 does not attach user-written sysout writers. The external writer interface must be used for special sysout processing.

**Purge Service:** When all processing required for a job is completed, the direct access space maintained by JES2 and all JES2 resources associated with the job are made available. The SMF purge exit is scheduled and the SMF purge record written.

**JES2 Warmstart** For a warmstart occurring under JES2, all spool volumes that were up during the last execution must be available, although they need not be mounted on the same drives.

Jobs with no job journal or with an irretrievable journal are not presented to the system for warmstart processing.

New jobs may be presented to the system and will be processed in priority order -- ahead of lower priority warmstart jobs.

## MVS Operational Changes

JES2 performs the reader function of the MVT Reader/Interpreter, and functions of the MVT Output Writer and Remote Job Entry. Unit record devices used for reading jobs and printing or punching output, and teleprocessing lines used for accessing remote stations are normally assigned to JES2. Thus a device, even though idle, may need to be disassociated from JES2 before it can be used by other system jobs or tasks.

Converter/Interpreter parameters formerly placed in the Reader Procedure PARM field for each specific reader, are included in the JES2 job class related parameters established during JES2 initialization. All catalogued procedure libraries to be used by jobs, LOGON, or system tasks must be included in the JES2 catalog JCL. JES2 class-related parameters and Job Control Cards associate the procedures with jobs by job class.

A card reader can be dedicated to JES2 in such a manner that the operator need only ready the device to cause JES2 to read a jobstream.

A catalogued procedure is provided to read a job stream from any QSAM-supported device through the internal reader interface. This method is used to read jobs from local devices other than card readers; it must also be used if input stream positioning is desired.

Column binary cards must be read by an installation-written routine and passed to JES2 through the internal reader interface.

The OS/360 standard separator page is not supported -- the special JES2 separator page is a combination of those provided by MVT HASP and OS/360.

An external writer is provided to support user-written output writers and the standard OS/360 separator page. This allows a user to write data to devices not directly supported by JES2 or JES3 (e.g., tape and disk), and to use installation-written job separator routines.

Remote stations may be attached with point-to-point based or dial lines. Multi-point support is not provided.

The direct sysout writer is not supported. A sysout data set is available for output service processing prior to the end of the job if the data set is unallocated at CLOSE. This facility is called sysout spinoff.

JES2 output service attempts to minimize forms, UCS, and FCB loading while processing output data sets. Operator interaction is minimized, and data may be processed in an order different from MVT. A JES2 generation option allows an installation to specify that output in the message class of a job be processed contiguously.

Sysout data sets and system messages are processed directly by JES2 and written into the JES2 spool data set, rather than into separate direct access data sets. These data sets may be retrieved at a time sharing terminal or by an external writer. Data sets freed by a foreground user are immediately available for output processing by JES2, or for inspection via the OUTPUT command.

A time sharing user can use JES2 control cards to specify job-related parameters for his submitted job.

JES2 schedules the IEFUSO SMF exit and the SMF purge exit, and writes the SMF output, purge, and subsystem activity records (types 6, 26, and 43-46 respectively).

## JES2/HASP Compatibility

Figure 18 describes some areas of compatibility between HASP and JES2. Some additional items and further clarification follow:

MVT/HASP	VS2 Release 1/HASP	JES2
Tape input (via HASP START command)	Internal reader (via a cataloged procedure)	Internal reader (via a cataloged procedure)
Execution task monitor	Dynamic dispatcher	System resources management routines
Binary synchronous code adapter and synchronous transmitter receiver communications facilities	Binary synchronous code adapter communications facilities only	Binary synchronous code adapter communications facilities only
Pseudo devices for SYSIN/SYSOUT via QSAM, BSAM, or EXCP	Pseudo devices for SYSIN/SYSOUT via QSAM, BSAM, or EXCP	Subsystem interfaces for SYSIN/SYSOUT via QSAM and BSAM only
Installation-written output writers only via the system output writer	Installation-written output writers only via the system output writer	Subsystem interface allows installation-written routines to obtain SYSOUT data sets from JES2
HASP console services <u>or</u> multiple console support	Multiple console support with HASP interfaces	Multiple console support with JES2 interfaces
Purge card	Purge record written to SMF data set	Purge record written to SMF data set
System output queued by four types	System output queued by forms, printer set-up, and 36 output classes	System output queued by forms, printer set-up, and 36 output classes
No equivalent function	Forms, printer set-up, and routing information may be specified on a HASP OUTPUT control card	Forms, printer set-up, and routing information may be specified on a JES2 OUTPUT control card
Restricted checkpoint/restart support for jobs using SYSIN/SYSOUT	Restricted checkpoint/restart support for jobs using SYSIN/SYSOUT	Full checkpoint/restart support
No SMF SYSIN record counts No SMF output limiting support No SMF output record	No SMF SYSIN record counts No SMF output limiting support SMF output record is written to the SMF data set	Full SMF support
Numeric forms designation	Alphanumeric forms designation	Alphanumeric forms designation
TSO SYSOUT data sets are written by the system output writer	TSO SYSOUT data sets are written by the system output writer	Time sharing SYSOUT data sets are processed by JES2 and are retrievable via the OUTPUT command
Background jobs scheduled from a terminal must use system facilities rather than HASP	TSO users may submit jobs to HASP for scheduling, inquire of their status, and cancel them	Time sharing users may submit jobs to JES2 for scheduling, inquire of their status, and cancel them

Figure 18. JES2 Differences From MVT/HASP and VS2 Release 1/HASP

- JES2 is distributed on the MVS Starter System rather than on a separately ordered component tape. Also, JES2 is automatically started by the system at IPL, rather than by the operator.
- During initialization, if errors are found in the arrangement of mounted spool volumes the operator is informed and JES2 waits for the situation to be corrected.

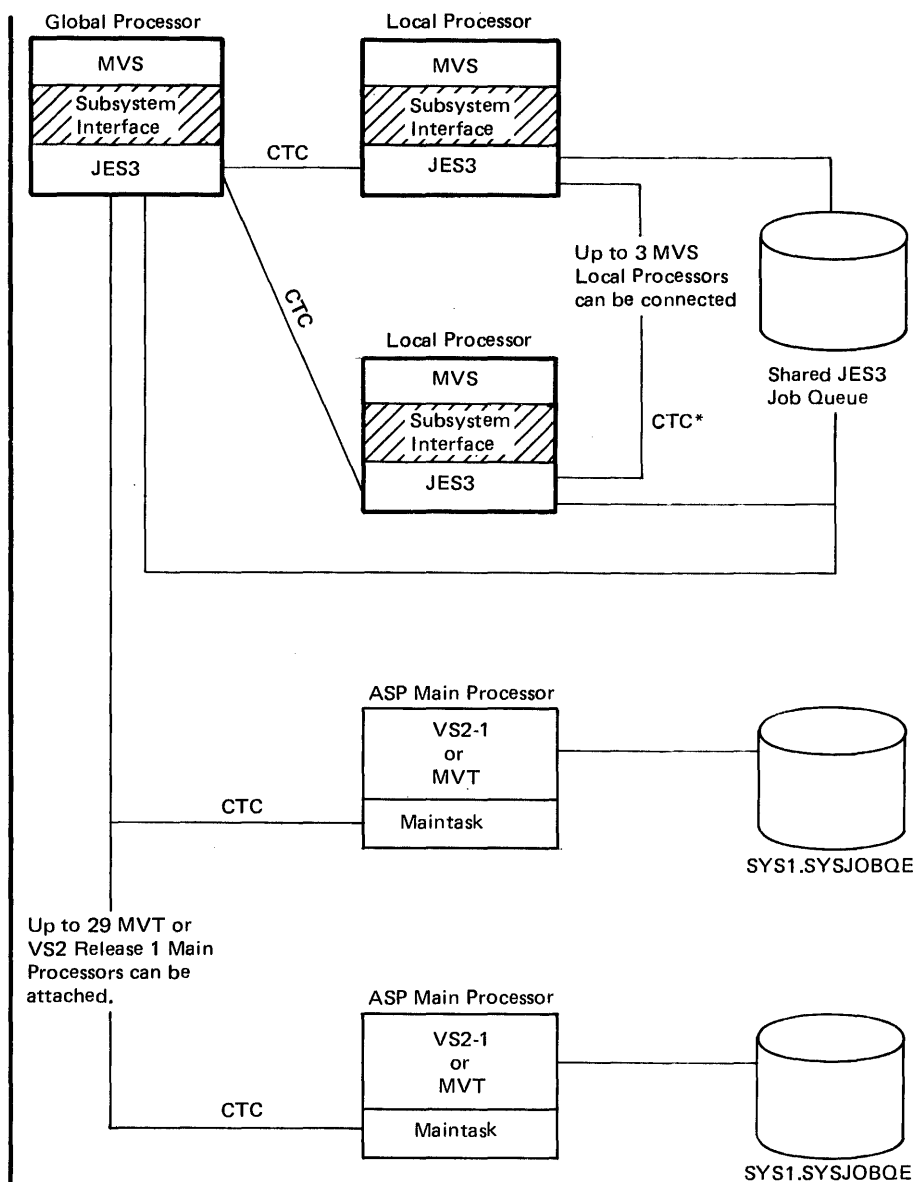
- Pseudo-devices are no longer used for specification of input and output devices and internal readers. System interfaces are provided with data sets identified by JES2-generated internal names, and the Internal Readers identified by the presence of the name INTRDR as the user-written Writer name in the SYSOUT parameter of the DD statement.
- Extended setup control affects the operation of output devices. Output service dequeues data for output devices based on output classes associated with each output device, rather than according to the characteristics of the device (e.g. special forms printer). Any of the 36 valid output classes may be specified for a given output device. Default classes specified during job entry subsystem generation can be overridden by the operator. The order of appearance of data sets on an output device may be different from MVT HASP for a given job based on special forms requirements. SYSOUT is sent directly to JES2; otherwise output service processing is compatible between JES2 and VS2 Release 1 HASP.

### ***The JES3 Job Entry Subsystem***

JES3, which includes some scheduling functions not provided by JES2 plus support for loosely-coupled systems, is designed for the user with a medium-to-large job-shop environment. As with JES2, JES3 replaces the OS readers and writers, and SYS1.SYSJOBQE. Figure 19 shows a possible JES3 multiprocessing configuration, with MVS and other systems connected via channel-to-channel connectors (CTCs). JES3 executes in all processors. The controlling processor in a JES3 configuration is called a Global processor; under ASP it was called the "Support" and "Local Main" processor. The other JES3 processors are called Local processors; under ASP they were called "Main" or "Real Main" processors. The Global processor controls job input, processes job output, and gives commands to the Local processors through the CTCs. From the global console, the multiple processors of a JES3 complex appear as a single system.

The primary enhancement to the ASP concept of loosely-coupled systems is the addition of dynamic system interchange (DSI). In case the Global processor fails, DSI allows the functions of the Global processor to be switched (operator-assisted) to any attached MVS Local processor. This processor then becomes the Global processor for itself and any MVS or MVT processor which is attached to it.





\* A CTC connection between Local processors is recommended (not required) to allow JES3 dynamic Global switching in the event of Global failure.

Figure 19. Sample JES3/ASP Multiprocessing Configuration

## JES3 Scheduling Features

Some of the features which enhance loosely-coupled systems and assist uniprocessing environments are:

- Preexecution setup is provided for direct access, graphic, unit record, and tape devices which are not assigned to JES3 for job input and output functions. JES3 determines the device and volume requirements for each job.
- Selection of a job for a processor is based on the ability of that processor to provide sufficient devices. JES3 switches pooled devices among processors to further control job/processor selection.
- A user exit is provided to determine which processor is eligible to execute a given job. Information regarding current workload is available at this exit.
- The operator is instructed to premount the volume(s), after the eligible processors are determined and the devices are reserved for the job.
- Optionally, preexecution setup need occur for only the largest number of devices used in any one step of the job. In previous releases (through ASP 3.0), the number of devices setup was the total number of devices used in all steps. For example:

Device	# used in step 1	#used in step 2	#used in step 3	ASP total	JES3 total
2314	2	4	1	7	4
2400	3	0	5	8	5

- Multiple jobs on different processors can share direct access volumes. Data sets, with disposition specified as shared, can also be shared between processors.
- JES3 volumes can be set up dynamically for dynamically allocated data sets. These devices are returned to the system when the data sets are deallocated.
- Dependent job control (DJC) allows simple or complex job interdependencies that are present in many commercial data processing installations, to be scheduled through JES3. The success or failure of one job under DJC can cause execution, holding, or cancelling of other dependent jobs.
- Devices can be pooled for jobs under DJC, or for job-class groups. This allows devices to be reserved for jobs with data set dependencies.
- Deadline Scheduling increases the probability of a job completing by a specified time. The selection priority of the job is dynamically adjusted.
- Network job processing (NJP) allows input and output from compatible JES3 installations. Jobs cannot be transmitted between JES3 and ASP.

## Job Definition

A job in the JES3 system consists of one or more job segments. A job segment (e.g., PRINT or PUNCH) is defined as a logical portion of a job. It is not directly analogous to a job step in MVT, rather it relates the stages of job processing. Except for jobs specifying special JES3 processes (e.g., NJP), job segments are defined by JES3 to control the job's input, control statement conversion, device scheduling, execution, printing and punching, and job purge.

All JES3 support functions are implemented by dynamic support programs (DSPs). A support function is defined as any process performed by the Global

processor as a part of a job in the system. Tasks such as card reading, printing, card punching, and servicing the Local and Global processors are examples of support functions. DSPs can be activated when explicitly requested by the operator or by nonstandard jobs.

### JES3 Job Flow

All jobs are initially read into the Global processor by JES3, and are assigned a unique JES3 job number. Jobs can be submitted via a tape, disk, or card reader attached to the Global processor. In addition, jobs can be submitted to the global processor from remote terminals via remote job processing (RJP), TSO terminals, the internal reader, and from other JES3 systems via network job processing (NJP). The job flow (see figure 20) is as follows:

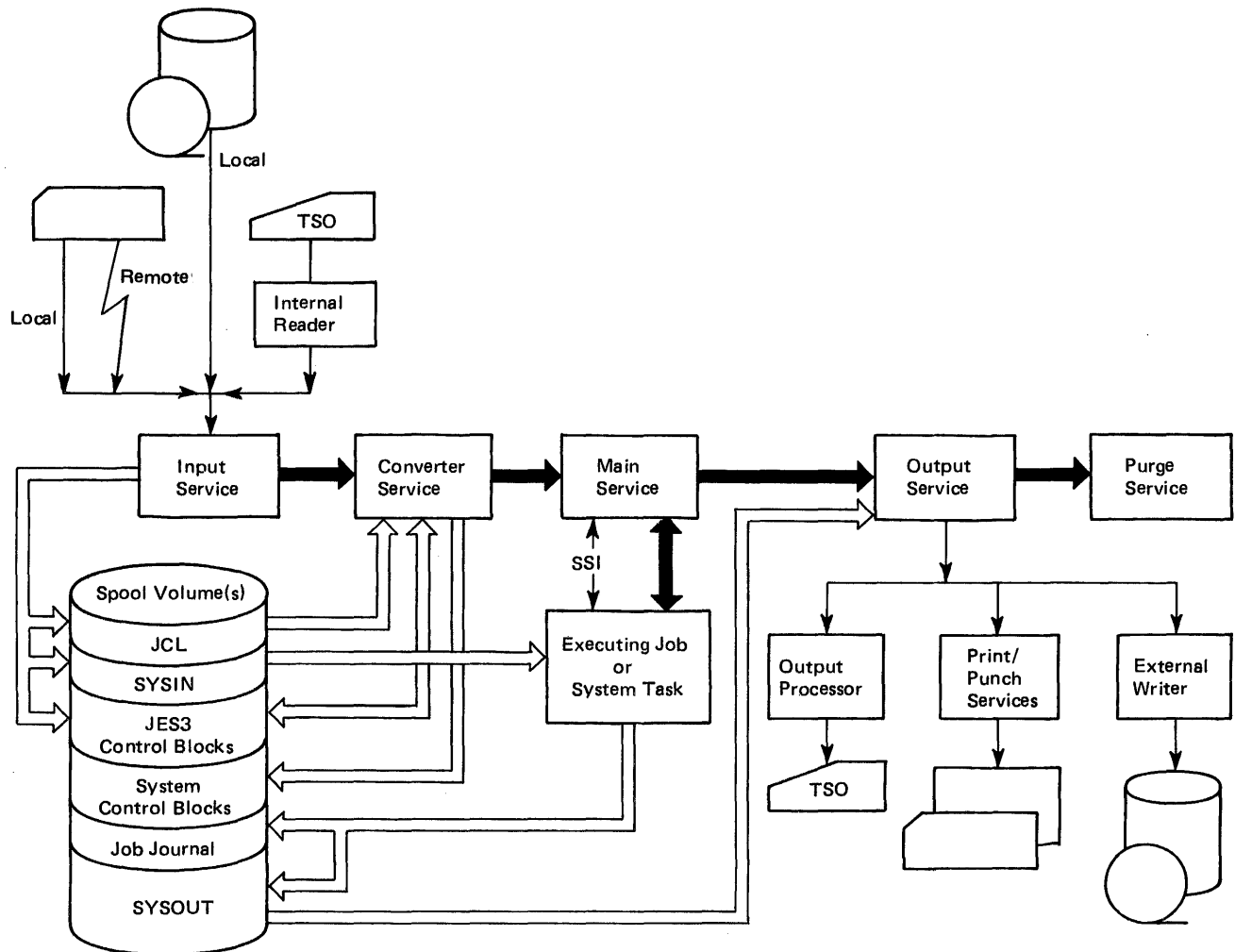


Figure 20. JES3 Job Flow

1. The job is read into the system and is initially placed on a spooling disk pack. An entry for the job is placed in the JES3 job queue. The support function that performs this task is called Input Service.

2. Immediately after input service, converter service is invoked. If a special catalogued procedure library is specified, it is opened and passed to the converter for JCL conversion. Converter service performs two functions:

- Interprets and converts JCL into system control blocks.
- Determines job setup requirements from generated system control blocks.

During the first phase, JCL conversion and interpretation, syntax and logical errors are detected. This early JCL analysis allows cancellation of jobs with system or JES3 JCL errors before the job enters the setup and execution queues.

During the second phase, the system control blocks are scanned and the setup requirements for each job are determined. As in the first phase, jobs with errors are scheduled for output service, bypassing execution.

3. Jobs with no JCL errors are passed to the device scheduling function for preexecution setup. Based on device requirements established by converter service, devices are selected. Any necessary volumes that require mounting are requested from the tape and DASD library console operators (See the section called Device Scheduling).
4. Based on device requirements, initialization specifications, job definition, and user exit, the job will be scheduled for processing on either the Global, a Local, or a Main processor. If the selected processor is Local, the Global processor will pass the job to the Main for processing. JES3 then signals the selected processor to begin executing the job. After processing is complete, control is passed to the Global processor for printing and punching. (Refer also to the section called Job Selection and Scheduling.)
5. Output data sets are queued for the external writer, the TSO OUTPUT command, or JES3 printers or punches.
6. Output data sets are printed by the JES3 print service function. Print service informs the operator of any special printer setup necessary. The printer(s) can be either local or remote as defined by the job.
7. Output data sets are punched by the JES3 punch service function. A local or remote punch may be defined by the job.
8. A support function, purge, releases for use by other jobs, all DASD JES3 queue space associated with the completed job. At this time the SMF purge record is written for the job.

In most cases, the standard job sequence described above will meet the needs of the application programmer; however, through the inclusion of JES3 control statements in the job's JCL, the job function sequence can be altered and special JES3 processing such as NJP can take place. The inclusion of JES3 processing control statements in the JCL causes the job to be classified as nonstandard.

## Device Scheduling

Immediately after the interpretation of the JCL for a job, a table summary of all data set, volume, and device requirements is created. If a JOBCAT or STEPCAT DD statement specifies a special catalog, or if access to the system catalog determines a special catalog is needed, the catalog is dynamically allocated during the building of this table. The table is completed when all JCL cross reference processing is complete and all volume and unit requirements are determined. The job is then enqueued for selection by the device scheduling function.

The device scheduling function selects jobs for preexecution setup by priority, subject to workload parameters as specified by the installation.

JES3 manages all references to tape devices and mountable direct access volumes. Jobs referencing permanently resident volumes will be automatically routed to the Local processor on which the volume resides. Reference to nonspecific direct access volumes are not handled by JES3; they are passed to the system for normal allocation processing.

Jobs that require special data set volumes to be mounted are not submitted for execution until the appropriate units are available for mounting. Instructions are then issued to the operator to mount and/or ready the required volumes.

All volumes are assumed to reside in an installation tape and disk library. Messages are sent to these libraries requesting that all volumes required for preexecution setup by JES3 be forwarded to the tape and disk setup areas. This action is performed for each job requiring setup prior to requesting that the volumes be mounted.

Jobs which do not require setup devices, or for which setup has already been effected, are executed while the setup process is being carried out for other jobs.

JES3 waits for all operators involved to complete their required tasks before allowing the job to execute on a Local processor. If all devices allocated to a job are shared by Local processors, that job will be eligible for execution on any Local processor to which the devices are attached. When setup is complete, the job is released to an execution queue for processing.

In addition, device scheduling provides:

- Setup and system messages for direct access devices and tapes will be routed to the console located nearest each device.
- Multiple jobs on different processors can share direct access volumes and data sets. The data sets can be shared providing the disposition is specified as shared.
- Operator commands are available to determine the status of jobs, devices, and volumes in setup control.
- JCL references to permanently resident direct access volumes require no operator action.
- The number of devices setup can optionally be limited to the number needed at any one time during the job. During deallocation the device will be returned to the device scheduling function for use by other jobs. Devices used for passed data sets can optionally be made available for other allocations.

### **Job Selection and Scheduling**

Each processor in the JES3 complex requests work from the Global processor. Jobs are selected for each processing stage in priority order. Priority is initially established via the priority parameter on the JOB statement or with a JES3 control statement. The selection priority of a job can be changed by the operator, by priority aging, or by deadline scheduling.

The priority aging feature allows JES3 to increase the priority of a job after it has been passed over by the JES3 an installation-specified number of times, either because of an insufficient number of devices or a low priority level relative

to that of other jobs in the system. Raising a job's priority will cause the job to be given preferential treatment in the selection of devices.

The deadline scheduling feature allows the installation to specify a time of day by which the job should be scheduled. If the job is not scheduled by this time, JES3 will increase the priority of the job at installation-defined intervals until it is scheduled.

Certain segments of a job can be designated by a JES3 control statement to be executed in a remote processor. A job can be specified as nonstandard, executed in a remote processor, and its output returned to the submitting processor for printing or punching. The job can be executed in the submitting processor and printed in the remote processor. The user can control NJP processing or allow the operator to control it by making the job eligible for operator-controlled transmission to another processor.

Using dependent job control (DJC), an installation can cause JES3 to control job selection based on dependencies among jobs. With JES3 control cards, the user can specify that one set of jobs (predecessor jobs) be completed ahead of other jobs (successor jobs). Jobs in a DJC network are held after JCL interpretation until predecessor jobs have completed execution or indicated to JES3 that critical processing is complete. This ensures that interjob catalog and device dependencies are resolved before setting up successor jobs. A device pool can be assigned to a set of jobs under DJC, assuring the availability of critical devices when needed.

JES3 job scheduling provides flexibility in balancing the installation workload among the JES3 processors. When a system initiator becomes available for job processing, JES3 selects a job by priority from the available classes, and from among jobs requiring no setup volume mounting, or whose setup volumes have been mounted and verified.

The installation can also specify that JES3 is to attempt to balance the CPU and I/O load by selecting a job whose CPU-to-I/O ratio complements those jobs currently running in the system. A CPU-to-I/O ratio is specified for each job via a JES3 control statement, or is a default by job class. The installation can also specify the number of initiators that can be started on each processor, and the maximum number of jobs from any one class which can be simultaneously scheduled.

## **JES3 Operational Environment**

The operational environment of the loosely coupled JES3 system is considerably different from that of other systems. All JES3 functions for all processors are controlled from consoles attached to the Global processor. The installation can allow system commands to be entered and system messages to be received at these JES3 consoles, reducing or removing the need for an operator to be stationed at system consoles on each separate processor. The implementation of multiple JES3 operator consoles, the separation of JES3 global functions, and the two-way operator communication with JES3 support functions, cause the loosely coupled complex to appear as a single system rather than composed of several separate and independently operated computer systems.

JES3 provides an installation with flexibility in the location of machine-room equipment. By using additional operator consoles, JES3 installations can physically separate the operational functions (card input/output, printing, and tape setup), locating them in areas more convenient to the local work flow. The

card reader/punches and the printers can be located in a job dispatching area where programs are submitted for execution and output is returned. The mountable input/output units can be placed in an area that is convenient to the tape and disk library. In addition, an operator console can be placed at the tape and disk librarian's desk to receive library volume requests. The central processing units can then be placed in some other area that is free of the congestion that often surrounds peripheral units.

### **The Role of the JES3 Operator**

From the Global Processor, the global console operator manages the work flow of the JES3 complex and can modify specific jobs by changing their priorities. Other operators control the devices attached to the Global JES3 system (such as printers, card readers and card punches), mount and demount the Global and Local Processor mountable units, manage the disposition of required special volumes, and monitor the flow of jobs through the system.

Each JES3 support function (processing stage) responds to a number of operator commands that permit the operator to cancel job processing, to restart processing, or to resume processing after an operator intervention request. Some functions, such as Print Service (at the print stage), provide additional options. For each printer, processing can be restarted either at the beginning of the data set or job, or at a checkpoint made within several pages of the current position. The restart can be accomplished on the same printer or on a newly assigned printer. In addition, a request can be made to reload the Universal Character Set buffer of a local printer. Background programs can be invoked from an operator console as well as from the input JCL and JES3 control cards.

The JES3 console inquiry function permits the global console operator to determine the status of the JES3 work queue, the status of a given job in the queue, the amount of space left in the JES3 queue, a summary of the workload backlog, and the status of the workload for any JES3 processing stage. This inquiry feature also permits the operator to determine whether the backlogs are adequate or too high, and to obtain an estimate of when a given job will be processed in the current queue. With this information, and with the ability to change job priorities or to place a job in hold status and later release it for processing, the operator can manage the system. He can also inquire into the status of functional components of the JES3 system, such as the number of buffers currently in use, or invoke a support function which will display the entire system status on a printer attached to the JES3 Global system.

**The Operator's Console:** Operational control is maintained at every stage of processing for the following functions:

- Deletion of a job. The operator can delete a job from the system by issuing the MODIFY with CANCEL message from the console.
- Restart of job processing. The operator can restart applicable job segments on the Global Processor by issuing the RESTART message at a console.
- Change of job priority. The operator can change the priority of a particular job. This option is usually used to expedite job processing for a particular job.
- Holding and releasing of a job in queue. The operator can withhold a job from processing. Further, a job previously in hold status can be released to be scheduled for additional processing. Jobs may be held and released on the basis of priority or, if received from remote workstations, on an individual

terminal basis. Jobs in hold status from a particular terminal can be released entirely, or the operator can specify the number of jobs to be released.

- Initiation of batch support functions. Using the CALL facility, the operator can request the system to schedule support functions (card-to-tape, tape-to-printer, etc.) to be executed concurrently with the standard preprocessing and postprocessing facilities.
- Dynamic reconfiguration of Global processor input/output and terminal devices. The operator can remove, make available, or reroute the output of such devices as printers, card reader/punches, and Local processors.
- System status inquiry. The operator can query the system regarding the job queue for functions such as estimated processor running time and estimated number of lines of print. In addition, the operator can request the status of an individual job, as well as a printout of the entire job queue status.

Messages can be sent between JES3 consoles.

### **Initialization, Control Statements, and Commands**

During initialization, JES3 configuration and processing options are defined in much the same manner as with ASP. This process is controlled by JES3 initialization control cards and allows the user flexibility in modifying JES3 operation.

Initialization consists of:

- Loading the resident portion of the JES3 system.
- Loading optional modules.
- Formatting the JES3 queue devices if necessary.
- Allocating disk space for JES3 data sets.
- Defining job scheduling algorithms.
- Defining installation standards.
- Defining console message routing.
- Defining the number of processors and the operating characteristics of each.
- Establishing support processor device allocation tables.
- Starting of procedures on selected Main processors.

JES3 processes JCL statements in a manner similar to MVT ASP or VS2 Release 1 ASP. Minimal syntax change to existing JCL is required.

JES3 control cards communicate special instructions to the system for forms control on the printer or card punch, the requirements for execution on a specific processor, and special scheduling requirements for job processing.

### **JES3/ASP Compatibility**

Unlike ASP which is an optional extension of OS/360, JES3 is a component of MVS. JES3 functions totally replace SYS1.SYSJOBQE and the OS readers and writers. Figure 21 relates some of the JES3 compatibility items to the systems which support them.



MVT/ASP 3.0	VS2 Release 1/ASP 3.1	VS2 Release 2/JES3
A single system job queue for multiple OS/MVT CPUs.	A single system job queue for intermixed MVT and VS2 multiple CPUs.	A single system job queue for intermixed MVT, VS2-1, VS2-2 multiple CPUs.
All SYSIN/SYSOUT data is sent across the CTC.	All SYSIN/SYSOUT data is sent across the CTC.	JES3 systems share the spool queue. The CTC is used for control information only. MVT and VS2-1 SYSIN/SYSOUT data is sent across the CTC.
SYSOUT blocking is required in the Main processor.	SYSOUT blocking is required in the Main processor.	SYSOUT blocking is transparent to the VS2-2 user. It is required in MVT and VS2-1 processors.
ASP 'hot-jobs' remain active in the event of Support Processor or ASP failure.	ASP 'hot-jobs' remain active in the event of Support Processor or ASP failure.	All jobs running in the failing system are lost but jobs in other systems are not.
The Reader/Interpreter function for all CPUs is executed in the Support Processor.	The Reader/Interpreter function for all CPUs is executed in the Support Processor.	The Converter and Interpreter functions are executed in the Global Processor for VS2-2, and the R/I function executed in the support processor for MVT and VS2-1.
Main storage can be partitioned (fenced) for assignment to specific job groups.	Main storage can be partitioned for MVT processors only. It is not applicable to VS.	Main storage can be partitioned for MVT processors only. It is not applicable to VS.
ASP Dynamic Dispatching.	Available for MVT processors only. VS2 provides a dynamic dispatching function.	Available for MVT processors only. VS2 provides a dynamic dispatching function.
ASP job ENQ occurs at the volume level.	ASP job ENQ occurs at the volume level.	JES3 job ENQ occurs at the data set name level.
Pre and post-SYSGEN modification is necessary.	Pre and post-SYSGEN modification is necessary.	There is a JES3 generation process.
No support.	3270 Light Pen support.	3270 Light Pen and Function Key support.
Operator control via ASP Console Service. The CTC is the System Master Console.	Operator control via ASP Console Service. The CTC is the System Master Console.	Operator control is via JES3 Console Service. Messages and commands are optionally routed to JES3 Console Service via the CTC.
No Checkpoint/Restart support.	No Checkpoint/Restart support.	Full Checkpoint/Restart support.
ASP accounting.	ASP accounting.	ASP accounting is eliminated. Full SMF support is available.
MCS route codes are ignored by ASP.	MCS route codes are ignored by ASP.	MCS route codes can be mapped onto JES3 route codes for VS2-2 processors (only).
POLYASP is supported.	POLYASP is supported.	POLYASP is not supported.

Figure 21. JES3 Differences From MVT ASP and VS2 Release 1 ASP

## **SMF Considerations**

System management facilities (SMF) collect and record system and job information. The information can be used for job accounting and management information reports.

Most of the changes to SMF records and exits for MVS have been made to support new system control program design. Because these changes could affect existing data reduction routines, all records and exits should be studied to determine what effect the changes might have.

The following topics describe the changes that have been made to SMF for MVS.

### ***Differences from VS2 Release 1 System Management Facilities***

The most significant differences from VS2 Release 1 are:

- SMF records contain additional accounting information to record new system environmental characteristics.
- New SMF exits from the control program are provided.
- SMF supports an output limiting facility.
- The SMF data sets, SYS1.MANX and SYS.MANY, must be cataloged.
- The SMFDEFLT list in SYS1.PARMLIB has been changed. Multiple SMF SYS1.PARMLIB members may be maintained. Their names are of the form SMFPRMxx. Some of the parameters in these members are also different. See the topic "IBM-Created Lists" in the chapter "Defining the System" for a more detailed description.
- The type 20 Job Initiation record is now considered a job record and no longer controlled by the DSV option.
- The same SYS1.PARMLIB member is applicable to background and foreground use.

### ***Differences from MVT System Management Facilities***

In addition to the preceding, the most significant differences from MVT are:

- SMF is a standard facility of MVS.
- SMF does not support tape for recording SMF data.

### ***SMF Record Differences for MVS***

Figure 22 indicates how the SMF records have changed for MVS. The following topics explain why some of the more significant changes were made.

Record Type	Differences Between VS2 Release 2 and VS2 Release 1	Differences Between VS2 Release 2 and VS1	Differences Between VS2 Release 2 and MVT	Differences Between VS2 Release 2 and MFT
0	*****	*****	Virtual storage statistics added	Virtual storage statistics added
1	_____	_____	_____	_____
2	*****	*****	*****	*****
3	*****	*****	*****	*****
4	SYSIN/SYSOUT EXCP counts eliminated Virtual storage statistics added VIO statistics added Paging statistics added CPU time statistics separated into two fields	Virtual storage statistics added VIO statistics added Paging statistics added CPU time statistics separated into two fields	SYSIN/SYSOUT EXCP counts eliminated Virtual storage statistics added VIO statistics added Paging statistics added CPU time statistics separated into two fields	SYSIN/SYSOUT EXCP counts eliminated Virtual storage statistics added VIO statistics added Paging statistics added CPU time statistics separated into two fields
5	System resource manager statistics added CPU time statistics separated into two fields	System resources manager statistics added CPU time statistics separated into two fields	System resources manager statistics added CPU time statistics separated into two fields	System resources manager statistics added CPU time statistics separated into two fields
6	Job entry subsystem change	Job entry subsystem change	Job entry subsystem change	Job entry subsystem change
7	*****	*****	*****	*****
8	*****	*****	*****	*****
9	*****	*****	MP statistics eliminated	*****
10	*****	*****	*****	*****
11	*****	*****	MP statistics eliminated	*****
12	_____	_____	_____	_____
13	N/A	_____	N/A	_____
14	VIO statistics added	VIO statistics added	VIO statistics added	VIO statistics added
15	VIO statistics added	VIO statistics added	VIO statistics added	VIO statistics added
17	*****	*****	*****	*****
18	*****	*****	*****	*****
19	*****	*****	*****	*****
20	*****	*****	*****	*****
21	*****	*****	*****	*****
22	New record for MP statistics	New record for MP statistics	New record for MP statistics	New record for MP statistics
26	New record for job entry subsystem statistics	New record for job entry subsystem statistics	New record for job entry subsystem statistics	New record for job entry subsystem statistics
30	_____	N/A	_____	N/A
31	This record is now written when the MODIFY TCAM command with the TS option is issued.	New record for time sharing statistics	This record is now written when the MODIFY TCAM command with the TS option is issued	New record for time sharing statistics
32	_____	N/A	_____	N/A
33	_____	N/A	_____	N/A

Figure 22. Changes to SMF Records for VS2 Release 2 (Part 1 of 2)

Record Type	Differences Between VS2 Release 2 and VS2 Release 1	Differences Between VS2 Release 2 and VS1	Differences Between VS2 Release 2 and MVT	Differences Between VS2 Release 2 and MFT
34	Virtual storage statistics added VIO statistics added Paging statistics added CPU time statistics separated into two fields	New record for time sharing statistics CPU time statistics separated into two fields	Virtual storage statistics added VIO statistics added Paging statistics added CPU time statistics separated into two fields	New record for time sharing statistics CPU time statistics separated into two fields
35	System resources manager statistics added CPU time statistics separated into two fields	New record for time sharing statistics CPU time statistics separated into two fields	System resources manager statistics added CPU time statistics separated into two fields	New record for time sharing statistics CPU time statistics separated into two fields
38	N/A	N/A	————	N/A
40	*****	New record for dynamic allocation statistics	*****	New record for dynamic allocation statistics
41	————	N/A	————	N/A
42	————	N/A	————	N/A
43	New record for job entry subsystem statistics	Job entry subsystem change	New record for job entry subsystem statistics	New record for job entry subsystem statistics
44	N/A	————	N/A	N/A
45	New record for job entry subsystem statistics	New record for job entry subsystem statistics	New record for job entry subsystem statistics	New record for job entry subsystem statistics
47	New record for job entry subsystem statistics	Job entry subsystem change	New record for job entry subsystem statistics	New record for job entry subsystem statistics
48	New record for job entry subsystem statistics	Job entry subsystem change	New record for job entry subsystem statistics	New record for job entry subsystem statistics
49	New record for job entry subsystem statistics	New record for job entry subsystem statistics	New record for job entry subsystem statistics	New record for job entry subsystem statistics
62	*****	*****	New record for VSAM statistics	New record for VSAM statistics
63	*****	*****	New record for VSAM statistics	New record for VSAM statistics
64	*****	*****	New record for VSAM statistics	New record for VSAM statistics
67	*****	*****	New record for VSAM statistics	New record for VSAM statistics
68	*****	*****	New record for VSAM statistics	New record for VSAM statistics
69	*****	*****	New record for VSAM statistics	New record for VSAM statistics
70	New record for MF/1	New record for MF/1	New record for MF/1	New record for MF/1
71	New record for MF/1	New record for MF/1	New record for MF/1	New record for MF/1
72	New record for MF/1	New record for MF/1	New record for MF/1	New record for MF/1
73	New record for MF/1	New record for MF/1	New record for MF/1	New record for MF/1
74	New record for MF/1	New record for MF/1	New record for MF/1	New record for MF/1
<p>———— Record not supported      ***** Same statistics      N/A Not applicable</p>				

Figure 22. Changes to SMF Records for VS2 Release 2 (Part 2 of 2)

## New System Resources Manager Data

SMF record types 4, 5, 34, and 35 (step termination, job termination, time-sharing step termination and LOGOFF) provide new information obtained from system resources management routines. Fields in record types 4 and 34 are now used to record step termination data such as VIO page-ins, VIO page-outs, total page-ins, and total page-outs. Fields in record types 5 and 35 are used to record job termination data such as:

- Total number of service units used by a background job (type 5) or a terminal job (type 35).
- The total amount of time that the background job was active (type 5).
- The total amount of time that all transactions were active (type 35).
- The total number of transactions.

This information can be used to approximate TSO terminal response time for a session, and for adjusting IPS values in an attempt to obtain desired response time. (See the chapter “Directing the Use of System Resources” for a description of an installation performance specification (IPS) and service units.)

## SMF Recording Replaced by MF/1

Two records, type 1 (wait time) and type 12 (end-of-day), are no longer supported because the system activity measurement facility (MF/1) now records the data in place of SMF. The information from record types 1 and 12 is now in record types 70 (CPU activity) and 71 (paging activity), although record types 70 and 71 are formatted differently from record types 1 and 12. Furthermore, MF/1 records are only written to the SMF data set when MF/1 is operational. Therefore, an installation that wants to use the information that was in record types 1 and 12 should rewrite data reduction routines to recognize the new records and process the new format.

MF/1 processing must also be started so that the data will be gathered and recorded.

Three other new records (types 72, 73, and 74) have been created to record MF/1 data. See the topic “Tracing” in the chapter “Directing the Use of System Resources” for a more detailed description of their contents.

## New Storage Data

Three fields in both record types 4 and 34 now provide new step level accounting information such as:

- Either the size of the V=R address area specified on an ADDRSPC=REAL request, or the total size of the user’s private address space for an ADDRSPC=VIRT request.
- The amount of the user’s private address space (assigned from the high address) used for the local system queue area (LSQA), the scheduler work area (SWA), and subpools 229 and 230.
- The amount of the user’s private address space (assigned from the low address) used for the user’s programs.

## Job Entry Subsystem Statistics

Some SMF records have been modified and others created to record statistics provided by the job entry subsystem. They are:

- The type 6 (output) record contains new data such as data set control indicators, JES-assigned job numbers, output route code, and data format error indicators. This record will be produced by the job entry subsystem and external writer which handles all system output processing.
- The type 26 (job summary) record is produced when a job is ready to be purged from the system. It contains information about subsystem processing such as the type of job and how it entered the system, priorities for job selection and output processing, counts of lines or punched cards generated, and times at which the various phases of processing the job began and ended.
- Record types 43, 45, 47, 48, and 49 record information about subsystem activity. Included is data regarding the starting or stopping of a subsystem or a subsystem event (e.g., SIGNON). Record type 49 records the attempt to sign on or start a line with an invalid password.

## Multiprocessing Data

The type 22 record is new in MVS. It indicates the status of resources unique to a multiprocessing configuration. It serves the same function for CPUs, channels, and storage as record types 8, 9, 10, and 11 do for I/O devices.

## Virtual Storage Access Method Data

The virtual storage access method (VSAM) generates new records (types 64, 67, and 68) that contain information similar to existing data management SMF records. VSAM also records statistics when a VSAM data set is opened, allocated, and freed (record types 62, 63, and 69).

**Note:** Record types 63 and 67 are generated for VSAM recovery processing.

## Time Sharing Record Elimination

The following record types are not supported in MVS:

- 30-- Start TS
- 32-- Driver
- 33-- Driver Modify
- 41-- Modify TS
- 42-- Stop TS

These records were associated with starting and stopping TSO and with the TSO driver. These functions have been replaced by the MODIFY TCAM command and the system resources manager in MVS.

## *Changes to SMF Exits for MVS*

Figure 23 indicates the changes that have been made to SMF exits in other systems for MVS.

Exit	Difference Between VS2 Release 2 and VS2 Release 1	Difference Between VS2 Release 2 and VS1	Difference Between VS2 Release 2 and MVT	Difference Between VS2 Release 2 and MFT
IEFUJV	This exit is also taken after the internal text is created	This exit is also taken after the internal text is created	This exit is also taken after the internal text is created	This exit is also taken after the internal text is created
IEFUIV	N/A	————	N/A	N/A
IEFUJI	*****	*****	*****	*****
IEFUSI	*****	*****	*****	*****
IEFUSO	New exit for output limiting facility	*****	*****	*****
IEFACTRT	Counts are in terms of logical records rather than physical blocks	Counts are in terms of logical records rather than physical blocks	Counts are in terms of logical records rather than physical blocks	Counts are in terms of logical records rather than physical blocks
IEFUTL	This exit can now be taken from the foreground  There is a new interface to extend the time limit in micro seconds  For wait limits there is no longer an extension. A new time limit is set for the remainder of the step.	This exit can now be taken from the foreground  There is a new interface to extend the time limit in micro seconds  For wait limits there is no longer an extension. A new time limit is set for the remainder of the step.	This exit can now be taken from the foreground  There is a new interface to extend the time limit in micro seconds  For wait limits there is no longer an extension. A new time limit is set for the remainder of the step.	This exit can now be taken from the foreground  There is a new interface to extend the time limit in micro seconds  For wait limits there is no longer an extension. A new time limit is set for the remainder of the step.
IEFU83	*****	New exit for selecting records to be written	New exit for selecting records to be written	New exit for selecting records to be written
IEFUJP	New exit receives control when job is ready to be purged from the system	Change in exit parameters	New exit receives control when job is ready to be purged from the system	New exit receives control when job is ready to be purged from the system
<p>———— Exit not supported  ***** Unchanged  N/A Not applicable</p>				

Figure 23. Changes to SMF Exits for VS2 Release 2

## **Job Control Language**

Job control language statements contain information that is needed to control the processing of jobs.

This section describes JCL facilities, parameters, and subparameters that are either new or have changed from both VS2 Release 1 to MVS and MVT to MVS. Figure 24 summarizes the more significant JCL changes.



Statement	Parameter	Change
JOB	accounting information field	● In addition to standard OS usage, this field can be used for JES2 parameters.
	ADDRSPC	● Assigns pageable or nonpageable storage space, and defaults for pageable storage.
	CLASS	● (A-Z, 0-9). The job entry subsystem enqueues the job on the associated class queue.
	PERFORM	● Associates jobs with performance group definitions.
	PRTY	● This parameter is ignored when the JES2 job entry subsystem is used. Priority values can be specified on a subsystem control card.
	REGION	● Limits the size of variable GETMAIN requests.
	ROLL	● This parameter is ignored.
	TYPRUN	● If TYPRUN=SCAN, the JCL is converted but does not execute.
EXEC	ADDRSPC	● Assigns pageable or nonpageable storage space, and defaults for pageable storage.
	DPRTY	● The default is the value of the APG priority.
	DYNAMNBR	● Limits the number of data sets that a job step may hold for reuse.
	PERFORM	● Associates job steps with performance group definitions.
	REGION	● Limits the size of variable GETMAIN requests.
	ROLL	● This parameter is ignored; the job is not failed.
DD	AFF	● This parameter is ignored; the job is not failed.
	AMP	● Completes information in an access method control block (ACB) for the virtual storage access method (VSAM) data sets.
	COPIES	● Specifies multiple copies of an output data set.
	DCB	● DEN=4 subparameter indicates 6250 bits-per-inch for 9-track tape.
		● FRID subparameter specifies the SYS1.IMAGELIB member to be used to interpret documents read by a 3886 OCR device.
		● HIARCHY subparameter is ignored; the job is not failed.
	DEST	● Allows you to route output to specified destinations.
	DYNAM	● Total is added to DYNAMNBR value, if any.
	FREE	● Indicates when a data set is to be deallocated.
	JOBCAT	● Defines a private user catalog.
	OUTLIM	● With SYSOUT=, this parameter will specify the maximum number of logical records (for example, print lines) that may be written into this SYSOUT data set. An SMF exit routine is entered when this number is exceeded.
	RETAIN	● This parameter is ignored; the job is not failed.
	SEP	● This parameter is ignored; the job is not failed.
	SPACE	● Has a new default value for VIO data sets.
	SPLIT	● Converted to an equivalent SPACE parameter.
	STEPCAT	● Defines a private VSAM user catalog.
	SUBALLOC	● Converted to an equivalent SPACE parameter.
	SYSOUT=(c,p,f)	● c – (A-Z, 0-9) (* is valid in JES2 only.) This output class designation is used to queue the associated output. Data is dequeued in much the same manner as the MVT output writer. p – (p=INTRDR). This data set is sent directly to input service for processing as an input job stream. p – (p ≠ INTRDR). This data set is enqueued separately for the named writer. f – Data is enqueued separately with this alphanumeric forms designation.
UNIT	● The SEP subparameter is ignored, the job is not failed.	
VOLUME	● The RETAIN subparameter is ignored; the job is not failed.	
<ul style="list-style-type: none"> <li>● New or changed from MVT</li> <li>○ New or changed from VS2 Release 1</li> <li>● New or changed from MVT and VS2 Release 1</li> </ul>		

Figure 24. JCL Changes for VS2 Release 2

## **Changes to VS2 Release 1 JCL**

The following topics describe the differences from VS2 Release 1 JCL.

### **Performance Group Specification**

The PERFORM parameter is new for MVS. It is used to associate a job or job step with any of the several performance groups specified by installation-supplied performance group definitions. Each performance group definition specifies a rate for providing system resources to terminal jobs or batch jobs or job steps under varying conditions (see the chapter: "System Resources Management").

- When PERFORM=groupnr is coded on a JOB statement, that job receives system resources at the rate defined by the performance group identified by 'groupnr' (a number between 1 and 255).
- When PERFORM=groupnr is coded on an EXEC statement, the group number value applies only to the job step specified on the EXEC statement.
- When PERFORM.procstep=groupnr is coded on an EXEC statement, the group number value applies to the specified step within the procedure invoked by the EXEC statement.

If the PERFORM parameter is coded on the JOB statement, its value overrides any value for PERFORM parameters that may be specified on the EXEC statements for that job. If a PERFORM parameter is specified for a procedure, its value overrides any value for PERFORM parameters that may be specified on EXEC statements within the procedure.

If the PERFORM parameter is not coded, the default is 1 for background jobs and 2 for foreground users. Therefore if the previous job control language for a job is not changed, the job will be associated with one of only two possible performance group definitions.

### **Assigning Job Selection Priority**

The PRTY parameter of the JOB statement is ignored in MVS with the JES2 job entry subsystem. Instead, job selection priority must be specified on a JES2 control card.

The PRTY parameter is valid with JES3.

### **Specifying Data Sets in Paging Space (Virtual I/O Data Sets)**

A new Stage I system generation parameter, VIO, may be specified on the UNITNAME macro instruction to identify unit names that may be used to specify data sets that are to receive virtual I/O (VIO) processing. Any single volume temporary data set defined by a DD statement with a unit name for which VIO=YES was coded at system generation time, is processed as a virtual I/O data set.

The DD statement that defines the temporary data set must meet the following requirements if the temporary data set is to be allocated to the system's paging space:

- The DSNAME parameter must not be specified unless an & name value is supplied for it. (Data sets with non-system generated names are not allocated to paging space).
- The data set disposition must be new or passed.

- The UNIT parameter must specify a unit name that indicates system paging space. (VIO=YES must have been specified for that unit name at system generation time.)
- Indexed sequential data set organization must not be specified on the DSORG subparameter of the DCB parameter.
- No volume serial number may be specified.

If the SPACE parameter is not coded for virtual I/O data sets, the default value is 10 primary and 50 secondary blocks with an average block length of 1000.

### **Requesting Data Set Freeing at Data Set Close Time**

The FREE parameter of the DD statement is new in MVS. It is used to specify that a data set is to be freed when that data set is closed.

If FREE=END is specified, the data set is freed at the end of the job step. If FREE=CLOSE is specified, the data set is freed when the CLOSE macro instruction is issued for it.

If the FREE parameter is not coded, the default value is END. Therefore, if the current job control language for a job is not changed, all data sets will be released at the end of the job step.

If FREE=CLOSE is specified for a SYSOUT data set, the data set will be sent to the job entry subsystem when the CLOSE macro instruction is issued. The data set is eligible for immediate printout (spinoff).

**Note:** FREE=CLOSE should not be specified for a data set that is opened and closed more than once during a job step.

### **Specifying the Number of Resources That May Be Held for Reuse**

The DYNAMNBR parameter of the EXEC statement is new in MVS. While it extends (dynamic allocation) and replaces the function of the DD DYNAM statement in MVT and VS2 Release 1, it does not specify a value that limits the number of dynamic allocations allowed to a terminal job. Rather, it specifies a value that is used to control the number of not-in-use data sets that may be allocated to a job step. The DYNAMNBR parameter value is added to the number of DD statements in the job step to determine the number of data sets with the not-in-use attribute that a job step may hold.

The value of the DYNAMNBR parameter must be within the range of 0 to 1,635. If the DYNAMNBR parameter is not coded, the default value is 0.

### **Requesting Multiple Copies of an Output Data Set**

The COPIES parameter of the DD statement is new in MVS. It specifies the number of copies of an output data set that are to be produced by the job entry subsystem. The COPIES parameter may be coded only when the SYSOUT parameter is included on the DD statement.

### **Directing the Destination of an Output Data Set**

The DEST parameter of the DD statement is new in MVS. It specifies that an output data set is to be routed to a remote station or local device by the job entry subsystem. The DEST parameter may be coded only when the SYSOUT parameter is included on the DD statement.

## Support for Tape and Direct Access Devices

If INTRDR is not coded for the SYSOUT parameter of the DD statement, the data set is enqueued separately for the named writer. This writer is not invoked by the job entry subsystem -- the operator must start an external writer to attach any user-written writer. The main purpose of the external writer is to support output devices not supported by the job entry subsystem, specifically tape and direct access devices. The external writer uses the QSAM access method and any devices supported by QSAM.

To separate output destined for an external writer from output processed by the job entry subsystem, it is advisable to place it in separate classes not assigned to job entry subsystem devices.

## Format Record Identification for the 3886 OCR Device

A new subparameter (FRID) of the DCB parameter of the DD statement is used to specify the type of document that is supplied as input to a 3886 optical character recognition (OCR) device. The FRID subparameter indicates the member of the SYS1.IMAGELIB data set that is to be used for interpreting the document to be read by the 3886 OCR device.

## Output Limit Specification

The OUTLIM parameter on the DD statement is used to specify the limit for the number of logical records to be written to an output data set. It is coded the same way as the MVT OUTLIM parameter.

## Increased Data Set Definition Capability

In MVS, as many as 1,635 DD statements may be specified for each job step. In both MVT and VS2 Release 1, the limit was 255 DD statements per step.

## Increased Volume Sharing Capability

The following additional types of volumes may be shared between jobs and job steps in MVS:

- Private direct access volumes mounted in response to nonspecific requests.
- Direct access volumes for which deferred mounting has been requested.

## Interpretation of the REGION Parameter

In MVS, the address space available to an individual job (user private address space) is restricted only by the size of the individual virtual address space (16 megabytes) minus the space required for system control program use. Therefore, for jobs that specify ADDRSPC=VIRT (requesting V=V storage), the REGION parameter is not used to specify the size of the address space available to the individual job. It is, however, used to specify the maximum value for use by the GETMAIN service routines in servicing variable-length GETMAIN requests. When a variable-length GETMAIN is issued that would cause the address space allocated to a user to exceed the value of the REGION parameter, the user is limited to the REGION parameter value. For jobs that specify ADDRSPC=REAL (requesting V=R storage), the value specified is used to limit the size of the address space available to the individual job.

### **Job Step Dispatching Priority Default Value**

In MVS, the value of the APG priority will be used as the default dispatching priority for a step when the DPRTY parameter is not specified on the EXEC statement. In MVT and VS2 Release 1, the default dispatching priority for the step was the job's dispatching priority.

### **Changes to MVT JCL**

In addition to the differences described above, two JCL parameters are new to current MVT users.

### **Requesting Storage for Program Execution**

The ADDRSPC parameter is new in VS. It is used to assign pageable or nonpageable storage space to a job or a job step. It may be specified as either ADDRSPC=VIRT or ADDRSPC=REAL:

- If ADDRSPC=VIRT is specified, the job is assigned to pageable (V=V) storage.
- If ADDRSPC=REAL is specified, the job is assigned to nonpageable (V=R) storage.
- If the ADDRSPC parameter is not coded, the default value is VIRT. Therefore, if the current job control language for a job is not changed, the job will be executed in pageable storage. If the job must be executed in nonpageable storage, ADDRSPC=REAL must be specified.

If the ADDRSPC parameter is coded on the JOB statement, its value overrides any value in the ADDRSPC parameters that may be specified in the EXEC statements for that job.

### **Specifying the BTAM Online Terminal Test Option**

In MVS, a new value can be specified for the EROPT subparameter of the DCB parameter of the DD statement. If EROPT=T is coded, it indicates a request for the basic telecommunication access method (BTAM) online terminal test option.

### **JES2 JCL Considerations**

The JES2 job entry subsystem processes JCL statements in a manner similar to MVT/HASP and VS2 Release 1/HASP. The JCL considerations for JES2 therefore fall into two categories, depending upon whether or not an installation currently has a HASP system. In either case, however, no syntax changes to existing JCL will be required.

### **JES2 JCL for Non-HASP Users**

Two significant points for installations that are unfamiliar with HASP are:

- Extended JCL facilities are provided by JES2 control cards.
- The cataloged procedure to be used for job conversion can no longer be associated with an input stream but rather with a job class.

**JES2 Control Cards:** The JES2 control cards are identified by a /\* in columns 1 and 2 and a JES2 control verb beginning in column 3. They are used to:

- Supply information for the JES2 separator page and card.

- Identify a job selection priority (for queuing the job).
- Supply information for calculating a job selection priority if such a priority is not explicitly stated.
- Identify volumes required by the job, so that the job is held until the required volumes become available.
- Route system output data sets to the local printer or punch.
- Send a message to the central computer operator.
- Specify the DDNAME of the cataloged procedure library to be used for conversion of the job's JCL.
- Specify extended output processing setup options to be associated with data set groups.

**Assigning Job Selection Priority:** A JES2 control card (priority) is used to assign a selection priority to a job. If this control card is not included, the job entry subsystem assigns a priority determined either by the estimated execution time and estimated number of lines of output (values contained on another JES2 control card) or by the JES2 generation default values.

If the PRTY parameter is coded on a JOB statement, it will be ignored.

## JES2 JCL for HASP Users

The two most significant points are:

- In addition to the job copies and job routing facilities, it is now possible to specify data set copies and data set routing.
- Two new control cards have been created.

**/\*JOBPARM Control Card:** The /\*JOBPARM control card was new in HASP for VS2 Release 1 and is coded similarly for JES2. It specifies job-related parameters that are used by the JES2 job entry subsystem to:

- Establish the default job selection priority.
- Control the job's processing.
- Build the JES2 separator page and card.

These parameters extend the information that previously could be specified only on a JOB statement accounting information parameter by adding information such as the name of the cataloged procedure library to be used for JCL conversion.

**/\*OUTPUT Control Card:** The /\*OUTPUT control card was new in HASP for VS2 Release 1, and is coded similarly for JES2. It specifies characteristics and/or options for groups of output data sets.

It is used in conjunction with the form number subparameter in the SYSOUT parameter of the DD statement.

Any system output set whose DD statement form subparameter matches the FORMS code specified on as /\*OUTPUT control card, will be processed according to the specifications of the /\*OUTPUT control card.

## JES3 JCL Considerations

JES3 control cards submitted with the JCL for the job, permit the programmer to define job processing. Flexibility can be gained by use of control cards; installation standards can also be used for determining job processing. Control cards provide:

- Special instructions for forms, trains, and carriage tape control on the printer, or forms control on the punch.
- Specification of volumes that are to be mounted prior to execution.
- Special functions that are required for job processing.
- Specification of processor dependencies.

Errors in JES3 control cards are noted in the SYSMSG data set. The job is cancelled, and the SYSMSG data set printed. JES3 supports a continuation character in column 72 of the control card. Figure 25 summarizes JES3 features available to the application programmer.

Feature	JES3 Control Card
Print multiple copies	FORMAT
Punch multiple copies	FORMAT
Suppress printing of DATASET	FORMAT
Suppress punching of DATASET	FORMAT
Submit data mode 2 data	DATASET
Specify single-space print	FORMAT
Route a data set to a remote location	FORMAT
Specify special forms	FORMAT
Specify printer overflow off	FORMAT
Specify special print train	FORMAT
Specify special carriage tape or FCB load	FORMAT
Route TSO output to MVT or VS2/Release 1 TSO terminal	FORMAT
Select a processor for execution	MAIN
Specify a job class of 2 or more characters	MAIN
Specify job execution limits	MAIN
What to do if processor fails	MAIN
Submit a job from another location	MAIN
Specify DEADLINE for execution	MAIN
Build a DJC job interdependency	NET
Write a message to the operator	OPERATOR
List a deck of cards	PROCESS/FORMAT
Copy a deck of cards	PROCESS/FORMAT

Figure 25. Features Supported With JES3 Control Cards

JES3 control cards are identical to ASP Version 3 control cards with the following exceptions:

<b>FORMAT AC</b>	This parameter is supported only for data sets destined for TSO users on MVT or VS2 Release 1 processors.
<b>MAIN ACHOLD and ACMAIN</b>	These parameters are supported only for jobs submitted from MVT or from VS2 Release 1.
<b>MAIN HOTJOB</b>	This parameter is supported only for jobs executed on MVT and VS2 Release 1 processors.  A new parameter is supported which allows the job to specify the cataloged procedure libraries to be used to convert the JCL for the job.
<b>DATASET</b>	This control card is supported only for jobs run on MVT and VS2 Release 1 processors.

## Communication Between Address Spaces

In MVS, communication between address spaces can be facilitated by eliminating some of the actual movement of data. The multiple address space concept of Release 2 allows access to a common service area (CSA), rather than making it necessary to move data into a specific address space.

TCAM, for example, has been modified to utilize this design. Figure 26 describes TCAM under Release 1 versus TCAM under MVS. Under Release 1, data is placed in the TCAM buffer and then moved into the address space for a TSO user. Under Release 2, the TSO user is given the address of data that is placed directly into the CSA. It is not necessary for the data to be moved.



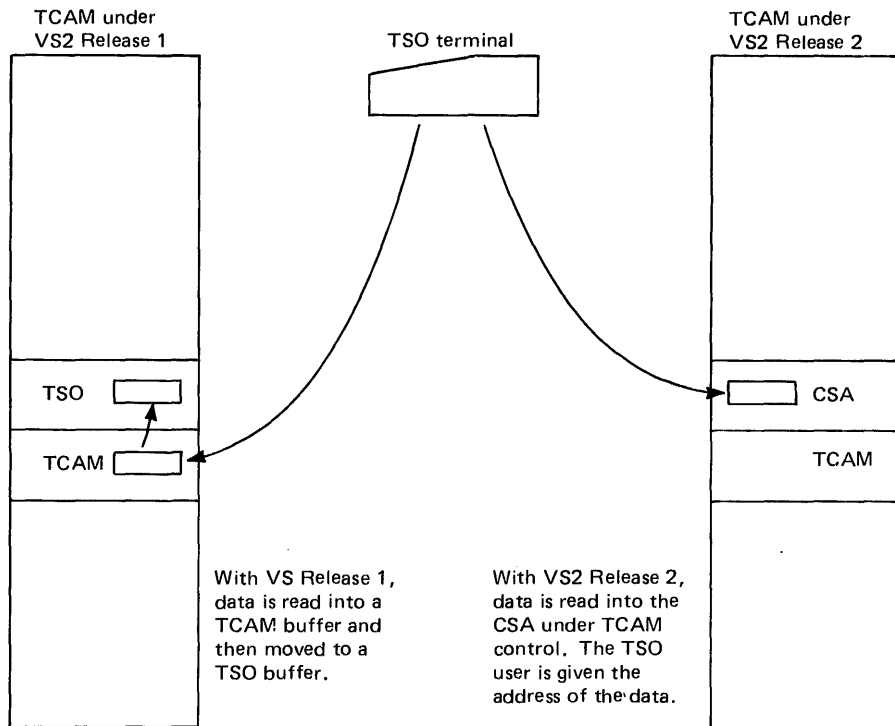


Figure 26. Communication Between Address Spaces

## Allocation Considerations

The function of device allocation has been redesigned with the intention of reducing serialization. The installation can take advantage of the new design by defining mount characteristics, a precedence list for devices, group names for subsets of devices, and preferring or avoiding certain types of allocation.

Some concepts necessary to the discussion of allocation philosophy are:

### Allocation Serialization

Serialization is defined as one or more allocations waiting for another allocation to free devices that are necessary to their process.

### Mount Characteristics

Allocation is first attempted to permanently resident or reserved direct access volumes. The reserved mount characteristic can be assigned to a volume by issuing the MOUNT command or by specifying the characteristic in a volume attribute list of SYS1.PARMLIB.

### User-Assigned Names

Allocation to a generic device name (e.g., 3330, 2314) serializes that entire set of devices. Similarly, allocation to a specific device in that set serializes the entire generic set. MVS provides the capability to give a name to a subset of devices, serializing only the devices in that subset. The UNITNAME macro instruction issued during system generation is the vehicle for naming devices in this manner.

An installation with 3330 Direct Access Storage Facilities at unit addresses 330-337 can use the UNITNAME macro as follows:

```
UNITNAME    UNIT=( 330,4 ),    NAME=SYSBATCH
UNITNAME    UNIT=( 334,4 ),    NAME=SYSTSO
```

With the MVS allocation design, an allocation to SYSBATCH will serialize only devices 330-333, instead of the entire 3330 generic set (See figure 27).

Generic Device Name	3330							
Group Name	SYSBATCH				SYSTSO			
Unit	330	331	332	333	334	335	336	337

Figure 27. Relationship Between User-Assigned and Generic Names

### Device Precedence List

The DEVDPREF operand of the SCHEDULR macro can be used by an installation to specify (by generic name) the order in which types of devices are to be preferred for allocation within a job step, or for dynamic allocation. (There is also a system default.)

An installation can set up the list with the fastest device types preferred, and allow for the fact that generics with more devices (or channels) give the system resources manager a wider choice of devices with which to meet the installation-specified objectives. Another possibility is to prefer device types with the largest available space.

### MVS Allocation Philosophy

The MVS design is such that allocation is attempted in four phases. When allocation is completed in any one phase, it is not necessary to enter the next phase -- and each phase takes additional time. Phase 4 is particularly time consuming. The following overview of the allocation process is intended to allow an installation to decide its allocation philosophy. Figure 28 shows serialization characteristics for the types of allocation requests described in the first two phases. The phases are:

1. Requests are processed that do not require units and volumes to be allocated. This includes dummy data sets, VIO data sets, and subsystem (SYSIN, SYSOUT) data sets.
2. Data sets are allocated to inherently-shareable units such as direct access units that contain permanently resident or reserved volumes. Teleprocessing devices are then allocated.
3. Data sets are allocated by generic device type. During this phase user-assigned names and the device precedence list can be used to best advantage.
4. The fourth phase is recovery allocation. Offline devices are considered, as well as devices allocated to other jobs. A decision to wait for a resource while holding or not holding other resources may be necessary. The mounting of SCRATCH volumes also occurs in this phase.

Allocation	Serialized With Other Allocations	Serialized With Other Components
Dummy data sets	No	No
VIO data sets	No	No
Subsystem data sets	No	No
Requests for permanently resident or reserved volumes and teleprocessing lines	No	OLTEP, DASDI, DDR, UNLOAD Reconfiguration, job deallocation

Figure 28. Serialization Characteristics of Suggested Allocation Methods

### ***Reducing Allocation Contention Among Job Steps***

To reduce allocation contention, the following characteristics of MVS allocation are presented for consideration by the installation:

- Requests for VIO data sets cause no serialization. They are temporary (new, delete) and do not specify a volume serial number.
- As many direct access devices as possible should be permanently resident or reserved. Allocation to these volumes causes no serialization with other allocations.
- Names should be defined by the user within major generic types (such as 3330). Subsets of users then request separate names, which can allow parallel allocation within the generic set.
- Requests for specific unit addresses (e.g., UNIT=283) serialize the entire generic device type. Requests of this nature do not make best use of the allocation design to reduce allocation contention.
- The device precedence list is of value primarily for new data sets allocating to user-assigned names which span device types (e.g., SYSSQ -- sequential devices; SYSDA -- direct access devices).
- The device precedence list can be set up to prefer:
  - speed 2305-2, 3330
  - space 3330, 2314, 2305-2
  - density 2400-3, 2400-4, 2400
- Private volumes (including private catalogs) should be mounted before jobs requiring them are run.

## **Operator Command Language**

Operator commands are statements to the control program issued via a console device or in an input stream. They are used to request information from the control program, alter normal operations, initiate new operations, or terminate existing operations.

In MVT all data coming into and out of the system was handled by system routines and, consequently, controlled by system operator commands. With the introduction of a job entry subsystem, the handling of the reading, writing, spooling, and queueing of data is a function of the job entry subsystem in control. All data flow prior to and immediately after execution is handled by a job entry subsystem and controlled by its unique operator commands. Hence in MVS, some of the historical system commands and the functions they performed will be replaced by operator commands of the job entry subsystem.

This section does not describe all of the operator commands needed to run jobs in MVS. Rather, it describes the manner in which the operator command language has been changed for this release. Also included is a brief functional description of those commands related to JES2, the job entry subsystem that is shipped with the initial version of Release 2.

Other areas of support include multiprocessing, VTAM, time sharing, and resource management. All have caused changes to the command language. Support for miscellaneous functions is described at the end of the section.

### ***Changes to Operator Commands***

Figure 29 contains operator commands and operands that have changed for MVS. Functions that are no longer supported by operator commands are usually controlled through the job entry subsystem. The specific job entry subsystem commands are not listed, but their general functional description is included under "JES Support". Note that a command may have changes supporting several separate functions.

Command	Operand		Change
CANCEL	jobname	⊙	Used to cancel jobs in execution (only).
	u = userid	⊙	Cancels time sharing terminal session.
	IN OUT ALL BRDR	⊙	No longer supported. The job entry subsystem provides functionally equivalent support.
CHNGDUMP	SET	⊙	Overrides the SYS1,PARMLIB defaults for SYSABEND or SYSUDUMP storage dumps.
	DEL	⊙	Eliminates the SET operand overrides.
CONTROL	M	⊙	Replaced by the T operand.
	T	⊙	Supports the TRACK command, which has replaced MONITOR A.
DISPLAY	JOBS [,LIST]	⊙	Active batch-type jobs, and started tasks and jobs.
	TS [,LIST]	⊙	Active time-sharing jobs. Functionally equivalent to DISPLAY USER.
	A [,LIST]	⊙	All the above.
	jobname	⊙	No longer supported. The job entry subsystem provides functionally equivalent support.
	N	⊙	
	Q	⊙	
	SQA	⊙	Replaced by DISPLAY TS.
	USER	⊙	
R [,LIST]	⊙	The optional LIST operand shows the first 67 characters of outstanding reply messages.	
M	○	Used with multiprocessing to selectively display the status of such system resources as CPUs, channels, devices, and storage. Generally compatible with M65 MP.	
HALT	EOD	●	Initiates closedown of entire operating system.
	TP	●	Initiates closedown of TCAM.
	NET	⊙	Initiates closedown of VTAM.
HOLD	Q	⊙	No longer supported. The job entry subsystem provides functionally equivalent support.
	Q=LIST	⊙	
	jobname	⊙	
	TP=stationname		Used to intercept a TCAM station.
MODIFY	BRDRQ	⊙	No longer supported. The job entry subsystem provides functionally equivalent support. The PAUSE and CLASS parameters are still supported via external writer.
	CLASS	⊙	
	JOBCLASS	⊙	
	OUTCLASS	⊙	
	PAUSE	⊙	

Figure 29. Changes to Operator Commands for VS2 Release 2 (Part 1 of 3)

Command	Operand		Change
MODIFY (Continued)	DRIVER	⊙	No longer supported because time sharing has been integrated into the system.
	HOLD	⊙	
	REGSIZE	⊙	
	SMF	⊙	
	SUBMIT	⊙	
	USERS	⊙	
	NET	⊙	Optionally changes some of the operating characteristics of VTAM.
MONITOR	A	⊙	Functionally replaced by the TRACK command.
MOUNT	LSQA=	⊙	This keyword can no longer be included. LSQA is now dynamically expandable during execution. Including this keyword will cause the command to be rejected.
MSGRT	MN=A	⊙	This operand is no longer supported. MSGRT establishes default routing for TRACK, STOPTR, DISPLAY and CONTROL commands.
	TR=A		This operand functionally replaces MN=A.
QUIESCE		⊙	The short form of the command (Q) is not supported. It is not necessarily issued before every reconfiguration, as required by M65 MP.
RELEASE	Q	⊙	No longer supported. The job entry subsystem provides functionally equivalent support.
	Q=LIST	⊙	
	jobname	⊙	
	TP=stationname		Used to release an intercepted TCAM station.
RESET	jobname, PERFORM=	⊙	Changes the performance group for the jobstep currently running.
	CLASS	⊙	No longer supported. The job entry subsystem provides functionally equivalent support.
	OUT	⊙	
	PRTY	⊙	
SEND	SAVE	⊙	The message is not sent until the next time the userid is logged on -- even if it is currently logged on.
	OPER	⊙	Specifies that the message text is to be sent to a specific console.
	BRDCAST	⊙	Specifies that the message is to be sent to all active consoles.
	CN	⊙	Specifies that the message is to be sent to a specific operator's console. Messages sent via the OPER or CN keywords will have respectively OPER or CN appended to the message.
SET	IPS=nn	⊙	This optional parameter can specify a member of SYS1.PARMLIB which contains the system resources manager parameters under which the system is to run.
	Q		No longer supported. The job entry subsystem provides functionally equivalent support.
	PROC AUTO	⊙	
START	LSQA=	⊙	This keyword is no longer supported since LSQA is dynamically expandable during execution. If this keyword is included, the command will be rejected.
	NET	⊙	Initiates startup of VTAM.

Figure 29. Changes to Operator Commands for VS2 Release 2 (Part 2 of 3)

Command	Operand		Change
STOPTR		⊙	Stops the periodic displays activated through the TRACK command.
TRACK	JOBS or J		Requests periodic displays of the number and names of jobs and users.
	TS	⊙	
	A		
		⊙	As opposed to MONITOR A, each graphics console is allowed a TRACK command. The TRACK command is rejected from a nonCRT console or TSO terminal.
VARY	CPU	⊙	Changes the status of a CPU to ONLINE or OFFLINE. Generally compatible with M65MP.
	CHANNEL	⊙	Generally compatible with M65MP. The channel range is now 0 to F. A channel with an online unallocated tape drive can be taken offline by VARY.
	PATH	○	Causes a path (logical connection) to be logically made available or removed from the system.
	STORAGE	⊙	Changes the status of an area of real storage to ONLINE or OFFLINE. M65MP Box number designations are not allowed. Address ranges may be entered in multiples of K. (K = 1024)
	NET	⊙	Used to add or remove VTAM terminals.
WRITELOG	START	⊙	Reinitiates the log function.
<ul style="list-style-type: none"> <li>● New or changed from MVT</li> <li>○ New or changed from VS2 Release 1</li> <li>⊙ New or changed from MVT and VS2 Release 1</li> </ul>			

Figure 29. Changes to Operator Commands for VS2 Release 2 (Part 3 of 3)

### Multiprocessing Support

The MVT VARY commands (VARY STORAGE, VARY PATH, VARY CPU, VARY CHANNEL) are supported in MVS. There is no MP support in VS2 Release 1.

The QUIESCE command suspends system activity, i.e., all CPUs are placed in manual state.

The DISPLAY MATRIX command is generally compatible with M65MP. The syntax has been changed for compatibility with TSO command language, and a new operand displays the highest real address. Two CPUs (0 and 1) and 16 channels (0-F) are supported.

### Diagnostic Support

The system trace function is now supported with a TRACE command, as well as the generalized trace facility (GTF). If TRACE is on, entries to the trace table are automatically discontinued when GTF is started and begun again when GTF is stopped. GTF is started and stopped from the master console.

### VTAM Support

The NET operand on the DISPLAY, HALT, START, and VARY commands identify them with VTAM.

### Time Sharing (TSO) Support

Since time sharing is now integrated into the system, MODIFY command operands such as USERS, SUBMIT, REGSIZE, DRIVER, SMF, and HOLD are not supported. Time sharing is started and stopped by the operator issuing a MODIFY TCAM with a TS=START or STOP operand. There is no TSO driver to modify. Its functions are supported by the system resources manager.

The background reader (BRDR) support by the START, STOP, and CANCEL commands is dropped since there is no longer a background reader. Reader functions are supported by the job entry subsystem.

### System Resources Management Support

START TS and the driver parameters of MODIFY are no longer supported. The functions of the driver are performed by the system resources manager.

The IPS parameter of the SET command specifies the member of SYS1.PARMLIB which contains the parameters which are used by the system resources manager.

### Miscellaneous Support

General support for MVS is provided with several new functions or commands.

**Displays:** The DISPLAY, TRACK and STOPTR commands are used to request or change the display of information concerning currently active tasks. Only jobs running under initiators and started tasks are displayed.

Each graphics console is allowed a TRACK command, which functionally replaces the MONITOR A command. TRACK will be rejected from a console with no CRT. The TRACK and STOPTR commands are used for timer or interval driven periodic displays, as distinguished from the event driven displays of the MONITOR and STOPMN commands.

DISPLAY USER, N, Q, SQA, and jobname, are not supported. These functions are supported through the job entry subsystem.

The M operand of the CONTROL command is replaced by the T operand -an equivalent function. CONTROL T affects the TRACK command instead of the MONITOR A command.

**Dumps:** Temporary changes to the defaults defined in SYS1.PARMLIB may be specified or eliminated with the CHNGDUMP command -new in this release.

**Messages:** The SEND command is modified to allow the message to be entered into the broadcast data set for a specified user, or to be sent to an operator console.

**LSQA:** The START and MOUNT commands no longer support the LSQA keyword. The local system queue area (LSQA) is dynamically expandable rather than fixed in size.

**System Log:** The system log function is started automatically during IPL, and can now be restarted after being closed.

### Job Entry Subsystem Support

In MVS, the job entry subsystem (JES2 or JES3) functionally replaces the SYS1.SYSJOBQE data set and the OS readers and writers. Support is therefore removed for operands associated with these functions. Commands affected by



these changes are CANCEL, DISPLAY, HOLD, MODIFY, RELEASE, RESET, and SET for queues, and SET, START, MODIFY and STOP for readers and writers. START, STOP, and MODIFY are still supported for external writers.

The job entry subsystem available with the initial version of MVS is JES2, which is generally compatible with HASP II. JES3, generally compatible with ASP Version 3, will be available at a later time.

Some of the operator commands entered (see figures 30-32) will be job entry subsystem commands. For the most part, the job entry subsystem operator commands are identical to those of the historical system commands, but with a special character identifier preceding the verb.

JES2 Command Definition	Command Function
BACKSPACE	Backspace a printer.
CANCEL	Cancel a job or a device function.
DISPLAY	Display disks, units, lines, remotes, messages, jobs, queues, activity, initiators, or outstanding requests.
Forward Space	Space a printer forward.
HALT (immediate)	Stops a device.
HOLD	Place all jobs or a specific job in a hold queue.
INTERRUPT	Interrupt a printer.
LIST	Displays queued job output.
OUTPUT	Cancel and release held data sets.
RELEASE	Remove all jobs or a specific job from a hold queue.
REPEAT	Repeat a device function.
RERUN	For device functions or jobs in execution.
Route Output	By routing group or job.
SET	For a device, initiator, job, or system job number base.
START	For a device, initiator, or system.
STOP (after current function)	For a device, initiator, system, or job.
Transfer Command	For an OS/VS2 command.

Figure 30. JES2 Command Functions

JES3 Command Definition	Command Function
CALL	Schedules a dynamic support function.
CANCEL	Cancels a dynamic support function.
DUMP	Takes a SNAP dump of the Global JES3 processor.
Inquiry	Obtains information regarding: <ul style="list-style-type: none"> <li>● The processing status of a specific job.</li> <li>● The current status of major JES3 queues.</li> <li>● The amount of space remaining in the JES3 spool data sets or JES3 buffer pool.</li> <li>● The status of devices (input, output, utility, and setup) under JES3 control.</li> <li>● The status of jobs under DJC.</li> </ul>
MODIFY	Alters the status of: <ul style="list-style-type: none"> <li>● A job or group of jobs — changes priority, cancel job(s), change hold status, etc.</li> <li>● A Deadline Scheduling algorithm.</li> <li>● DJC</li> <li>● job selection algorithm.</li> <li>● JES3 job queue.</li> <li>● work queued to a JES3 output device.</li> </ul>
RESTART	Discontinue processing of a job on a specified DSP and/or reschedule with different DSP processing options.
START	Restart processing of a stopped DSP and/or specify DSP processing options.
VARY	Remove processors or devices from JES3 control, or reconnect them.

Figure 31. JES3 Command Functions

JES3 Console Control Command Definition	Function
DELAY	Change the time interval between the display of messages on the JES3 graphic console that entered the command.
DISABLE	Disable a specific JES3 console.
ENABLE	Enable a specific JES3 console.
ERASE	Erase the screen of the JES3 graphic console that entered the command.
FREE	Purge all messages to the console that entered the command.
Inquiry	Obtain information regarding: <ul style="list-style-type: none"> <li>● 3270 program function keys.</li> <li>● Status of JES3 consoles.</li> <li>● Outstanding action (REPLY) messages.</li> </ul>
MESSAGE	Send a message to another JES3 console.
MODIFY	Alter the command authority or the JES3 route codes that are assigned to a specified JES3 console.
SEND	Send a system command to the Global or Local processor.
SWITCH	Switch the writing of messages from one JES3 console to another.

Figure 32. JES3 Control Command Functions

## **JES2 Command Functions**

A brief description of JES2 command functions follows.

**Job Related Functions:** The job entry subsystem has the ability to:

- Release all jobs or specific jobs from the hold queue.
- Display, for each active job entered through JES2, job related information (e.g., that the job is either executing, being purged, or on a device).
- Display the number of jobs queued for special forms processing for either remote or local, or both.
- Display jobs queued for JES2 services, and which services they are waiting for.
- To place jobs currently in the system and entered through JES2, on the hold queue.
- To cancel specific jobs (JES2 will issue SVC 34 to cancel jobs in execution in the system). Jobs cancelled will have their current activity deleted and will be scheduled for purge processing.
- Display for specific jobs entered through JES2, their current status (e.g., the job is in system execution, or queued for print processing).
- To schedule a job for purge processing after its current activity has completed.
- To set the new base number for automatic job number assignments (i.e., all jobs entered through JES2 will be assigned unique job identifiers starting from the operator-set base).

**Device List Functions:** Device list commands perform the following functions:

- Backspace a printer device.
- Restart a current activity on an output device, (i.e., the current activity on the designated device will be terminated and, in the case of printer/punch devices, the activity will be returned to the appropriate queue in order of priority and made eligible for selection. In the case of remote job entry lines, JES2 will, upon completion of the current line I/O, cancel all activities on the line).
- Forward space a printer device.
- Interrupt current activity on a printer device, (i.e., the current activity on the designated printer(s) will, if active, be checkpointed and terminated, with the activity requeued on the appropriate JES2 queue).
- Repeat a current activity on a printer/punch device, (i.e., after the current activity has been completed, the activity will be requeued on the appropriate queue to schedule a repetition of the activity).
- START/STOP a device.
- Place all jobs to be read into JES2 from a specified reader, on the hold queue.
- Set a printer or punch to handle different characteristics (e.g. forms, FCB loads, and UCS trains/chains).

**System Functions:** System commands which control the ability of JES2 to process jobs through the system may be broken down into two groups:

- Initiator Commands -- those commands which control the actual selection and submission of jobs from JES2 to the system for processing.

- JES Commands -- those commands which control the ability of JES2 to process jobs for any function.

**Initiator Commands:** Initiator commands perform the following function:

- Display for all JES2 initiators, their status (e.g., the job classes, their processing, whether they are active, inactive (waiting for new jobs), or draining.
- Stop an initiator.
- Start an initiator.
- Set the job classes of jobs that the initiator will handle.
- With respect to jobs read into JES2 and on an input queue, change that job's queuing priority and job class.

**JES START/STOP Commands:** With JES2 START/STOP commands, the facility exists:

- For an operator to stop JES2, which will stop all JES2 job processing and prevent any new function from beginning.
- For an operator to start JES2.

**Miscellaneous Display Function:** Display functions include:

- The ability to display the device addresses and volume serial numbers of all online direct access storage devices.
- The ability to display the status (JES2) of all devices (readers, printers, punches) on a remote line.
- The ability to display the status (JES2) of all JES2 controlled, non-direct access devices attached to the local system as well as the corresponding hardware address of the device.

**Remote Job Entry Function:** With remote job support the operator has the ability to:

- Display a message at a remote terminal, whether logged-on, signed-on, or inactive.
- Route a job's output to a remote terminal, whether that terminal is logged-on, signed-on, or inactive.

### **JES3 Command Functions**

Operational control is maintained at every stage of processing for the following functions:

- The operator can delete a job from the system by issuing the CANCEL message from the console.
- The operator can restart applicable job segments on the Global processor by issuing the RESTART message at a console.
- The operator can change the priority of a particular job. This option is usually used to expedite the start of job processing.
- A job can be held in or released from a queue. A job previously in hold status can be released to be scheduled for additional processing, and jobs can be held and released on the basis of priority.

- Jobs received from remote workstations can be held and released on an individual terminal basis. Jobs in hold status from a particular terminal can be released entirely, or the operator can specify the number of jobs to be released.
- Using the CALL facility, the operator can request the system to schedule background support functions to be executed concurrently with the standard preprocessing and postprocessing facilities.
- Input/output and terminal devices attached to the Global JES3 processor can be dynamically reconfigured. These devices can either be removed from available-for-processing status, or be made available by the operator. Also, their output can be rerouted.
- The operator can query the system regarding the job queue for various functions, such as estimated processor running time and estimated number of lines of print. In addition, the operator can request the status of an individual job, as well as a printout of the entire job queue status.

## **Time Sharing Considerations**

Changes to the time sharing (TSO) facilities for MVS support system resources management, VSAM, allocation, the job entry subsystem, the EDIT function, accounting, and the integration of time sharing into the system. Changes to the time sharing commands of MVT and VS2 Release 1 are described in Figure 33.

### ***System Resources Management Support***

The system resources management routines allow an installation to group together into performance groups, user transactions with similar performance requirements. (See the chapter "Directing the Use of System Resources.")

An operand of the LOGON command allows the user to specify a performance group number.

The ACCOUNT facility specifies the performance groups that the time-sharing user is authorized to request at LOGON time.

### ***VSAM Support***

Both VSAM and non-VSAM data sets can exist in MVS. A combination of the TSO utilities available in MVT and VS2 Release 1, and access method services are used to manipulate data sets in MVS.

The following access methods services commands are fully supported for interactive use at terminals: DEFINE, ALTER, CNVTCAT, DELETE, LISTCAT, REPRO, PRINT, EXPORT, IMPORT, and VERIFY. These commands are discussed in the topic "Data Set Conversion" in this chapter.

The DELETE and LISTCAT utilities are not supported. Their functions are supported for the VSAM catalog, with commands of the same name, by access method services.

LISTALC and LISTDS support both VSAM and non-VSAM data sets.

The ALTER command provides the functions of PROTECT and RENAME. The latter two apply only to non-VSAM data sets.

### ***Allocation Support***

The ALLOCATE, FREE, and ATTRIB commands support dynamic allocation through device type and disposition specification, concatenation and deconcatenation of data sets, multivolume and multiunit requests, and setting DCB parameters relating to tape volumes.

ALLOCATE and SEND are added as subcommands of the EDIT command.

The MOUNT/NOMOUNT parameter of the ACCOUNT command specifies that dynamic allocation requests can cause volume mounting.

Command	Subcommand	Parameter		Change
ACCOUNT	ADD	DEST(stationid)/ <u>LOCAL</u>	⊙	Specifies the default destination for SYSOUT data sets defined by this user. The default is the central output facility.
		MOUNT/ <u>NOMOUNT</u>	⊙	Specifies that dynamic allocation requests can cause volume mounting.
	PERFORM(group)/ <u>NOPERFORM</u>	⊙	Specifies the performance group (if any) that the TSO user is authorized to request at LOGON.	
	CHANGE	The new parameters are the same as those listed under the ADD subcommand.	⊙	The functions are basically the same as described under the ADD subcommand, except that there are no default values.
	SYNC		⊙	Builds a new BROADCAST data set and synchronizes it with the UADS.
ALLOCATE		DATASET	⊙	Extended to allow concatenation of cataloged data sets.
		DEST (stationid) HOLD/ <u>NOHOLD</u>	⊙	These parameters support redirected SYSOUT. DEST allows the user to specify a workstation to which all SYSOUT data sets are directed. HOLD/NOHOLD enables the user to place the SYSOUT data sets on a hold queue for later processing.
		DUMMY AVBLOCK/TRACKS/ CYLINDERS RELEASE ROUND KEEP/DELETE/ CATALOG/ UNCATALOG	• • • ⊙ ⊙	These parameters and their functions are directly analogous to their JCL counterparts.
		UNIT (unit-type)	⊙	These parameters provide device support. UNIT specifies the kind and maximum number of devices allocated to the file or data set. The (unit-type) may be an installation - defined group name, a generic device - type, or a specific device address.
		UCOUNT (unit-count)/ <u>PARALLEL</u>	⊙	The unit count specifies the maximum number of devices to be simultaneously mounted for the file or data set.
		VOLUME MAXVOL POSITION PRIVATE LABEL VSEQ	⊙	Through these parameters, multivolume data sets are supported. The user can define the volume serial numbers, limit, relative position of the data set, and volume and label information. These parameters have corresponding JCL statements in MVT and VS2 Release 1.
		SYSOUT (class)	⊙	Class can be specified for SYSOUT.
ATTRIBUTE		OPTCD RECFM BUFOFF DSORG LIMCT DEN TRTCH DIAGNS	⊙	This command has been extended to allow the specification of certain tape characteristics that support allocation.

Figure 33. Changes to Time Sharing Commands for VS2 Release 2 (Part 1 of 3)

Command	Subcommand	Parameter		Change
CANCEL/STATUS		jobname (jobid)	⊙	The jobid is optionally needed to uniquely identify a particular job, if more than one has been submitted with the same jobname.
EDIT	ALLOCATE	Same as command	⊙	ALLOCATE is added as an EDIT subcommand.
	SEND	Same as command	⊙	SEND is added as an EDIT subcommand.
	RUN	LIB (dslist)	⊙	Concatenated load libraries are supported for the RUN subcommand.
	DELETE (DEL)		⊙	The DELETE subcommand abbreviation is changed to DEL.
	END		⊙	The EDIT input data set is freed upon command termination.
	PROFILE	Same as command	⊙	PROFILE changes are available under EDIT.
EXEC		PROMPT	⊙	This parameter allows prompting during the execution of CLIST operations.
FREE		DEST (stationid)	⊙	These parameters support redirected SYSOUT. The option is provided to change the job entry subsystem user to which SYSOUT data sets are directed, and to override the current HOLD specification.  The disposition parameters are analogous to their JCL counterparts.
		HOLD/NOHOLD	⊙	
		KEEP/DELETE/ CATALOG/ UNCATALOG/ SYSOUT (class)	⊙	
LOGOFF		HOLD/DISCONNECT	⊙	A telecommunications line may be held (i.e., not dropped) at LOGOFF time.
LOGON		MOUNT/NOMOUNT	⊙	Specifies that dynamic allocation requests can cause volume mounting.
		PERFORM/ NOPERFORM	⊙	Specifies the performance group (if any) that is authorized for this user.
		RECONNECT	⊙	Allows the user to log back onto the system and pick up the session -- within an installation - defined period of time.
OPERATOR	CANCEL		⊙	The jobname, unit address, and identifier operands are no longer supported. Their functions are handled by the job entry subsystem.
	DISPLAY	A jobname N = LIST Q = LIST	⊙	The parameters are no longer supported. Displaying job and queue data is a function of the job entry subsystem.
		USER	⊙	Replaced by DISPLAY TS.
		R [,LIST] JOBS [,LIST] TS [,LIST]	⊙	DISPLAY TS LIST provides information formerly obtained with DISPLAY A or DISPLAY USERS.
	MODIFY		⊙	No longer supported.
	MONITOR	DSNAME	⊙	The DSNAME and SPACE parameters are no longer supported for these subcommands.
	STOPMN	SPACE	⊙	

Figure 33. Changes to Time Sharing Commands for VS2 Release 2 (Part 2 of 3)



Command	Subcommand	Parameter		Change
OPERATOR (continued)	SEND	SAVE  DELETE/LIST/ALL BRDCST  OPER CN	<ul style="list-style-type: none"> <li>•</li> <li>⊙</li> <li>⊙</li> <li>•</li> </ul>	<p>The message is placed in the broadcast data set, and will not be printed at the receiving terminal until the <u>next</u> LOGON -- even if the user is currently logged onto the system.</p> <p>There is no longer a default parameter.</p> <p>Specifies that the message is to be sent to all active consoles.</p> <p>Specifies that the message is to be sent to a specific operator console. Messages sent via the OPER or CN keywords will have respectively OPER or CN appended to the message.</p>
OUTPUT		DEST (stationid)  HOLD/ <u>NOHOLD</u>  KEEP/ <u>NOKEEP</u>  NEWCLASS DELETE  NOPRINT	<ul style="list-style-type: none"> <li>⊙</li> <li>⊙</li> <li>⊙</li> <li>⊙</li> <li>⊙</li> </ul>	<p>Specifies the remote station user to which SYSOUT data sets -- for jobs and classes specified in other parameters -- are to be directed.</p> <p>Output data sets are held on a queue until the user wants to print them to his terminal (HOLD), or are to be printed immediately by the system writer.</p> <p>Specifies whether to keep the SYSOUT data set after printing it.</p> <p>These parameters perform functions previously performed by variations of the NOPRINT parameter.</p> <p>No longer supported. Functionally replaced by NEWCLASS and DELETE.</p>
PROFILE		LIST  <u>MODE/NOMODE</u>	<ul style="list-style-type: none"> <li>⊙</li> <li>⊙</li> </ul>	<p>Prints out the user's profile.</p> <p>Mode messages are printed under the EDIT command.</p>
SEND		USER (*)  SAVE  WAIT/NOWAIT  OPER CN	<ul style="list-style-type: none"> <li>⊙</li> <li>⊙</li> <li>⊙</li> </ul>	<p>The message is sent to the stationid of the terminal issuing the request.</p> <p>The message is placed in the broadcast data set, not to be printed until the next LOGON -- even if the recipient is currently logged onto the system.</p> <p>WAIT -- The transmitting terminal is locked until the receiving terminal can accept the message.</p> <p>NOWAIT -- If the receiving terminal is busy, the message is not sent.</p> <p>Specifies that the message is to be sent to a specific operator console. Messages sent via the OPER or CN keywords will have respectively OPER or CN appended to the message.</p>
SUBMIT			<ul style="list-style-type: none"> <li>⊙</li> </ul>	<p>The account number under which the user logs on is included on the JOB card generated by the SUBMIT command.</p> <p>The jobid is printed with the JOB SUBMITTED message.</p>
TIME			<ul style="list-style-type: none"> <li>⊙</li> </ul>	<p>The local time-of-day and the cumulative service units are printed.</p>
<ul style="list-style-type: none"> <li>• New or Changed from MVT.</li> <li>○ New or Changed from VS2 Release 1.</li> <li>⊙ New or Changed from MVT and VS2 Release 1.</li> </ul>				

Figure 33. Changes to Time Sharing Commands for VS2 Release 2 (Part 3 of 3)

## ***Job Entry Subsystem Support***

The FIB (foreground initiated background) commands SUBMIT, CANCEL, STATUS, and OUTPUT, interface with the job entry subsystem to submit or access a background job.

The restriction that was previously specified at SYSGEN limiting the number of jobs submitted by a user, is removed in MVS.

ALLOCATE, FREE, OUTPUT, and SUBMIT allow SYSOUT data sets to be routed to a remote station.

Some of the operands of the DISPLAY subcommand are no longer supported, since displaying job and queue data are functions of the job entry subsystem.

## ***Edit Support***

The EDIT command is used to create, modify, display, and store data. FBS (fixed-block standard) record format is used for new data sets created by the SAVE command, when fixed records are being edited.

ALLOCATE, SEND, and PROFILE are now subcommands of EDIT.

Initial entry into EDIT places the current line reference at the top of the data set. Initial entry through the null line ENTRY continues to position this reference at the end of the data set.

## ***Accounting Support***

A utility (see the topic "Reformatting the UADS" in this chapter) is provided to convert the SYS1.UADS data set to the MVS format. LOGON supports both the old and new format. However, since all updating of accounting fields is done in the new format, accounting fields of an old format UADS must be reinitialized.

The ACCOUNT command and subcommands can now be run in the background.

## ***Time Sharing Integration Support***

Under MVT and VS2 Release 1, time sharing (TSO) was an option which was started by the operator. A TSO task was created which could be modified with a subcommand of the OPERATOR command. This MODIFY subcommand is no longer supported, since there is no time sharing task.

Time sharing is started and stopped by the operator issuing a MODIFY TCAM TS=START or STOP. All the installation performance optimization (e.g., driver Q parameters), performance objectives (e.g., background percentage), and system control functions previously specified by START or MODIFY TS commands, are performed by the system resources management routines.

The SYS1.UADS and SYS1.BROADCAST data sets must be mounted at system initialization since they are allocated at that time.

The SPACE and DSNAME parameters of the MONITOR subcommand are not supported, since there is no time sharing task.

## Catalog Support

Both VSAM catalogs and OS control volumes (CVOLs) are supported in MVS, but the system master catalog is a VSAM catalog.

The VSAM catalog supports both OS and VSAM data sets, and utilities are provided for:

- Creating a VSAM catalog
- Converting to a VSAM catalog from an OS catalog
- Updating a VSAM catalog from an OS catalog
- Updating an OS catalog with the non-VSAM data set changes from SMF data

The OS CVOL support is provided to ease the multi-CPU installation through the migration to MVS. It is not necessary to convert VSAM user catalogs and OS CVOLs back and forth if there are conversions of operating systems due to changing workloads.

## The VSAM Catalog

The structure of VSAM catalogs is different from that of OS catalogs. For VSAM data sets, the VSAM catalog functions somewhat like an OS VTOC in that the VSAM catalog entry contains the cylinder and track address of the data set, as well as volume and device type information. For non-VSAM data sets, the catalog contains the same information as its OS counterpart. As in OS, a non-VSAM data set can be found either through the catalog or through the VTOC of the volume containing the data set. Figure 34 shows simplified relationships between OS and VSAM catalogs.

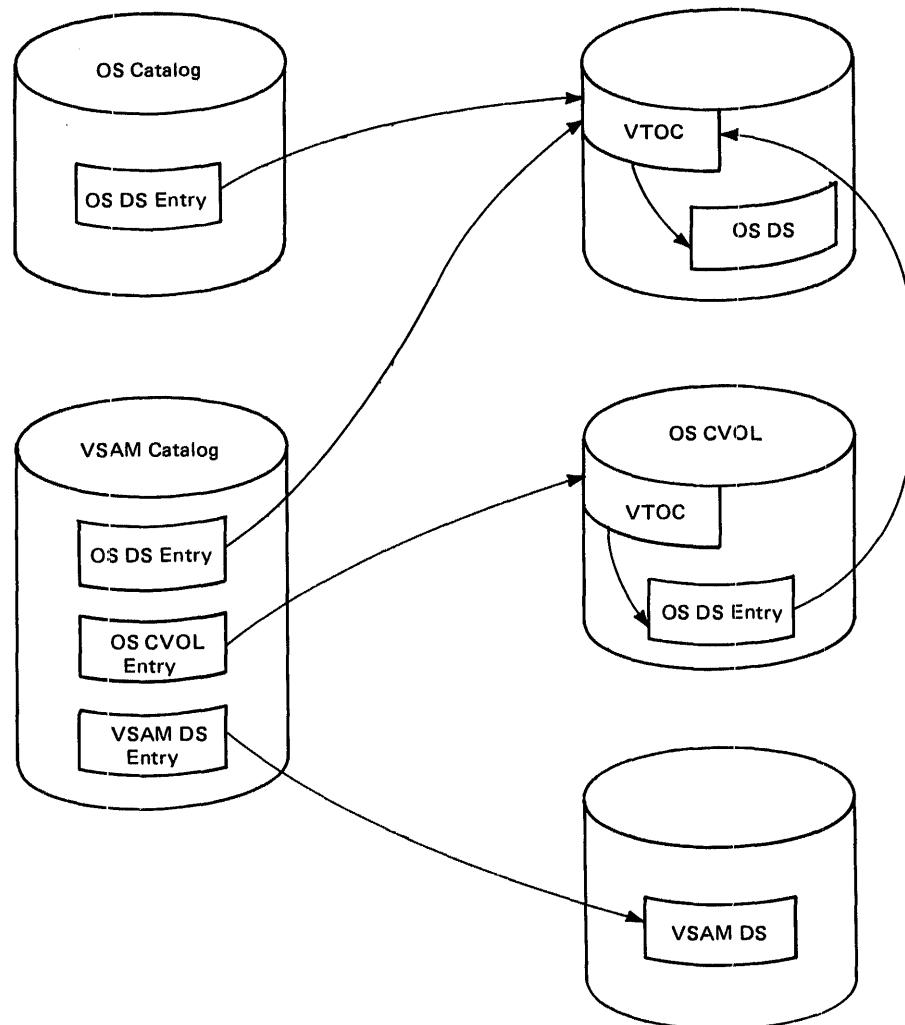


Figure 34. Simplified Catalog Structure – OS vs. VSAM

## VSAM Order of Search

The order of search for LOCATE may be altered in MVS by using VSAM DD statements defining a specific catalog to be used as the primary catalog for a single job or job step. The order of search for the specified data set is then:

- First, all JOBCAT/STEP CAT catalogs in the order they are specified. (This does not apply to OS CVOLs.)
- Second, the catalog indicated by a catalog connector (analogous to CVOL pointer) in the data set name,

or

the master catalog, if there is no connector in the data set name.

MVS, through the access method services, provides the means to create VSAM catalogs and to update the VSAM catalogs with entries from OS catalogs. It also provides through the IEHUCAT utility, the means for updating OS catalogs with the non-VSAM data set entry changes to the VSAM catalogs. Some changes to the catalog functions of system utilities and to the CAMLST macro have resulted.

## Creating or Updating the VSAM Catalog

Creating and/or updating VSAM catalogs is accomplished on the MVS system through the access method services (Figure 35).

Function	Access Method Services Verb
Define a VSAM data set or catalog, and catalog non-VSAM data sets.	DEFINE
Define and delete aliases for catalog names and non-VSAM data set names.	DEFINE, DELETE
List VSAM and non-VSAM catalog entries or records of a data set.	LISTCAT, PRINT
Alter entries in a specified VSAM catalog	ALTER
Print data set records.	PRINT
Move or copy a VSAM catalog.	REPRO
Convert to a VSAM catalog.	CNVTCAT
Convert a sequential or an indexed sequential data set to the VSAM format.	REPRO
Convert a VSAM data set to sequential format.	REPRO
Make a data set portable from one operating system to another.	EXPORT, IMPORT
Copy a VSAM data set for reorganization.	REPRO
Create a back-up data set.	EXPORT
Support generation data groups (GDGs).	DEFINE, DELETE

Figure 35. Functions of the Access Method Services

The process of creating the VSAM catalogs and updating them with entries from the OS catalogs might typically go like this:

- At SYSGEN, the DATASET macro is used to create the new VSAM master catalog.
- The new system's catalogs are created using DEFINE. For example, one master catalog might be defined to correspond to the OS system catalog, and one user or private catalog defined for each OS CVOL.

- CNVTCAT is run using an OS system catalog as input and the MVS master as the target catalog. All CVOL pointers are related to their corresponding VSAM user catalogs. The result is that the VSAM master catalog is a functional replacement for the input catalog.
- CNVTCAT is run for all OS CVOLS with VSAM catalog(s) as target.
- ALTER can be used to password protect the catalog(s).

### Updating an OS Catalog From a VSAM Catalog

The IEHUCAT utility is used to update the OS catalogs to reflect all actions taken for the non-VSAM data sets in a VSAM catalog. IEHUCAT is run on the system that has the catalogs to be updated.

MVS catalog management will optionally write to the SMF file, records reflecting additions, deletions, and alterations to the catalogs. IEHUCAT uses this information from the SMF file to update the OS catalogs. No SMF records are written when OS CVOLs are accessed.

### Catalog Utility and Macro Changes

The following changes have been incorporated into the IEHLIST, IEHPROGM, and IEHMOVE utilities, and the CAMLST macro:

- IEHLIST -- LISTCTLG has been eliminated. It has been replaced by the access method services function, LISTCAT.
- IEHMOVE -- All catalog-related functions have been deleted or replaced by access method services.

Deleted	Functions
MOVE	DSGROUP
COPY	DDSGROUP
MOVE	CATALOG
COPY	CATALOG

- IEHPROGM -- All catalog-related functions have been removed and replaced by access method services, except for CATLG and UNCATLG which are still available for non-VSAM data sets.

Function	Access Method Services Replacement
BLDX	none
DLTX	none
BLDA	DEFINE
DLTA	DELETE
BLDG	DEFINE
CONNECT	DEFINE or IMPORT
RELEASE	DELETE or EXPORT

- CAMLST -- The CAMLST macro is supported as follows:

Operation	Supported
LOCATE by name	yes
LOCATE by TTR	no
CATLG	yes - non-VSAM only
UNCATLG	yes - non-VSAM only
RECATLG	yes - non-VSAM only
BLDX	no
DLTX	no
BLDA	no - use DEFINE
DLTA	no - use DELETE
BLDG	no - use DEFINE
LNKX	no - use IMPORT or DEFINE
DRPX	no - use EXPORT or DELETE

## OS CVOL Support

The CATALOG and CATBX, UNCATALOG and UCATDX, RECATALOG, and LOCATE by name functions are supported in MVS. In addition, VS2 Release 1 and MVT-type TSO requests that formerly accessed CVOL catalogs are supported. Figure 36 describes the method of processing requests on OS CVOLs.

OS Requests	Processing method
CATALOG	CATBX
CATBX	CATBX
UNCATLG	UCATDX
UCATDX	UCATDX
LOCATE by name	LOCATE by name
RECATLG	RECATLG
Access Method Services DELETE	UCATDX
Access Method Services LISTCAT	LOCATE

Figure 36. Method of Processing OS CVOL Requests

### Defining an OS CVOL for MVS

In order to use an OS CVOL catalog in MVS the user should:

- Identify the CVOL catalog to the VSAM master catalog by defining (DEFINE) SYSCTLG as a nonVSAM data set. The data set name must be in the form SYSCTLG.xxx. That is, the first qualifier must be 'SYSCTLG.' followed by any installation-defined unique name.



- Define an alias to create a unique simple name in the VSAM master catalog (DEFINE ALIAS for the SYSCTLG). This alias acts as a CVOL pointer.

### Accessing CVOL Catalogs

The first simple name in an OS MVT catalog could indicate that the data set was not on that volume, and direct the search to another volume (a CVOL). In MVS, a similar relationship is used to specify a user catalog on which to search for a data set.

A DEFINE ALIAS command is issued to create a simple name in a VSAM master catalog. This simple name is connected to the name of the user catalog, and the resultant simple name can then be used as the first simple name of the data set name.

This technique allows a DEFINE ALIAS command to create a simple name that is connected to the name of a CVOL. Since alias names are unique in the VSAM master catalog, a given high-level name or TSO userid can refer to a VSAM user catalog or a CVOL catalog, but not both. CVOL catalogs existing on shared DASD can be accessed and updated from MVS systems without prior conversion on that CVOL. However, when the user builds the VSAM master catalog, every CVOL and every high-level name to be accessed must be defined.

A CVOL catalog is defined in a VSAM master catalog by issuing the DEFINE NONVSAM command with the data set name in the form SYSCTLG.UNIQUENAME. This restricts the name SYSCTLG to CVOL catalogs.

### Compatibility for OS CVOL Requests

The following items should be considered when using CVOLs with this release:

- In MVS, there is no facility for executing a BLDX or DLTX in a CVOL. OS CATALOG and UNCATLG are mapped to CATBX and UCATDX respectively.
- For an OS LOCATE by name, the maximum number of volumes that can be returned is 20. The LOCATE by TTR to obtain additional volume lists is not supported.
- Output for VSAM requests to OS CVOLs is mapped into VSAM formats.
- An alias name for a CVOL can never be deleted at the highest index level.
- JOBCAT/STEPSCAT control cards are not supported for CVOLs.
- A CVOL may not point to another CVOL. There is no support for this in MVS.
- OS CVOL functions not supported include: BLDG, LNKX, DRPX, BLDA, DLTA, BLDX, DLTX, LOCATE by TTR, and EXTEND into secondary extents.
- Access method services functions not supported for CVOLs include: DEFINE, ALTER, and LISTCAT for listing an entire catalog.
- The CAMLST macro does not support the CVOL parameter; it will be ignored. The CVOL must be specified in the VSAM master catalog.
- CATALOG/RECATALOG requests are restricted to a maximum of 255 volumes in the volume list.

- OS CVOL management does not write records to the SMF data set.
- The installation must control the use of the name SYSCTLG in the VSAM master catalog.
- It is strongly recommended that SYSCTLG data sets be password-protected for writing.

## **Data Set Conversion**

The broadcast data set and the user attribute data set must be converted to the new MVS format.

### ***Reformatting the UADS***

In order to use any of the new keywords on the ACCOUNT command subcommands, the Release 2 format of the user attribute data set (UADS) must be used. The Account facility utility program named UADSREFM is used to create a new UADS or to reformat an existing UADS. This program reformats the UADS by reading a member from the existing UADS, copying its structure, incorporating the required new information, and writing the reformatted member to the new UADS. This reformatting process also eliminates any inefficient spacings that may have resulted from previous manipulations of a member's structure.

The UADSREFM utility program is invoked in an Account Facility input stream that includes a UADSREFM control card in the control data set.

If the MVS UADS format is not used, an error message will be issued when any of the new ACCOUNT subcommand keywords are used.

### ***Formatting the Broadcast Data Set***

The SYNC subcommand of the ACCOUNT command is used to create a broadcast data set in the Release 2 format. The SYNC subcommand also synchronizes the user identifications in the UADS with the user identifications in an existing broadcast data set. It should therefore be used immediately after the creation of a new UADS or the reformatting of an existing UADS.

The SYNC subcommand may be used from the foreground or included in the control data set of an Account Facility input stream.

If the MVS format of the broadcast data set is not used, an error message will be issued the first time an attempt is made to access it. Processing will continue, but the broadcast data set will be inaccessible.

## **Programming Considerations With Virtual=Real Addresses**

When the ADDRSPC=REAL parameter is coded, a job is assigned to nonpageable (V=R) storage. The nonpageable dynamic area is used for programs that are not to be paged during execution. Use of this area should be reserved for programs that cannot be executed in pageable dynamic storage and for programs that are not readily adaptable to a pageable environment.

If the nonpageable dynamic area is to be used, the following items should be considered:

- A large nonpageable dynamic area has a degrading effect on system performance, even when no regions in this area are allocated. This situation occurs because the allocation routines will perform extended searches and page manipulation in order to avoid allocation of long-term fixed pages and SQA and LSQA pages in the nonpageable dynamic area.
- To minimize the allocation of long-term fixed pages and SQA and LSQA pages in the nonpageable dynamic area, the size of the nonpageable dynamic area should be kept as small as possible in relation to the actual size of real storage. In a heavily overcommitted system, allocation of long-term fixed pages and SQA and LSQA pages in the nonpageable dynamic area becomes more likely and may present severe restrictions in allocating regions in this area.
- If long-term fixed pages or SQA or LSQA pages are allocated in the nonpageable dynamic area, regions cannot be allocated in the area that intersects with these pages. Since the anticipated duration of these pages is the life of the IPL, it is possible that the existence of these pages will have a cumulative degrading effect on the system.
- If short-term fixed pages are specified when long-term fixed pages should be specified, undetectable allocation in nonpageable dynamic area will occur. Also, severe system degradation will occur. (The long or short fix option selected should be based on the length of time between when a page is fixed and freed. As a rule of thumb, the long option should be specified if the time interval of the fix can be measured as a number of seconds.)
- If the actual usage of the nonpageable dynamic area is close to the size of the nonpageable dynamic area, any allocation of long-term fixed pages or SQA or LSQA pages that does occur is more likely to impact the ability to perform allocation in this area.
- When allocation is attempted for a region in the nonpageable dynamic area, any contiguous space that does not contain any long-term fixed pages or SQA OR LSQA pages may be used. Although this will increase the probability of successful allocation for this request, it will also increase the probability of fragmentation in the nonpageable dynamic area.
- In a heavily I/O-oriented system, allocation in the nonpageable dynamic area may be blocked due to a high amount of fixed pages for I/O. Conversely, the existence of an allocated region in the nonpageable dynamic area may prevent I/O fixing. This situation occurs because allocated pages for the nonpageable dynamic area are included in fix threshold calculations.

## Program Conversion

MVS is generally upward compatible with MFT, MVT, VS1 and VS2 Release 1. Object programs and load modules that operate in MFT, MVT, or VS1 and follow IBM programming conventions described in *OS/VS Supervisor Services and Macros GC27-6979*, *OS/VS Data management Services Guide GC26-3768*, and *OS/VS Data Management Macros GC26-3793*, will operate without change in this release. Some exceptions are:

- Programs sensitive to the PSW format (if migration is from an OS system).
- Programs that modify, are modifications of, or depend upon implementation of, the MFT, MVT, VS1, or VS2 Release 1 control program.
- Programs sensitive to the OS catalog structure.
- Programs that reference areas not normally available to a problem program through system functions.
- In a multiprocessing environment with shared real storage, multi-tasking programs which reference the same area should be reviewed for synchronization dependencies.
- Programs dependent upon MVT or VS2 Release 1 integrity exposures.
- EXCP, NOTE or POINT to a SYSIN or SYSOUT data set.
- User-written appendages which might now run enabled.
- Issuing the SSM instruction causes abnormal termination.

The capability to execute with virtual addresses equal to real (ADDRSPC=REAL) for programs that must operate entirely in real storage is provided for:

- programs that use EXCP or XDAP with user written appendages.

**Note:** Programs that use EXCP should observe the conventions outlined in the channel characteristics manual for the host system.

- programs that dynamically modify channel programs.
- programs that are time dependent (such as MICR).

## *TCAM Conversion*

OS TCAM message control programs must be reassembled to run in the MVS environment. This reassembly allows the MCPs to benefit from the virtual storage capability of MVS. Under MVS, TCAM runs as a subsystem in a virtual address space. Certain TCAM elements, such as the buffer pool, I/O appendages, control blocks, and tables, are fixed in real storage for the duration of the TCAM task.

In most cases, TCAM application programs need only to be linkage edited.

## **Multiprocessing Considerations**

MVS supports two CPUs configured to share storage units in a mode called tightly coupled multiprocessing. The two CPUs must be of the same model type; either two model 158 CPUs or two model 168 CPUs.

All available real storage is addressable by both CPUs. The only copy of the control program that exists in storage treats each CPU as a resource; nominally either CPU can be dispatched to perform any ready task at any time. Because two CPUs increase the demand for service from the single set of supervisor routines, a system of locks has been introduced to permit functions to be executed in parallel, one by each CPU. Where each CPU requests the same service at the same time, the locks ensure that the service is rendered in sequence.

In most cases the control program manages the multiprocessing configuration in such a way that it is logically indistinguishable from a uniprocessor. However certain unique aspects, in addition to those described in the *Introduction to MVS*, should be noted, including:

- Hardware configuration considerations.
- Operator control of configuration.
- Recovery considerations.
- Programmer considerations.
- Loosely-coupled configurations.

### ***Hardware Configuration Considerations***

Some hardware considerations for MVS are:

- Except for the presence of two CPUs, the minimum configuration is the same as for a uniprocessor.
- The real storage provided is typically double that provided for a single CPU of the same model.
- I/O devices need be accessible through only one channel path by one CPU. Unit record equipment is typically attached this way. For a device attached in this manner, no attachment of another control unit may be made to the channel position of the same number on the other CPU.
- Tape and direct access devices are normally accessible by both CPUs. In a typical case, control units are equipped with programmable 2-channel switches, and each path is attached at the same channel of each CPU. This type of I/O configuration is called symmetrical.
- 2914 manually controlled switching units can be used to improve the availability of asymmetrically attached I/O.

### ***Operator Control of Configuration***

Multiprocessing requires some additional education of operators. They must be aware of the following facets of the system:

- By means of the VARY command, one CPU can be dynamically added to or deleted from the configuration. The VARY command can also be used to associate a channel with a tape or direct access device.

- Addressable storage is configured from a collection of units. The real address range to be represented by a unit can be set by means of hardware dials. Addressable real storage can be dynamically decremented or incremented, and need not be contiguously addressable.
- Two uniprocessors can be dynamically configured from a multiprocessor.
- A primary use of the VARY and QUIESCE commands is to logically delete elements (storage units, CPU, channels, control units, and devices) from the configuration for test purposes.
- If a CPU encounters a check condition, the system automatically attempts to continue operation without using the offending element. The operator is informed of the action taken, and under certain circumstances controls the follow-on procedure.
- JES3 provides console service on the Global processor specifically designed for controlling a multiple processor complex. JES3 allows the operator to assign from the Global processor, a master console for a Local processor, to send commands to Local processors and receive replies directed to that console, and to route messages (using MCS route codes) from all Local processors to JES3 or MCS consoles of the Global processor. JES3 console services supports up to 96 route codes, including routing of device messages by user-specified device groups.

### ***Recovery Considerations***

Several recovery and retry functions are applicable to MP in MVS, some of which require operator intervention.

#### **Alternate Path Retry (APR)**

When multiple channel paths exist to a device, channel recovery retries occur on an alternate channel path. In an MP configuration the alternate path can be from the other CPU, but must previously have been assigned to the device performing the I/O operation. APR is an integral (not optional) part of MVS.

#### **Dynamic Device Reconfiguration (DDR)**

When a device error occurs, it may be possible to continue processing by moving the volume (tape or disk) or data stream (unit record) to another physical device. DDR is the primary recovery mechanism for permanent device errors. If an unallocated device (if direct access, it may be allocated) with characteristics similar to the failing device exists in the system, then the operator can be asked to SWAP the volume to the new device. Internal control blocks are exchanged, tapes repositioned, and I/O retried in order to continue processing. The swap of a direct access device may involve exchanging two volumes between their current spindles. The whole process is transparent to an application program. Other DDR considerations are as follows:

- DDR may also be invoked by the operator through a SWAP command, or the operator can override or reject a system-initiated SWAP.
- Devices operating with shared DASD support may only be swapped to the same address, and then only if it is not currently reserved. This address may be a different spindle with both pack and plug moved, but that requires an open spindle.

- Unit record record DDR can only be initiated by the operator using the SWAP command. The operator must take the device out of the ready state unless the current device status is "intervention required" or "not ready".
- Tape DDR requires an accurate block count for repositioning. This can have implications for EXCP, non-standard label, and emulator users. 3400 series tapes can only swap to other 3400 series tapes, but the 2400 series can swap to the 2400 or 3400 series. This can be significant for an installation deciding upon mixing types of tape drives.

### **Alternate CPU Recovery (ACR)**

When one CPU in an MP configuration fails, I/O and the routine being processed need to be recovered by the "good" CPU. This is done under the supervision and direction of ACR. Basically, the I/O is handled by error recovery as if a channel check had occurred. This means that alternate path retry and DDR are both possible as I/O recovery procedures. DDR is particularly important here, since it may reposition tapes through swapping to the same unit, thus offering retry for interrupted channel programs that otherwise would not be possible.

The routine that is active on the failing CPU is marked as possibly recoverable. If it has no recovery routines established (STAE, ESTAE, FRR) it is abnormally terminated. Programs with CPU affinity requirements to the failing CPU are also abnormally terminated.

### **Manual Switching**

Manual switching procedures usually involve stopping the CPU to which the path is to be switched. Generally the device should be reset prior to the switch in. The switching may be via the toggle switches on the control unit, or through a 2914 Manual Switch. The 2914 model 1 will reset the control units on the string being switched as part of the mechanical switching process. It can be used to switch unit record equipment between the CPUs of a multiprocessor.

### ***Programmer Considerations***

A problem program which executes successfully on a uniprocessor under MVS will, in almost all cases, run successfully on a multiprocessing configuration under the same control program. Exceptions that do arise usually involve multitasking where, under multiprocessing, two tasks can be executed simultaneously -- one by each CPU. The locking structure is described in the **Introduction to OS/VS2 Release 2, GC28-0661**. Situations that must be considered are:

- Implied wait dependencies, where a low priority task that POSTs a higher priority task, will not be resumed on a uniprocessor until the higher priority task issues a WAIT. With multiprocessing, the lower priority task can be resumed immediately by the second CPU. Both tasks might then be executed concurrently.
- Similarly, a task that issues ATTACH may find that the attached task immediately commences execution on the second CPU.
- If two tasks share the same data, and if logical consistency is necessary, explicit methods of ensuring serial access (e.g., WAIT/POST or ENQ/DEQ) must be used since the priority differences between the tasks will no longer ensure sequential access. Furthermore, data which is fetched from or stored

into such shared fields, even by a single instruction such as MVC, may not be internally consistent if these programmed sequencing controls are not used.

- Instruction streams (i.e., individual instructions) must not be dynamically modified by other instructions being executed by a CPU.
- Where only one CPU has a feature required for execution of a problem program, CPU affinity scheduling for that program can be specified at SYSGEN time. For example, an emulator might be installed on only one CPU in a Model 168 multiprocessing configuration. CPU affinity scheduling can be used to ensure that only the appropriate CPU is used to execute emulator-dependent job steps. CPU affinity scheduling is not required when both CPUs have installed the same emulator feature.

### ***Loosely-Coupled Configurations***

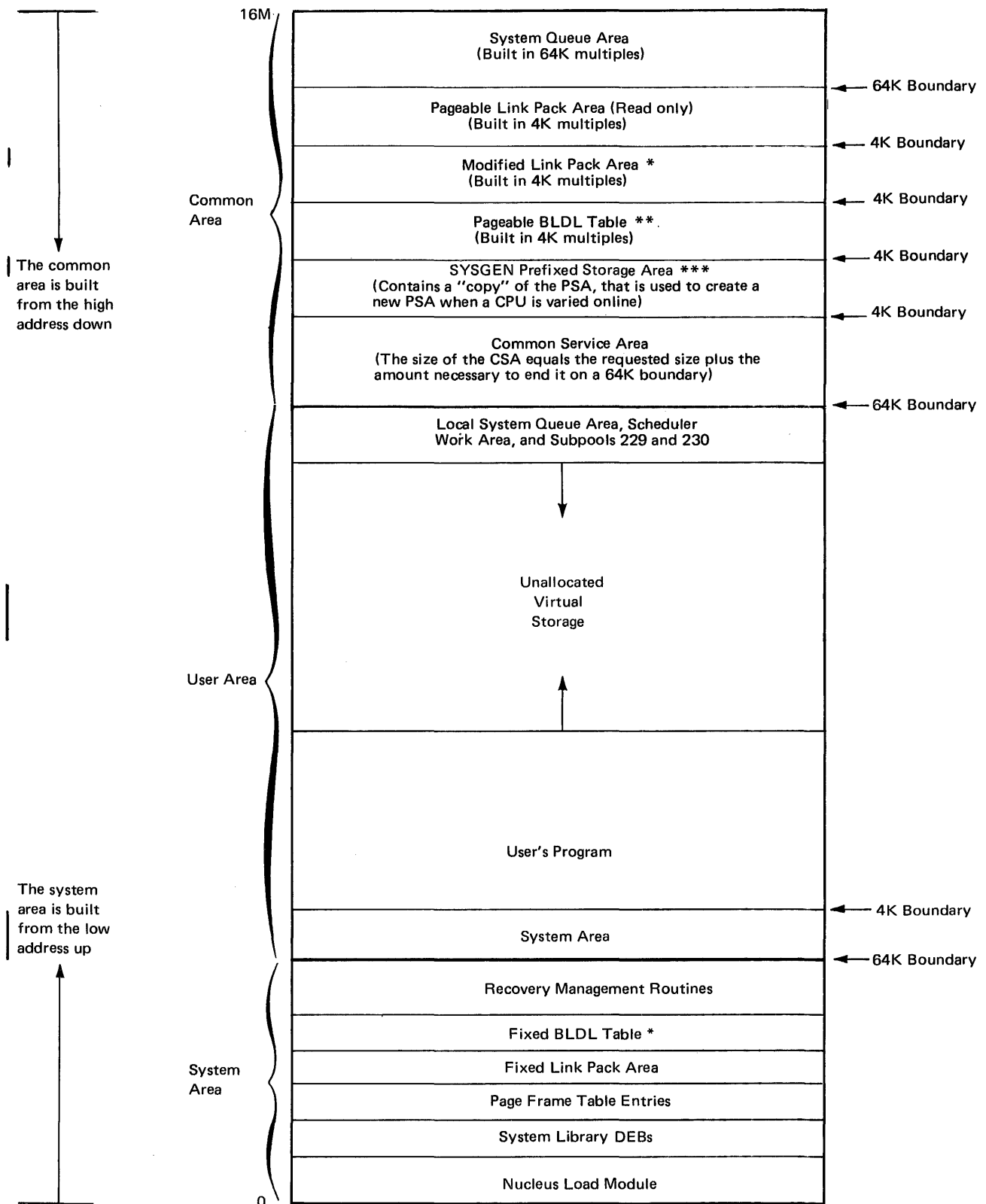
OS MVS supports up to four CPUs configured to share I/O devices. JES3 is the job entry subsystem that will manage the processing requirement of from one to four MVS uniprocessors or multiprocessors in any combination. JES3 executes in a processor called the Global system; the other attached MVS processors are Local systems. JES3 is also resident in the attached Local processors enabling dynamic global switching in the event of system failure. JES3 can attach ASP Version 3 Main Processors to a Global system. ASP Version 3 Main Processors can execute OS/MVT Release 21 or OS/VS2 Release 1.

Channel-to-channel connections are required from all processors to the Global system. Shared DASD must be accessible from all MVS JES3 processors for spool and checkpoint data sets.

Additional information on JES3 can be found in **Introduction to OS/VS2 Release 2**, GC28-0661 and in the JES3 section of this manual.



# Appendix A: Virtual Storage Map



\* The MLPA is optional.  
 \*\* The pageable and fixed BLDL tables are mutually exclusive.  
 \*\*\* The SYSGEN PSA is only for an MP system.

Figure 37. Virtual Storage Map



## Glossary

This glossary provides definitions of MVS terms. For definitions of terms not included, see IBM Data Processing Glossary, GC20-1699.

**ABP:** Actual block processor.

**ACR:** Alternate CPU recovery.

**actual block processor:** One of several programs that translates I/O requests into the proper format for the I/O supervisor.

**address space:** The virtual storage assigned to a batch or terminal job, a system task, or a task initiated by the START command. Each address space consists of the same range of addresses.

**address space identifier:** A unique, system-assigned identifier for an address space.

**address translation:** The process of changing the address of an item of data or an instruction from its virtual address to its real storage address. See also dynamic address translation.

**alternate CPU recovery:** a facility that attempts system recovery when a CPU fails by transferring work to another CPU.

**APAR:** Authorized program analysis report. A request for correction of a problem caused by a defect in a current unaltered release of a program. A PTF or corrected code is issued to the customer and the correction is incorporated into subsequent releases of the program.

**APF:** Authorized program facility.

**APG:** Automatic priority group.

**ASID:** Address space identifier.

**ASM:** Auxiliary storage manager.

**ASP:** See JES3.

**asymmetric I/O:** In a tightly coupled multiprocessing environment, an I/O device configuration in which any I/O device is physically attached to only one CPU.

**authorized library:** A library that can contain authorized programs.

**authorized program:** A system program or user program that is allowed to use restricted functions.

**authorized program facility:** A facility that permits the identification of programs that are authorized to use restricted functions.

**automatic priority group:** A group of jobs, each having its own address space at a single installation-specified priority level, that are dispatched according to a system resources manager algorithm that attempts to provide optimum use of CPU and I/O resources.

**auxiliary storage:** Data storage other than main storage.

**auxiliary storage management:** A facility that controls the allocation and release of pages on external page storage, and schedules I/O operations on the page data set.

**basic control (BC) mode:** A mode in which the features of a System/360 computing system and additional System/370 features, such as new machine instructions, are operational on a System/370 computing system. See also extended control (EC) mode.

**BC mode:** Basic control mode.

**block processor:** See actual block processor.

**change bit:** A bit associated with a page in real storage. The change bit is turned "on" by hardware whenever the associated page in real storage is modified.

**channel program translation:** In a channel program, replacement by software of virtual addresses with real addresses.

**common area:** The area of virtual storage that is addressable by all address spaces.

**common service area:** A part of the common area that contains data areas addressable by all address spaces, but protected during its use by the key of the requester. Abbreviated CSA.

**communication task:** A function that handles all communication between programs and operator consoles.

**control program keys:** Protection keys (0-7) that are reserved for control program use.

**control registers:** A set of registers used for operating system control of relocation, priority interruption, program event recording, error recovery, and masking operations.

**converter:** The routine that transforms spooled JCL statements into an internal format, and processes commands entered through the input stream.

**CPU affinity:** In a multiprocessing configuration, a function that allows a unit of work to be directed to a particular CPU for execution.

**cross memory services lock:** A global suspend lock that is used for services that apply to more than one private address space.

**CSA:** Common service area.

**DAT:** Dynamic address translation.

**deadline scheduling:** A feature allowing an installation to specify a time of day by which a job should be scheduled. The priority of the job is increased at defined intervals after this time.

**demand paging:** Transfer of a page from external page storage to real storage at the time it is needed for execution.

**dependent job control:** In JES3, a function that allows jobs to be executed or bypassed depending on the result of another job.

**device precedence list:** A list specifying by generic name, the order in which types of devices are to be preferred for allocation.

**disabled page fault:** A page fault that occurs when interruptions are disallowed by the CPU.

**DSP:** Dynamic support program.

**DSS:** Dynamic support system.

**dynamic address translation:** (1) The change of a virtual storage address to a real storage address during the execution of an instruction. (2) A hardware feature that performs the translation. Abbreviated DAT.

**dynamic area:** The portion of virtual storage that is divided into address spaces that are assigned to job steps and system tasks.

**dynamic support program:** In JES3, any process performed by the global processor as a part of a job in the system. Card reading and printing are examples.

**dynamic support system (DSS):** An interactive debugging facility that allows authorized personnel to monitor and analyze events and alter data.

**dynamic system interchange:** In JES3, allows the operator to switch the global processor functions to a local processor in the case of global processor failure.

**EC mode:** Extended control mode.

**enabled page fault:** A page fault that occurs when interruptions are allowed by the CPU.

**ESTAE/ESTAI exit routines:** Task recovery exit routines from which percolation to a higher-level exit may occur.

**execution service:** Provides the interface support between the job entry subsystems and MVS.

**extended control (EC) mode:** A mode in which all the features of a System/370 computing system, including dynamic address translation, are operational. See also basic control (BC) mode.

**external page storage:** The portion of auxiliary storage that is used to contain pages.

**external page table (XPT):** An extension of a page table that identifies the location on external page storage of each page in that page table.

**external slot:** See slot.

**fixed:** Not capable of being paged out. A fixed page in a private address area can be swapped out.

**fixed BLDL table:** A BLDL table that the user has specified to be fixed in the lower portion of real storage.

**fixed link pack area:** An extension of the link pack area that occupies fixed pages in the lower portion of real storage.

**fixed page:** a page in real storage that is not to be paged out.

**fragmentation:** The inability to assign real storage locations to virtual addresses because the available spaces are smaller than the page size.

**frame:** Same as page frame.

**FRR:** Functional recovery routine.

**functional recovery routine:** A recovery routine that is

used by locked programs, SRBs, and supervisor control routines.

**global:** Pertaining to more than one private address space.

**global lock:** A lock that protects serially reusable resources related to more than one private address space.

**global processor:** In JES3, the controlling processor of the configuration.

**global services:** Services that apply to more than one private address space. Also used when referring to multiple JES3 processors.

**graphics access method (GAM):** A facility that supports the 2250 Display Unit and the 2260 Display Station through the use of graphic programming services (GPS) and the graphic subroutine package (GSP).

**hardware error recovery management system:** A facility that attempts recovery from hardware malfunctions. It consists of the machine check handler (MCH) and the channel check handler (CCH).

**HASP:** See JES2.

**installation performance specification (IPS):** A set of installation-supplied control information used by the workload management routines. An IPS includes performance group definitions, performance objectives, workload level numbers and coefficients used to define the service rate.

**internal writer:** A facility in the job entry subsystem (JES2 or JES3) that allows user written output writers to write data on devices not directly supported by the job entry subsystem.

**internal reader:** A facility that transfers jobs to the job entry subsystem (JES2 or JES3).

**interaction:** The acceptance by a system of a line of input from a terminal, processing of the line, and a return of data, if any, to the terminal.

**interprocessor communication:** In a multiprocessing system, a function that provides communications between CPUs that are under control of the same control program.

**interval service value (ISV):** In the system resources manager, a category of information contained in a period definition, which specifies the minimum amount of service that an associated job will receive during any interval that the address space is in real storage.

**invalid page:** A page that cannot be directly addressed by the dynamic address translation feature of the central processing unit.

**IPC:** Interprocessor communication.

**IPS:** Installation performance specification.

**ISV:** Interval service value.

**JES2:** A functional extension of the HASP II program that receives jobs into the system and processes all output data produced by the job.

**JES3:** A functional extension of ASP Version 3, that performs job entry, scheduling, and output processing functions. It also provides loosely coupled multiprocessing support for up to four System/370 processors.

**job entry subsystem:** Performs functions analogous to the reader, writer, and SYS1.SYSJOBQE of previous systems. See JES2 and JES3.

**job journal:** A system data set used for restart by the job entry subsystem.

**LCH:** Logical channel queue.

**LGN:** Logical group number.

**link pack area (LPA):** An area of virtual storage containing selected reenterable and serially reusable routines that are loaded at IPL time and can be used concurrently by all tasks in the system.

**link pack update area:** An area in virtual storage containing modules that are additions to or replacements for link pack area modules.

**local lock:** A suspend lock that protects the resources assigned to a particular private address space.

**local processor:** In JES3, processors -- other than the global processor -- running MVS.

**local service:** A supervisory service that applies to only one private address space. Also used when referring to a single JES3 processor.

**local system queue area (LSQA):** One or more segments associated with each address space that contain system control blocks.

**lock:** A means of serialization used by supervisory control program routines. See global lock, local lock, spin lock, suspend lock.

**logical channel queue:** A queue of operations to be performed on a channel.

**logical group:** A collection of pages that are related to each other. An address space may be composed of multiple groups: one for the LPA, one for the SWA, and one for the private address space.

**logical group number:** An identifier of a logical group. It is the base value used by the real storage manager and the auxiliary storage manager to compute the identifier of a logical page. All the pages of a VIO data set may be represented by a single LGN.

**LPA:** Link pack area.

**LSQA:** Local system queue area.

**main processor:** In JES3, processors running OS ASP or VS2 Release 1 ASP.

**master address space:** The virtual storage used by the master scheduler task.

**MCP:** Message control program.

**Message control program:** A program that is used to control the sending or receiving of messages to or from remote terminals.

**MF/1:** System activity measurement facility.

**MIC:** Missing interruption checker.

**missing interruption checker:** A program that periodically tests active I/O operations in order to detect missing I/O interruptions and inform the operator of abnormal conditions before they can affect system operation.

**MP:** Multiprocessing

**multiple address space:** A feature that provides each user with a private address space.

**multiprocessing (MP) system:** A computing system employing two or more interconnected processing units to execute programs simultaneously.

**MVS:** Another name for OS/VS2 Release 2. Taken from the multiple virtual storage concept of this release.

**node:** An addressable point in a teleprocessing network.

**nonstandard job:** In JES3, a job with a JES3 control statement included in the JCL.

**page:** (1) A fixed-length block of instructions, data, or both, that can be transferred between real storage and external page storage. (2) To transfer instructions, data, or both, between real storage and external page storage.

**page data set:** A data set in external page storage, in which pages are stored.

**page fault:** A program interruption that occurs when a page that is marked not in real storage is referred to by an active page. Synonymous with page translation exception.

**page fixing:** Marking a page nonpageable so that it remains in real storage.

**page frame:** A block of real storage that can contain a page. Synonymous with frame.

**page-in:** The process of transferring a page from external page storage to real storage.

**page-out:** The process of transferring a page from real storage to external page storage.

**page reclamation:** The process of making addressable the contents of a page in real storage that has been marked invalid. Page reclamation can occur after a page fault.

**page stealing:** To take away an assigned page frame from a user to make it available for another purpose.

**page table (PGT):** A table that indicates whether a page is in real storage and correlates virtual addresses with real storage addresses.

**page translation exception:** A program interruption that occurs when a virtual address cannot be translated by the hardware because the invalid bit in the page table entry for that address is set. Synonymous with page fault. See also segment translation exception, translation specification exception.

**paging:** The process of transferring pages between real storage and external page storage.

**paging device:** A direct access storage device on which pages (and possibly other data) are stored.

**paging rate:** The average number of page-ins and page-outs per unit of time.

**parallel:** Pertaining to the simultaneity of two or more processes.

**PER:** Program event recording.

**percolation:** In error recovery, the passing of control to a higher-level recovery routine from another recovery routine along a preestablished path.

**performance group:** An installation-defined category that includes users with similar requirements and associates them with a performance specification.

**performance objective:** A category of information contained in an installation performance specification. Each performance objective specifies the service rate(s) that an associated job is to receive, for a number of different workload levels.

**period definition:** In the system resources manager, a category of information contained in a performance group definition, that indicates which performance objective is to be followed, either during a particular real-time period or until a particular amount of service has been accumulated by an associated job.

**PGT:** Page table.

**priority aging:** A feature that allows the priority of a job to be raised dynamically as a function of the number of times the job is eligible for execution, but bypassed.

**private address space:** An address space assigned to a particular user.

**program event recording (PER):** A hardware feature used to assist in debugging programs by detecting and recording program events.

**program management:** Routines that perform supervisory services related to the moving of programs from the system library or user libraries to virtual storage.

**quick cell facility:** A high-performance storage allocation technique using a fixed block size.

**RAS:** Reliability, availability, serviceability.

**real address:** The address of a location in real storage.

**real storage:** The storage of a System/370 computing system from which the central processing unit can directly obtain instructions and data, and to which it can directly return results.

**real storage management:** Routines that control the allocation of pages in real storage. Abbreviated RSM.

**reconfiguration:** The process of placing a CPU, main storage, or channel offline for testing; and adding or removing components while the system is in operation.

**recovery routine:** A routine that is entered when an error occurs during the performance of an associated operation. It isolates the error, assesses the extent of the error, indicates subsequent action, and attempts to correct the error and resume operation.

**recovery termination manager:** A program that handles

all normal and abnormal termination of tasks by passing control to a recovery routine associated with the terminated function.

**reference bit:** A bit associated with a page in real storage; the bit is turned "on" by hardware whenever the associated page in real storage is referred to. There is a reference bit in each of two storage keys associated with each page frame.

**refreshable:** A refreshable load module can be replaced by a new copy during its execution without changing either the sequence or results of processing.

**relocate hardware:** See dynamic address translation.

**relocation interrupt:** See page fault.

**remote terminal access method (RTAM):** A facility that controls operations between the job entry subsystem (JES2 or JES3) and remote terminals.

**response/throughput bias (RTB):** In the system resources manager, a category of information contained in a period definition, which indicates how the workload manager is to weigh trade-off between satisfying some system throughput objective, and satisfying the IPS-specified service rate.

**RMTGEN:** Generation of remote work stations for remote job entry.

**RSM:** Real storage manager.

**RTAM:** Remote terminal access method.

**RTB:** Response/throughput bias.

**scheduler work area (SWA):** One or more segments in virtual storage that contain most of the job management control blocks, such as the JCT, JFCB, SCT, and SIOT. There is one SWA for each initiator.

**secondary storage:** Auxiliary storage.

**segment:** A contiguous 64K area of virtual storage.

**segment table (SGT):** A table used in dynamic address translation to control user access to virtual storage segments. Each entry indicates the length, location, and availability, of a corresponding page table.

**segment translation exception:** A program interruption that occurs when a virtual address cannot be translated by the hardware because the invalid bit in the segment table entry for that address is set. Synonymous with program interruption code 16.

**service:** See service rate.

**service rate:** In the system resources manager, a measure of the rate that system resources (service) are provided to individual jobs. It is used by the installation to specify performance objectives, and used by the workload management routines to track the progress of individual jobs.

**service unit:** Same as service.

**slot:** A continuous area on a paging device in which a page can be stored.

**software recording facility:** A facility used by functional recovery routines (FRRs) to write system error records.

**spin lock:** A lock that prevents a CPU from doing work until the lock is cleared. Contrast with suspend lock.

**SQA:** System queue area.

**SRB:** Service request block.

**SRM:** System resources manager.

**suspend lock:** A lock that prevents requesters from doing work on a CPU, but allows the CPU to continue doing other work. Contrast with spin lock.

**SWA:** Scheduler work area.

**swapping:** A paging technique that writes the active pages of a job to external page storage, and reads pages of another job from external page storage into real storage.

**symmetric processors:** Processors with identical configurations.

**symmetric storage configurations:** Machine configurations with identical storage units.

**system activity measurement facility (MF/1):** A facility that collects information about system resource usage such as paging activity and the use of the CPU, channels, and I/O devices, to produce trace records and reports.

**system key:** A key that protects system data from damage or modification by unauthorized users.

**system queue area (SQA):** An area of virtual storage reserved for system-related control blocks. It contains fixed pages and is assigned protection key zero. Abbreviated SQA.

**system resources manager:** A group of programs that controls the use of system resources in order to satisfy the installation's performance objectives.

**thrashing:** A condition in which the system can do little useful work because of excessive paging.

**timer supervision:** Routines that provide the date and time of day, measure time intervals, schedule activities, and set the interval timer.

**TPIOS:** A facility that supports programmed telecommunications control units (TCUs) and generates channel programs for the channel scheduler.

**transaction:** In the system resources manager, a performance measurement made on an address space basis. It is a job or a job step for either batch or remote batch entry. It is a terminal interaction for time sharing (TSO) jobs.

**translation tables:** Page tables and segment tables.

**uniprocessing:** Sequential execution of instruction by a central processing unit or independent use of a CPU in a multiprocessing system.

**VIO:** Virtual I/O.

**virtual address:** An address that refers to virtual storage and must, therefore, be translated into a real storage address when used.

**virtual equals real address area (V=R):** An area of virtual storage that has the same range of addresses as real storage, and is used for a program or part of a program that is not to be paged or swapped during execution.

**virtual I/O (VIO):** A facility that pages data into and out of external page storage, although to the problem program, the data appears to be read from or written to direct access storage devices.

**virtual memory:** See virtual storage.

**virtual storage:** Addressable space that appears to the user as real storage, from which instructions and data are mapped into real storage locations. The size of virtual storage is limited by the addressing scheme of the computing system and by the amount of auxiliary storage available, rather than by the actual number of real storage locations.

**virtual storage access method (VSAM):** An access method for direct or sequential processing of fixed and variable length records on direct access devices.

**virtual storage management (VSM):** Routines that allocate address spaces and virtual storage areas within address spaces, and keep a record of free and allocated storage within each address space.

**VSAM:** Virtual storage access method.

**VSM:** Virtual storage management.

**V=R area:** See virtual equals real address area.

**workload manager:** A part of the system resources manager that allows an installation to determine the performance that any group of users will receive, monitors the workload, and schedules resources accordingly.

**XPT:** External page table.





Indexes to OS/VS2 publications are consolidated in the OS/VS2 Master Index, GC28-0693, and the OS/VS2 Master Index of Logic, GY28-0694. For additional information about any subject listed below, refer to other publications listed for the same subject in the Master Index.

- A operand
  - DISPLAY command 125
  - MONITOR command 126
  - TRACK command 127
- ABP (actual block processor) 155
- access method services 142
  - highlights 14
  - initializing system data sets 28
  - replacement for IEHLIST, IEHMOVE, IEHPROGM functions 143
  - used to create or update a VSAM catalog 142
  - verbs 142
- access methods
  - BTAM 23
  - GAM 23,156
  - ISAM 23
  - QTAM 15
  - RTAM 158
  - TCAM 23,13
  - VSAM 12,159
  - VTAM 12,23
- ACCOUNT command 135
- ACR (alternate CPU recovery) 151
  - definition 155
- actual block processor (ABP)
  - definition 155
- ADD subcommand 135
- address range 13
- address space
  - definition 155
  - master scheduler 38
  - multiple virtual 13
  - swapping by the system resources manager 54
  - time sharing user 38
- address space identifier (ASID)
  - definition 155
- address translation
  - definition 155
- ADDRSPC job control language parameter 117,113
- ADDRSPC=REAL (see virtual equals real address area) 146
- AFF job control language parameter 113
- affinity, CPU 151
  - definition 155
- AFFINITY system generation macro instruction 22,27
- ALGLIB system generation macro instruction 27
- ALGOL system generation macro instruction 27
- ALLOCATE command 135
- ALLOCATE subcommand 136
  - EDIT command 138
  - job entry subsystem support 138
- Allocation 121-123,12,14
  - highlights 13
  - philosophy 122
  - time sharing support 134
- ALTER verb 142
- alternate CPU recovery (ACR) 151
  - definition 155
- alternate path retry 150
- ALTSYS system initialization parameter 38
- AMBLIST service aid 15
- ANS cobol V2 17
- APAR
  - definition 155
  - for integrity exposures 79
  - for type I programs 17
- APF (authorized program facility) 85
  - authorized program library list 42
  - definition 155
  - SYSGEN specification with SVCTABLE macro 23
- APF system initialization parameter 32,31
- APG (automatic priority group)
  - definition 155
  - job step dispatching priority default value 117
  - system initialization parameter specification 32
  - system resources manager algorithm 56
- APG system initialization parameter 32,31
  - difference from VS2 Release 1 37
- appendage, installation-written 148,41
- APR 150
- ASB (automatic SYSIN batching) reader 15
- ASID (address space identifier)
  - definition 155
- ASM (auxiliary storage manager)
  - definition 155
- ASP 12
  - (see also JES3)
  - definition 155
- ASSEMBLER system generation macro instruction 27
- asymmetric I/O 149
  - definition 155
- ATTRIBUTE command 135
- authorized libraries 85
  - definition 155
  - duplicate module name restriction 87
  - routines used only for their intended purpose 83
  - specification in SYS1.PARMLIB list IEAAPFxx 42
- authorized appendage list (IEAAP00) 41-42
- authorized program facility (APF) 85
  - authorized program library list 42
  - definition 155
  - SYSGEN specification with SVCTABLE macro 23
- authorized programs 79,155
  - definition 155
  - requests to load other authorized programs 83
- authorized program facility 85
- authorized program library list (IEAAPFxx) 42
- authorized state 79
- authorized system programmer 81
- AUTO operand 126
- automatic priority group (APG)
  - definition 155
  - job step dispatching priority default value 117
  - system initialization parameter specification 32
  - system resources manager algorithm 56
- automatic SYSIN batching (ASB) reader 15
- auxiliary storage
  - definition 155
  - paging activity report of use 70
- auxiliary storage management
  - definition 155
- AUXLIST system initialization parameter 37
- BACKSPACE addressed to SYSIN/SYSOUT DCB 15
- basic control (BC) mode
  - definition 155
- basic telecommunications access method (BTAM) 23
  - online terminal test option 117
  - system generation specification 23
- BLDL system initialization parameter 32,31
- BLDLF system initialization parameter 32,31
- BRDCAST operand 126
- BRDR operand 125
- BRDRQ operand 125
- broadcast data set (SYS1.BROADCAST) 48
  - formatting 146
  - mounted for system initialization 138
  - new format description 49
- BTAM (basic telecommunications access method)
  - online terminal test option 117
  - system generation specification 23

CAMLST macro instruction 143  
 CANCEL command 125  
 CANCEL/STATUS command 136  
   job entry subsystem support 138  
 CANCEL subcommand 136  
 catalog (see VSAM catalog)  
 catalog connector 141  
 catalog integrity 87  
 catalog utility and macro changes 143  
 CENPROCS system generation macro instruction 29,27  
 CHAN MF/1 parameter 69  
 CHANGE subcommand 135  
 channel activity measurement 68,72  
 channel activity report 72  
   tuning example 73  
 channel-CPU overlap activity measurement 68  
 CHANNEL operand 127  
 channel program translation 155  
 CHANNEL system generation macro instruction 22,27  
 channel-to-channel connector 12,96  
 CHECKER system generation macro instruction 27  
 checkpoint/restart  
   checkpoint data sets 86  
   JES2 support 95  
 CHNGDUMP command  
   DEL 125  
   SET 125  
 CHRDR MF/1 parameter 69  
 CKPTREST system generation macro instruction 23,27  
 CLASS operand  
   MODIFY command 125  
   RESET command 126  
 CLPA system initialization parameter 32,31  
 CMD system initialization parameter 32,31  
 CN operand 126  
 CNVTCAT verb 142  
 COBOL F  
   compiler 17  
   library 17  
 COBLIB system generation macro instruction 27  
 COBOL system generation macro instruction 27  
 COMM MF/1 parameter 69  
 command library (SYS1.CMDLIB) 48  
   adding installation-written members 29  
 commands  
   operator 124-133  
   time sharing (TSO) 134-138  
 common area 153  
   definition 155  
 common service area (CSA) 120,153  
   definition 155  
   system initialization parameter specification 32  
 communication task  
   definition 155  
   initialization 30  
 concatenation  
   JOBLIB and STEPLIB integrity considerations 87  
   libraries containing APF-authorized programs 42  
   with SYS1.LINKLIB 43  
 configuration  
   minimum for multiprocessing 149  
   minimum for starter system 21  
 CONSOLE system generation macro instruction 23,27  
 control algorithm 56  
 CONTROL command  
   control function 56  
   control program extensions 85  
   control volume (see CVOL)  
 conversational remote job entry (CRJE) 15  
 conversion considerations 89-151  
 converter  
   definition 155  
 COPIES job control language parameter 115,113  
 counterfeit control block 81  
 CPQE system initialization parameter 37  
 CPU activity measurement 68,70  
 CPU activity report 70  
 CPU affinity 151  
   definition 155  
 CPU execution 57  
 CPU load balancing algorithm 54  
   weighted by resource factor coefficients 67,56  
 CPU MF/1 parameter 69  
 CPU operand 127  
 CPU resource factor coefficient 67  
 CPU service definition coefficient 66-67,57  
 CRJE (conversational remote job entry) 15  
 cross memory services lock  
   definition 155  
 CSA (common service area) 120,153  
   definition 155  
   system initialization parameter specification 32  
 CSA system initialization parameter 32,31  
 CTC 96  
 CTRLPROG system generation macro instruction 23,27  
 CVIO system initialization parameter 33,31  
 CVOL 144-146,141-143  
   integrity support 87  
 CYCLE MF/1 parameter 70  
  
 DASD MF/1 parameter 69  
 DAT (dynamic address translation) 155  
 data movement 120  
 data reduction routines 109  
 data sets  
   catalog conversion 139-146  
   concatenation 42,43  
   freeing 115  
   page 50-51  
   reclaim 52  
   system 47-50  
   SYS1.BROADCAST conversion 146  
   SYS1.UADS conversion 146  
   temporary 50,12  
 DATAMGT system generation macro instruction 23,27  
 DATASET system generation macro instruction 23,27  
   for adding installation-written members to system  
   libraries 28  
   for initializing new system data sets 28  
 DCB job control language parameter 113  
 DCMLIB system generation macro instruction 27  
 DD statement  
   difference from MVT 113  
   difference from VS2 Release 1 113  
   maximum specification 116  
 DDR 150-151  
 deadline scheduling 98,102  
 dedicated work files 15  
 DEFINE verb 142  
 DEL operand 125  
 DELETE subcommand 136  
 DELETE verb 142  
 dependencies, compatibility 148  
 dependent job control (DJC) 98,102  
 DEQ macro instruction  
   ensuring serial access for multiprocessing 151  
   major name restrictions for authorized programs 83  
 DEST job control language parameter 115,113  
 device allocation algorithm 56  
 device list functions of JES2 commands 131  
 DEVICE MF/1 parameter 69  
 device precedence list 122,123  
 diagnostic support 127  
 direct system output (DSO) 15  
 disabled page fault  
   definition 156  
 DISPLAY command 125,128  
 DISPLAY subcommand 136  
   job entry subsystem support 138  
 DJC 98,102  
 DPRTY job control language parameter 117,113  
 DRIVER operand 126  
 driver, TSO 16  
 DSI 96  
 DSO (direct system output) 15  
 DSP 98  
 DSS (dynamic support system)  
   definition 156  
   NUCMAP system initialization parameter 34  
   system data set SYS1.DSSVM 47,49,50  
 dump data set (SYS1.DUMP) 48  
   device specification by a system initialization

- parameter 33
- maximum number specified 36
- dump default lists
  - IEAABDxx 42
  - IEADMPxx 42
  - overriding defaults with the CHNGDUMP command 128,125
- DUMP system initialization parameter 33,31
  - difference from VS2 Release 1 36
- DYNAM job control language parameter 113,115
- dynamic address translation (DAT)
  - definition 156
- dynamic allocation validation exit 87
- dynamic device reconfiguration (DDR) 150-151
- dynamic support program (DSP) 98
- dynamic support system (DSS)
  - definition 156
  - NUCMAP system initialization parameter 34
  - system data set SYS1.DSSVM 47,49,50
- dynamic system interchange (DSI) 96
- DYNAMNBR job control language parameter 115,113
  
- EC (extended control) mode
  - definition 156
- EDIT command 136
- EDIT system generation macro instruction 23,27
- EDITOR system generation macro instruction 29,27
- emulator 151
  - specified by AFFINITY macro instruction 22
- EMULATOR system generation macro instruction 27
- enabled page fault
  - definition 156
- END subcommand 136
- ENQ macro instruction
  - ensuring serial access for multiprocessing 151
  - major name restrictions for authorized programs 83
- ENQ/DEQ algorithm 56
- enqueue residence value (ERV) 67,56
- EOD operand 125
- EROPT JCL subparameter of DCB parameter 117,113
- error recording data set (SYS1.LOGREC) 47
  - change from MVT and VS2 Release 1 49
- ERV (enqueue residence value) 67,56
- EXCP addressed to SYSIN/SYSOUT DCB 15,16
- EXEC command 136
- EXEC statement 113
- execution service 92
- execution task monitor 92
- EXPORT verb 142
- extended control (EC) mode 156
- external page storage
  - definition 156
  - MF/1 counts 71,70
- external page table (XPT)
  - definition 156
- external writer 94,116
  
- fetch protection 81,87
- FIB (foreground initiated background) commands 138
- file protection attributes 86
- FIX system initialization parameter 33,31
- fixed
  - definition 156
- fixed BLDL table 153
  - definition 156
  - system initialization parameter specification 32
- fixed link pack area (LPA) 153
  - definition 156
  - specifying routines with the FIX parameter 33
- fixed LPA list (IEAFIXxx) 42
- fixed page 147
  - definition 156
- foreground initiated background (FIB) commands 138
- FORTLIB system generation macro instruction 27
- FORTRAN G 17
- FORTRAN H 17
- FORTRAN system generation macro instruction 27
- fragmentation
  - definition 156
- virtual storage 13
- FREE command 136
  - job entry subsystem support 138
- FREE job control language parameter 115,113
- freeing data sets 115
- FRID job control language subparameter of DCB parameter 116,113
- Full ANS COBOL V2 17
- functions not supported
  - MVT 15
  - VS2 Release 1 16
  
- GAM (graphics access method)
  - definition 156
  - system generation specification 23
- GDG (generation data group) 142
- generalized trace facility (GTF) 127
  - replacement for trace and TSO trace 16
  - TRACE system initialization parameter 37
- GENERATE system generation macro instruction 23,27
- generation data group (GDG) 142
- GENTSO system generation macro instruction 27
- GETMAIN, restriction for variable-length 116
- GJOBCTL system generation macro instruction 27
- GJP (graphic job processor) 15
- global
  - definition 156
- global lock
  - definition 156
- global processor 96
- global services
  - definition 156
- GRAPH MF/1 parameter 69
- graphic job processor (GJP) 15
- graphics access method (GAM)
  - definition 156
  - system generation specification 23
- GRAPHICS system generation macro instruction 29,27
- GTF (generalized trace facility) 127
  - replacement for trace and TSO trace 16
  - TRACE system initialization parameter 37
  
- HALT command 125,127
- hard copy log 33
- HARDCPY system initialization parameter 33,27
- hardware error recovery management system
  - definition 156
- HASP 156
  - (see also JES2)
- HELP system generation macro instruction 27
- HIARCHY job control language subparameter of DCB parameter 113
- hierarchy support
  - unsupported MVT function 15,38
- highlights of VS2 Release 2 12
- HOLD command 125
- HOLD operand 126
- HRAM system initialization parameter 38
- HSVC system initialization parameter 38
  
- IBM-created SYS1.PARMLIB lists 41-44
- IEAABDxx (dump default list)
  - description 42
  - overriding defaults with the CHNGDUMP command 128,125
- IEAAPFxx (authorized program library list) 42
- IEAAPPO0 (authorized appendage list) 41
- IEABLDxx (resident BLDL list) 43
- IEADMPxx (dump default list)
  - description 42
  - overriding defaults with the CHNGDUMP command 128,125
- IEAFIXxx (fixed LPA list) 42
- IEAIPsxx (see installation performance specification)
- IEALOD00 (LPA directory load list) 43
- IEALPaxx (modified link pack area list) 45
- IEAOPTxx (system resources management tuning parameter list) 67,45-46

IEAPAK00 (LPA packing list) 43  
IEASY5xx (system parameter list)  
description 44  
recommendations for values 39  
IEBUPDAT utility program 15  
IEBUPDTE utility program 15  
IEFACTRT SMF exit 111  
IEFUIV SMF exit 111  
IEFUJI SMF exit 111  
IEFUJP SMF exit 111  
IEFUJV SMF exit 111  
IEFUSI SMF exit 111  
IEFUSO SMF exit 111  
IEFUTL SMF exit 111  
IEFU83 SMF exit 111  
IEHIOSUP utility program 15  
IEHLIST utility program 143,15,16  
IEHMOVE utility program 143,15,16  
IEHPROGM utility program 143,28,15,16  
IEHUCAT utility program 143  
IMAGELIB system generation macro instruction 27  
IMBMDMAP program 15  
IMCOSJQD service aid 16  
IMPORT verb 142  
IN operand 125  
indexed sequential access method (ISAM) 23  
INIT operand  
DISPLAY command 125  
TRACK command 127  
initial program loader (IPL) 30  
initialization, master scheduler 30  
initialization, new system data sets 28  
initialization, system 30-46  
parameters 30-39  
input service 91  
installation-created SYS1.PARMLIB lists 45-46  
installation performance specification (IPS) 58  
definition 156  
IEAIPSxx list 65-67,42  
parameters 65-67  
tuning example 73-77  
using SMF data to adjust 109  
installation verification procedure (IVP) 20,21  
integrity, system 79-87  
areas of concern 81-85  
definition 79  
installation responsibility 80  
procedural responsibilities 86-87  
interaction 156  
internal reader 91  
definition 156  
internal writer  
definition 156  
interprocessor communication (IPC)  
definition 156  
interregion communication 120  
INTERVAL MF/1 parameter 69  
interval service value (ISV) 66  
definition 156  
IOC resource factor coefficient 67  
IOC service definition coefficient 67,57  
IOCONTRL system generation macro instruction 27  
IODEVICE system generation macro instruction 23,27  
I/O appendage integrity 87  
I/O device activity measurement 68  
I/O device activity report 72  
tuning example 73  
I/O load balancing algorithm 54  
weighted by resource factor coefficients 67,56  
I/O measure 57  
I/O, Virtual 51-52  
IPC (interprocessor communication)  
definition 156  
IPL (initial program loader) 30  
IPS (see installation performance specification) 58  
IPS operand 126  
IPS system initialization parameter 33,31  
IQAORDER list 40  
IRBMF1xx (MF/1 parameter list) 43  
ISAM (indexed sequential access method) 23  
ISV (interval service value) 66

IVP (installation verification procedure) 20,21

JCL (job control language) 112-118  
(see also parameters, JCL)  
changes to MVT 117,113  
changes to VS2 Release 1 114-117  
for JES2 117-118  
summary of changes 113  
JCL validation exit 87

JES2  
definition 156  
differences from MVT/HASP and VS2  
Release 1/HASP 93-95  
generation 24-25  
highlights 12  
input service 91  
JCL 117-118  
job flow 91  
operator commands 124-131  
SMF record support 110  
time sharing commands 138  
JES2 control cards 117-118  
JES2 generation 24-25  
JES3 96-105  
commands 132-133,130  
comparison with ASP 104-105  
definition 156  
generation 25,24  
highlights 12  
JCL 118  
job flow 99  
multiprocessing 152  
job class, JES2 25  
job control language (JCL) 112-118  
changes to MVT 117,113  
changes to VS2 Release 1 114-116  
for JES2 117-118  
summary of changes 113  
job entry subsystem (see JES2, JES3)  
data set 47,49  
generation 20-21,23-25  
highlights 12  
job initiation record 106  
job journal 93  
job purge SMF exit 111  
job queue data set (SYS1.SYSJOBQE) 49  
unsupported MVT function 16  
unsupported VS2 Release 1 function 16  
job segment 98  
job selection priority 117  
JOB statement 113  
job step dispatching priority default value 117  
job summary record (SMF) 110  
JOBCAT catalog 141  
JOBCLASS operand 125  
jobname operand 125-126  
JOBPARM control card 118  
JOBS operand  
DISPLAY command 125  
TRACK command 127  
journal, job 93

link library (SYS1.LINKLIB) 47  
adding installation-written members 29  
link library list (LNKLSTxx) 43  
link pack area (LPA)  
defined 157  
fixed LPA 153  
modified LPA 153  
pageable LPA 153  
link pack area library (SYS1.LPALIB) 47  
adding installation-written members 29  
LINKLIB system generation macro instruction 29,27  
LISTCAT verb 142  
LNK system initialization parameter 33,31  
LNKLSTxx (link library list) 43  
load balancing 54  
LOADER system generation macro instruction 29,27  
local lock  
definition 157

local main processor 96  
 local processor 96  
 local service  
   definition 157  
 local system queue area (LSQA) 153  
   definition 157  
 local time constant (PARMTZ) 45  
 locking technique 83,149  
 log data sets 50  
   specification of output class with LOGCLS 33  
   specification of WTL limit with LOGLMT 33  
 LOGCLS system initialization parameter 33,27  
 LOGLMT system initialization parameter 33,27  
 LOGOFF command 136  
 LOGON command 136  
   system resources manager support 134  
 loosely coupled multiprocessing (MP) 152,12  
 LPA (link pack area)  
   definition 157  
   fixed LPA 153  
   modified LPA 153  
   pageable LPA 153  
 LPA directory load list (IEALOD00) 43  
 LPA packing list (IEAPAK00) 43  
 LSQA (local system queue area) 147  
   definition 157  
 LSQA operand  
   MOUNT command 128,126  
   START command 128,126  
 LSQACEL system initialization parameter 37  
  
 M operand 125  
 MACLIB system generation macro instruction 29,27  
 macro instructions, system generation  
   AFFINITY 22,27  
   ALGLIB 27  
   ALGOL 27  
   ASSEMBLER 27  
   CENPROCS 29,27  
   CHANNEL 22,27  
   CHECKER 27  
   CKPTREST 23,27  
   COBLIB 27  
   COBOL 27  
   CONSOLE 23,27  
   CTRLPROG 23,27  
   DATAMGT 23,27  
   DATASET 23,27  
   DCMLIB 27  
   EDIT 23,27  
   EDITOR 29,27  
   EMULATOR 27  
   FORTLIB 27  
   FORTRAN 27  
   GENERATE 23,27  
   GENTSO 27  
   GJOBCTL 27  
   GRAPHICS 29,27  
   HELP 27  
   IMAGELIB 27  
   IOCONTRL 27  
   IODEVICE 23,27  
   LINKLIB 29,27  
   LOADER 29,27  
   MACLIB 29,27  
   OUTPUT 27  
   PAGE 29,27  
   PARMLIB 29,27  
   PL1 27  
   PL1LIB 27  
   PTOP 27  
   RESMODS 29,27  
   RPG 27  
   SECONSLE 29,27  
   SCHEDULR 23,27  
   SORTLIB 27  
   SORTMERG 27  
   SUPRVSOR 27  
   SVCTABLE 23,27  
  
   SYSUTILS 27  
   TELCMLIB 27  
   TSO 23,27  
   TSOPTION 27  
   UCS 29,27  
   UNITNAME 23,27  
 macro library (SYS1.MACLIB) 47  
   adding installation-written members 29  
 main processor 96  
 main storage (see real storage)  
 main storage hierarchy support 15  
 main storage occupancy algorithm 55  
 manual switching 151  
 master address space 37,38  
   definition 157  
 master scheduler initialization 30  
 MATRIX operand of DISPLAY command 127,125  
 MAXUSER system initialization parameter 34,31  
 MCP (message control program)  
   definition 157  
   TCAM 148  
 measurement facilities  
   MF/1 68-77  
   SMF 106-111  
 measurement facility control (MFC) 68  
 MEMBER MF/1 parameter 70  
 MEMBERS system generation parameter 28  
 message control program (MCP)  
   definition 157  
   TCAM 148  
 MFC (measurement facility control) 68  
 MF/1 (system activity measurement facility) 68-77  
   definition 157  
   highlights 13  
   operation 68-69  
   parameters 69-70  
   reports 70-72  
   SYS1.PARMLIB member 43  
   tracing 72  
   tuning example 73-77  
 MF/1 parameter list (IRBMF1xx) 43  
 MF/1 reports 70-72  
   channel activity 72  
   CPU activity 70  
   I/O device activity 72  
   paging activity 70-71  
   workload activity 71-72  
 MIC (missing interruption checker)  
   definition 157  
   initialization 30  
 MIN system initialization parameter 37  
 minimum configuration  
   for multiprocessing 149  
   for starter system 21  
 missing interruption checker (MIC)  
   definition 157  
   initialization 30  
 MLPA system initialization parameter 34,31  
 MN operand 126  
 MOD system initialization parameter 38  
 modified link pack area 153  
   system initialization parameter specification 33  
 modified link pack area list (IEALPaxx) 45  
 MODIFY command 125  
   MODIFY subcommand 136  
 MONITOR command 128,126  
   MONITOR subcommand 136  
 MOUNT command 126,128  
 MP (see multiprocessing)  
 MPA system initialization parameter 37  
 MPS system initialization parameter 38  
 MSGRT command 126  
 multiple address spaces 13  
   definition 157  
   specifying maximum number with MAXUSER  
   parameter 34  
 multiple locks 149  
 multiple virtual address spaces 13  
   definition 157  
   specifying maximum number with MAXUSER  
   parameter 34

multiprocessing (MP), tightly coupled  
   considerations for VS2 Release 2 149-151  
   definition 157  
   highlights 12  
   operator command support 127  
   SMF record support 110  
   system integrity considerations 82  
 multiprocessing (MP), loosely coupled 12  
 multiprogramming 13  
 multitasking  
   considerations for a multiprocessing  
   environment 82,151

**N operand 125**  
**NET operand 127**  
   DISPLAY command 125  
   HALT command 127  
   MODIFY command 126  
   START command 126  
   VARY command 127  
 Network job processing (NJP) 98,102  
 NIP (nucleus initialization program) 30  
 NJP 98,102  
 NOCHAN MF/1 parameter 69  
 NOCHRDR MF/1 parameter 69  
 NOCOMM MF/1 parameter 69  
 NOCPU MF/1 parameter 69  
 NODASD MF/1 parameter 69  
 NODEVICE MF/1 parameter 69  
 NOGRAPH MF/1 parameter 69  
 nonpageable dynamic area 146-147  
 nonstandard job 100  
 non-standard control program interface 81  
 not-in-use data sets 115  
 NOTE addressed to SYSIN/SYSOUT DCB 15,16  
 NOOPTIONS MF/1 parameter 70  
 NOPAGING MF/1 parameter 69  
 NORECORD MF/1 parameter 69  
 NOREPORT MF/1 parameter 69  
 NOTAPE MF/1 parameter 69  
 NOTRACE MF/1 parameter 69  
 NOUNITR MF/1 parameter 69  
 NOWKLD MF/1 parameter 69  
 nucleus 153  
 nucleus initialization program (NIP) 30  
 nucleus library (SYS1.NUCLEUS) 47  
   adding installation-written members 29  
 nucleus-only generation 20,15  
 NUCMAP system initialization parameter 34,31

**OPER operand 126**  
**operator**  
   commands 124-131  
   console, JES3 103  
   control of multiprocessing configurations 149-151  
   reducing interaction during system initialization 39  
**OPERATOR command 136**  
**operator commands 124-131**  
   job entry subsystem support 128-131  
   multiprocessing support 127  
   summary of changes 125-127  
   system resources management support 128  
   time sharing support 128  
   VTAM support 127  
 operator intervention restriction 34,31  
 OPI system initialization parameter 34,31  
 OPT system initialization parameter 34,31  
 OPTIONS MF/1 parameter 70  
 ordered seek queueing 16  
**OUT operand**  
   CANCEL command 125  
   RESET command 126  
**OUTCLASS operand 125**  
**OUTLIM job control language parameter 116,113**  
**OUTPUT command 137**  
   job entry subsystem support 138  
**OUTPUT control card 118**  
 output limit specification 116  
 output record (SMF) 110

output service 93  
**OUTPUT system generation macro instruction 27**

**page**  
   definition 157  
**page data sets 47**  
   adding paging space to the system 50  
   change from VS2 Release 1 49  
   definition 157  
   highlights 14  
   naming (PAGE parameter) 34  
**page fault**  
   definition 157  
**page fixing**  
   definition 157  
**page frame**  
   definition 157  
   MF/1 counts 71  
**page frame table entries 153**  
**page-in**  
   counts in SMF records 106  
   definition 157  
   rates and percentages in MF/1 reports 71  
**page-out**  
   counts in SMF records 106  
   definition 157  
   rates and percentages in MF/1 reports 71  
**page reclamation**  
   definition 157  
   rates and percentages in MF/1 reports 71  
**page replacement algorithm 55**  
**page stealing 55**  
   definition 157  
**PAGE system generation macro instruction 29,27**  
**PAGE system initialization parameter 34,31**  
   adding paging space to the system 50  
   difference from VS2 Release 1 36  
**page table (PGT)**  
   definition 157  
**page translation exception**  
   definition 157  
**pageable BLDL table 153**  
   system initialization parameter specification 32  
**pageable link pack area 153**  
   creation using the CLPA parameter 32  
   SYS1.LPALIB 47  
   transient area replacement 16  
**pageable real storage counts 71**  
**paging**  
   definition 157  
   replacement for rollout/rollin 15  
   replacement for scatter load 15  
   space 50  
**paging activity measurement 68,70-71**  
**paging activity report 70-71**  
**paging device**  
   definition 157  
**PAGING MF/1 parameter 69**  
**paging rate**  
   MF/1 counts 71  
   definition 158  
**paging space (page data sets) 47,50**  
   adding paging space to the system 50  
   change from VS2 release 1 49  
   definition 158  
   highlights 14  
   naming (PAGE parameter) 34  
**PAL system initialization parameter 37**  
**parameters, JCL**  
   ADDRSPC 116-117,113  
   AFF 113  
   COPIES 115,113  
   DCB 113  
   DEST 115,113  
   DPRTY 117,113  
   DYNAM 113,115  
   DYNAMNBR 115,113  
   FREE 115,113  
   HIARCHY 113  
   OUTLIM 116,113

**PERFORM** 113,114  
**PRTY** 117-118,114,113  
**REGION** 116,113  
**RETAIN** 113  
**ROLL** 113  
**SEP** 113  
**SPACE** 113  
**SPLIT** 113  
**SUBALLOC** 113  
**SYSABEND** 42  
**SYSUDUMP** 42  
**UNIT** 113  
parameters, MF/1 69-70  
  example 73-75  
parameters, system initialization  
  **ALTSYS** 38  
  **APF** 32,31  
  **APG** 32,31  
  **AUXLIST** 37  
  **BLDL** 32,31  
  **BLDLF** 32,31  
  **CLPA** 32,31  
  **CMD** 32,31  
  **CPQE** 37  
  **CSA** 32,31  
  **CVIO** 33,31  
  **DUMP** 33,31  
  **FIX** 33,31  
  **HARDCPY** 33,31  
  **HRAM** 38  
  **HSVC** 38  
  **IPS** 33,31  
  **LNK** 33,31  
  **LOGCLS** 33,31  
  **LOGLMT** 33,31  
  **LSQACEL** 37  
  **MAXUSER** 34,31  
  **MIN** 37  
  **MLPA** 34,31  
  **MOD** 38  
  **MPA** 37  
  **MPS** 38  
  **NUCMAP** 34,31  
  **OPI** 34,31  
  **OPT** 34,31  
  **PAGE** 34,31  
  **PAL** 37  
  **QBF** 38  
  **RAM** 38  
  **REAL** 34,31  
  **RERP** 38  
  **RSVC** 38  
  **SMF** 35,31  
  **SQA** 35,31  
  **SQACEL** 37  
  **SQS** 38  
  **SYSP** 35,31  
  **TMSL** 37,38  
  **TRACE** 37  
  **TSOAUX** 37  
  **VAL** 36,31  
  **VRREGN** 36,31  
  **WTOBFRS** 36,31  
  **WTORPLY** 36,31  
**PARMLIB** system generation macro instruction 29,27  
**PARMTZ** (local time constant) 45  
password protection  
  integrity 87  
  **PASSWORD** data set 48  
  system data sets 86,80  
  system libraries 86  
  tape data sets 86-87  
**PATH** operand 127  
**PAUSE** operand 125  
**PDS** system generation parameter 28  
percent channel active value 72  
percent channel busy and CPU waiting value 72  
**PERFORM** job control language parameter 113,114  
performance groups 158  
performance group number 66  
  **JCL** specification 114  
  time sharing **LOGON** command 134  
performance group periods 158  
  definition 158  
  tuning example 73-77  
performance group period report 71  
performance group report 71-72  
performance objective number 67,59-65  
performance objectives 59-65  
  definition 158  
  effect of changing 76  
  motive for specifying 59  
  tuning example 73-77  
performance specifications 58  
performance variable 57  
period duration parameter 66  
**PGT** (page table)  
  definition 158  
physical addresses for devices 21  
physical environment, computing system 80  
**PL1** system generation macro instruction 27  
**PL/1 F**  
  **compiler** 17  
  **library** 17  
  **syntax checker** 17  
**PLILIB** system generation macro instruction 27  
**POINT** addressed to **SYSIN/SYSOUT DCB** 15,16  
preexecution setup 98,100  
pre-fixed storage area (PSA) 153  
**PRESRES** list 46,40  
**PRINT** verb 142  
priority aging 101  
priority I/O queueing 16  
priority, job selection 117  
priority, job step dispatching 117  
private address space  
  definition 158  
  size recorded by **SMF** 109  
**PROC** operand 126  
procedure library (**SYS1.PROCLIB**) 48  
  adding installation-written members 29  
processor-only generation 20  
procname operand 125  
**PROFILE** subcommand 136  
  **EDIT** command 138  
program conversion 148  
programming conventions 148  
  **V=R** 146-147  
  multiprocessing 151-152  
protected system control block 81-82  
**PRTY** job control language parameter 116,114,113  
**PRTY** operand 126  
PSA (pre-fixed storage area) 153  
pseudo device 96  
**PTOP** system generation macro instruction 27  
  
**Q** operand  
  **HOLD** command 125  
  **RELEASE** command 126  
  **SET** command 126  
**QBF** system initialization parameter 38  
queued telecommunications access method (**QTAM**) 15  
**QUIESCE** command 127,126  
**QTAM** (queued telecommunications access method) 15  
  
**R** operand 125  
**RAM** system initialization parameter 38  
reader 91,156,15,16  
  internal 156  
  **OS** 15,16  
real address  
  definition 158  
  range set for multiprocessing 149-151  
real main processor 96  
real storage  
  **ADDRSPC=REAL** 148  
  definition 158  
  multiprocessing considerations 149-151  
  **MVT** 13  
  paging activity report of use 70-71  
  shared 12

real storage management (RSM)  
   definition 158  
 real storage occupancy  
   resource-use algorithm 54-55  
   service definition coefficient MSO 67  
   service received during interval of 67  
 REAL system initialization parameter 34,31  
   difference from VS2 Release 1 37  
 Reclaim (of data set) 52  
 reconfiguration 149-151  
   definition 158  
   JES3 104  
 RECORD MF/1 parameter 69  
 recovery  
   definition 158  
   multiprocessing 150  
 recovery management routines 153  
 recovery termination manager  
   definition 158  
   initialization 30  
 refreshable  
   definition 158  
   modules in the pageable link pack area 47  
 REGION job control language parameter 116,113  
 REGSIZE operand 126  
 RELEASE command 126  
 relocate hardware (dynamic address translation)  
   definition 155  
 remote job entry (RJE) 15  
 remote terminal access method (RTAM)  
   definition 158  
 remote terminal programs 25  
 REPORT MF/1 parameter 69  
 REPRO verb 142  
 RERP system initialization parameter 38  
 RESET command 126  
 resident BLDL list (IEABLDxx) 43,44  
 RESMODS system generation macro instruction 29,27  
 resource factor coefficients 67  
   use of 56  
 resource identification 83  
 resource manager constants 67  
 resource-use routines 54-55  
 response/throughput bias (RTB) 66  
   definition 158  
 RETAIN job control language parameter 113  
 RJE (remote job entry) 15  
 RMTGEN 25  
   definition 158  
   options 25  
 ROLL job control language parameter 113  
 rollout/rollin 15  
 RPG system generation macro instruction 27  
 RSM (real storage management)  
   definition 158  
 RSVC system initialization parameter 38  
 RTAM (remote terminal access method)  
   definition 158  
 RTB (response/throughput bias) 66  
   definition 158  
 RUN subcommand 136  
  
 SAMG (system activity measurement gathering) 68  
 SARG (system activity report generation) 68  
 satellite graphic job processor (SGJP) 15  
 SAVE operand 126  
 scatter load 15  
 scheduler work area (SWA) 153  
   definition 158  
   SYS1.SYSJOBQE replacement 16,49  
 SCHEDULR system generation macro instruction 23,27  
 secondary storage (auxiliary storage)  
   definition 155  
   paging activity report of use 70-71  
 SECONDSLE system generation macro instruction 29,27  
 security, system 86-87  
 segment  
   definition 158  
 segment table (SGT)  
   definition 158  
  
 SEND operator command 128,126  
 SEND subcommand  
   EDIT command 138,136  
   OPERATOR command 137  
 SEND time sharing command 137  
 SEP job control language parameter 113  
 separator page 93  
 serialization, allocation 121,13  
 serialization techniques 82-83  
   disablement 82  
   ENQ/DEQ 83,151  
   locking 83  
   WAIT/POST 151  
   service 57  
 service definition coefficients 67,65  
   use of 57-59  
 service rate 59-65  
   definition 158  
   effect of change for a performance objective 76  
   IPS specification 67  
   tuning example 73-77  
   workload activity reports 71-72  
 service units 57-65  
   definition 158  
   MF/1 workload activity reports 71-72  
   SMF record counts 106  
   tuning example 73-77  
 SER0 routine 15  
 SER1 routine 15  
 SET command  
   used to identify the CPU with an acceptable clock  
   value 127  
   used to restore time zone constant to IPL value 127  
   used to specify an IPS 126  
 SET operand 125  
 setup  
   JES2 96  
   JES3 98,100  
 SGJP (satellite graphic job processor) 15  
 SGT (segment table)  
   definition 158  
 shared DASD, recovery of 150  
 SIO instruction counts 72  
 slot  
   definition 158  
   MF/1 counts 71  
 SMF (system management facilities) 106-111  
   differences from MVT 106  
   differences from VS2 Release 1 106  
   exits 111  
   records 107-108  
   system initialization parameter specification 35  
 SMF data sets (SYS1.MANX/SYS1.MANY) 48  
   change from MVT 50  
   change from VS2 Release 1 49  
 SMF operand 126  
 SMF parameter list (SMFPRMxx) 44  
 SMF system initialization parameter 35,31  
 SMFDEFLT parameter list 44,40  
 SMFPRMxx (SMF parameter list) 44  
 software recording facility  
   definition 158  
 SORT/MERGE 17  
 SORTLIB system generation macro instruction 27  
 SORTMERG system generation macro instruction 27  
 SPACE job control language parameter 113  
 SPACE system generation macro parameter 28  
 spin lock  
   definition 159  
 spinoff 93,94  
 SPLIT job control language parameter 113  
 spool 99  
 SQSA (system queue area) 153  
   definition 159  
   system initialization parameter specification 35  
 SQA operand 125,147  
 SQS system initialization parameter 35,31  
 SQACEL system initialization parameter 37  
 SQA system initialization parameter 38  
 SSM instruction 148



stage I 20  
stage II 20  
standard control program interface 81  
START command 128,126  
START operand 127  
starter system 21  
    minimum configuration 21  
    physical addresses for devices 21-22  
STEPCAT catalog 141  
STIDP instruction 38  
STOP command 128  
STOP MF/1 parameter 70  
STOPMN subcommand 136  
STOPTR command 128,127  
STORAGE operand 127  
storage map, virtual 153  
store protection 87  
SUBALLOC job control language parameter 113  
SUBMIT command 137  
    job entry subsystem support 138  
SUBMIT operand 126  
support processor 96  
suspend lock  
    definition 159  
SUPRVSOR system generation macro instruction 27  
SVC library (SYS1.SVCLIB) 47  
    adding installation-written members 29  
SVC routines calling SVC routines 84  
SVC routines, installation-written  
    integrity considerations 85  
    SYSGEN specification with SVCTABLE macro 23  
SVCTABLE system generation macro instruction 23,27  
SWA (scheduler work area) 153  
    definition 159  
    replacement for SYS1.SYSJOBQE 16,49  
swap counts 71  
swapping  
    definition 159  
    ERV effect on 67  
    ISV effect on 66  
    use by the system resources manager 57  
switching, manual 151  
symmetric processors  
    definition 159  
symmetric storage configuration  
    definition 159  
symmetrical I/O 149  
SYNC subcommand 146,135  
SYSABEND job control language parameter 42  
SYSCTLG data set 49  
    conversion to VSAM format 139-145  
SYSDSN major name 83  
SYSGEN (see system generation)  
SYSIEA01 major name 83  
SYSIEECT major name 83  
SYSIEFSD major name 83  
SYSOUT MF/1 parameter 69  
SYSP system initialization parameter 35,31  
SYSPSWRD major name 83  
system activity measurement facility (MF/1) 68-77  
    definition 159  
    highlights 13  
    operation 68-69  
    parameters 69-70  
    reports 70-72  
    SYS1.PARMLIB member 43  
    tracing 72  
    tuning example 73-77  
system activity measurement gathering (SAMG) 68  
system activity report generation (SARG) 68  
system area 153  
system catalog (SYSCTLG) 49  
    conversion to VSAM format 139-145  
    highlights 12  
system data sets 46-50  
    changes to MVT 50  
    changes to VS2 Release 1 49-50  
    optional 48,49  
    password protection 86  
    required 47-48  
system environment recording routines (SER0 and SER1) 15  
system generation  
    changes 25-29  
    JES2 generation 24-25  
    job entry subsystem generation 20,24-25  
    macro instructions (see macro instructions, system generation)  
    procedures 20-29  
    RMTGEN 25  
    starter system 21-22  
    unsupported VS2 Release 1 macro instructions 29  
system initialization 30-46  
    parameters 30-39  
    (see also parameters, system initialization)  
    system parameter library (SYS1.PARMLIB) 39-46  
system integrity 79-87  
    areas of concern 81-85  
    definition 79  
    installation responsibility 80  
    procedural responsibilities 86-87  
system key 81  
    definition 159  
system parameter library (SYS1.PARMLIB) 39-46  
    IBM-created lists 41  
    installation-created lists 45  
    required system data set 47  
    summary of changes 40  
system management facilities (SMF) 106-111  
    differences from MVT 106  
    differences from VS2 Release 1 106  
    exits 111  
    records 107-108  
    system initialization parameter specification 35  
system parameter list (IEASYSxx) 44  
system parameter specification 37  
system protection key 81  
    definition 159  
system queue area (SQA) 153  
    definition 159  
    system initialization parameter specification 35  
system resources management 54-67,73-77  
    definition 159  
    highlights 13  
    JCL support 114  
    operator command support 128  
    parameters 65-67  
    SMF records 107-108  
    time sharing command support 134  
    tuning example 73-77  
system resources management tuning parameter list (IEAOPTxx) 67,45-46  
system restart 33  
system security 86-87  
system summary report 74  
system workload 58  
SYSUDUMP job control language parameter 42  
SYSUTILS system generation macro instruction 27  
SYSVTOC major name 83  
SYS1.ACCT data set 50  
    unsupported MVT function 16  
SYS1.BROADCAST data set 48  
    formatting 146  
    mounted for system initialization 138  
    new format description 49  
SYS1.CMDLIB data set 48  
    adding installation-written members 29  
SYS1.DCMLIB data set 48  
SYS1.DSSVM data set 47  
    change from MVT 50  
    change from VS2 Release 1 49  
    specifying creation of a new nucleus map with the NUCMAP parameter 34  
SYS1.DUMP data set 48  
    device specification by a system initialization parameter 33  
    maximum number specified 36  
SYS1.HELP data set 48  
SYS1.IMAGELIB data set 48  
    adding installation-written members 29  
    use for 3886 OCR devices 116  
SYS1.LINKLIB data set 47  
    adding installation-written members 29

SYS1.LOGREC error recording data set 47  
   change from MVT and VS2 Release 1 49  
 SYS1.LPALIB data set 47  
   adding installation-written members 29  
 SYS1.MACLIB data set 47  
   adding installation-written members 29  
 SYS1.MANX data set 48  
   change from MVT 50  
   change from VS2 Release 1 49  
 SYS1.MANY data set 48  
   change from MVT 50  
   change from VS2 Release 1 49  
 SYS1.NUCLEUS data set 47  
   adding installation-written members 29  
 SYS1.PAGE data set 49  
 SYS1.PAGEDUMP 49  
 SYS1.PARMLIB (system parameter library) 39-46  
   IBM-created lists 41  
   installation-created lists 45  
   required system data set 47  
   summary of changes 40  
 SYS1.PARMLIB lists  
   changes to 40  
   IEAABDxx 42  
   IEAAPFxx 42  
   IEAAP00 41  
   IEABLDxx 43-44  
   IEADMPxx 42  
   IEAFIXxx 42  
   IEAIPSxx 42  
   IEALOD00 43  
   IEALPAXx 45  
   IEAOPTxx 45-46  
   IEAPAK00 43  
   IEASYSxx 44  
   IQAORDER 40  
   IRBMF1xx 43  
   LNKLSTxx 43  
   PARMTZ 45  
   PRESRES 46,40  
   SMFDEFLT 44,40  
   SMFPRMxx 44  
   VATLSTxx 46  
 SYS1.PROCLIB data set 48  
   adding installation-written members 29  
 SYS1.SAMPLIB data set 48  
 SYS1.STGINDEX data set 48  
 SYS1.SVCLIB data set 47  
   adding installation-written members 29  
 SYS1.SYSJOBQE data set 49  
   JES2 91  
   JES3 96  
   unsupported MVT function 16  
   unsupported VS2 Release 1 function 16  
 SYS1.SYSVLOGX (see log data sets)  
 SYS1.SYSVLOGY (see log data sets)  
 SYS1.TELCMLIB data set 48-49  
   adding installation-written members 29  
 SYS1.UADS (user attribute data set) 48  
   formatting 146  
   mounted for system initialization 138  
   new format description 50  
 SYS1.VTAMLIB data set 49  
   adding installation-written members 29  
 SYS1.VTAMLST data set 49  
   adding installation-written members 29  
 SYS1.VTAMOBJ data set 49  
  
 T operand 125  
 tape data set password protection 86-87  
 TAPE MF/1 parameter 69  
 TCAM (telecommunications access method)  
   conversion considerations 148  
   system generation specification 23  
 TELCMLIB system generation macro instruction 27  
 telecommunications access method (TCAM)  
   conversion considerations 148  
   system generation specification 23  
 telecommunications library (SYS1.TELCMLIB) 48-49  
   adding installation-written members 29  
  
 temporary data set 51  
 TESTAUTH facility 51  
 TESTRAN program 16  
 tightly coupled multiprocessing (MP)  
   considerations for VS2 Release 2 149-151  
   definition 157  
   highlights 12  
   operator command support 127  
   SMF record support 110  
 TIME command 137  
 time dependency 148  
 time-of-check-to-time-of-use problem 83  
 time sharing  
   accounting support 138  
   allocation support 134  
   command changes 135-137  
   edit support 138  
   job entry subsystem support 138  
   SMF record elimination 110  
   system resources manager support 134  
   time sharing integration support 138  
   VSAM support 134  
   time slicing 16  
 TMSL system initialization parameter 37,38  
 TOD clock messages 30  
 TP operand 125  
 TRACE command 127  
 TRACE MF/1 parameter 69  
 trace, TSO 16  
 tracing, MF/1 72  
   SMF records 106  
 TRACK command 128,127  
 transaction 159  
 transient areas 16  
 translation tables 159  
 TS operand  
   DISPLAY command 125  
   TRACK command 127  
 TSO (see time sharing)  
 TSO driver 16  
 TSO system generation macro instruction 23,27  
 TSO trace 16  
 TSOAUX system initialization parameter 37  
 TSOPTION system generation macro instruction 27  
 turnaround time 73  
 tuning parameter list (IEAOPTxx) 67,45-46  
 tuning with MF/1 73-77  
 type I programs 17  
  
 u operand 125  
 UADS (see SYS1.UADS)  
 UADSREFM utility program 146  
 UCS system generation macro instruction 29,27  
 unallocating data sets (see freeing data sets)  
 uniprocessing 159  
 UNIT job control language parameter 113  
 UNITNAME system generation macro instruction  
   121-122,23,27  
   specifying VIO processing 114,51  
 UNITR MF/1 parameter 69  
 units code parameter 66  
 unsupported MVT functions  
   job control language summary 113  
   operator command summary 125-127  
   SMF record summary 107-108  
   summary 15-16  
   system data sets 50  
   system initialization parameters 37-38  
   time sharing command summary 135-137  
 unsupported Type I programs 17  
 unsupported VS2 Release 1 functions  
   job control language summary 113  
   operator command summary 125-127  
   SMF record summary 107-108  
   summary 16  
   system data sets 49-50  
   system generation macro instructions 29  
   system initialization parameters 37  
   time sharing command summary 135-137

user address space (see private address space)  
 user area 153  
 user-assigned names 121  
 user attribute data set (SYS1.UADS) 48  
   formatting 146  
   mounted for system initialization 138  
   new format description 50  
 user key 85  
 USER operand 125  
 user protection key 85  
 USERID operand 125  
 user-supplied addresses 81  
 USERS operand 126  
 utility programs  
   IEHIOSUP 15  
   IEHLIST 143,15,16  
   IEHMOVE 143,15,16  
   IEHPROGM 143,28,15,16  
   IEHUCAT 143

**V=R (virtual equals real) address area**  
   ADDRSPC JCL parameter specification 116-117  
   candidate programs 148  
   definition 159  
   programming considerations 146-147  
   REAL system initialization specification 34  
   VRREGN system initialization specification 36  
**V=R programming considerations 146-147**  
**VAL system initialization parameter 36,31**  
 validity checking 80  
   fetch/store operations 83  
   SVC routines calling SVC routines 84  
   unauthorized program requests for authorized library routines 84  
   user-supplied addresses 81  
**VARY command 149,127**  
**VATLSTxx (volume attribute list) 46**  
**VIO (virtual I/O)**  
   catalog for data sets (STGINDEX) 48  
   data set allocation 122-123  
   definition 159  
   highlights 12  
   JCL requirements 114-115  
   MF/1 report data 71  
   SMF record support 106  
   SYSGEN specification with the UNITNAME macro 23  
   system initialization parameter specification 33  
**virtual address**  
   definition 159  
**virtual equals real address area (V=R)**  
   ADDRSPC JCL parameter specification 117  
   candidate programs 146-148  
   definition 159  
   REAL system initialization specification 34  
   VRREGN system initialization specification 36  
**virtual I/O (VIO)**  
   catalog for data sets (STGINDEX) 50  
   definition 159  
   highlights 12  
   JCL requirements 114-115  
   MF/1 report data 71  
   SMF record support 106  
   SYSGEN specification with the UNITNAME macro 23  
   system initialization parameter specification 33  
**virtual memory (virtual storage)**  
   definition 159  
   highlights 13  
**virtual storage**  
   definition 159  
   highlights 13

**virtual storage access method (VSAM)**  
   catalog 139-145  
   definition 159  
   highlights 12  
   SMF record support 110  
   time sharing support 134  
**virtual storage management (VSM)**  
   definition 159  
**virtual storage map 153**  
**virtual telecommunications access method (VTAM)**  
   command support 125-127  
   highlights 12  
   system data sets 49  
**volume attribute list (VATLSTxx) 46**  
**volume sharing 116**  
**volume table of contents (VTOC) 139**  
**VRREGN system initialization parameter 36,31**  
**VSAM (virtual storage access method)**  
   catalog (see VSAM catalog)  
   definition 159  
   highlights 12  
   SMF record support 110  
   time sharing support 134  
**VSAM catalog 139-145**  
   creating 142-143  
   master catalog 47  
   search strategy 141  
   structure 139-140  
   updating 142-143  
   use of access method services 142-143  
**VSM (virtual storage management) 159**  
**VTAM (virtual telecommunications access method)**  
   command support 125-127  
   highlights 13  
   system data sets 49  
**VTAM data sets**  
   SYS1.VTAMLIB 49  
   SYS1.VTAMLST 49  
   SYS1.VTAMOBJ 49  
**VTOC (volume table of contents) 139**

**WKLD MF/1 parameter 69**  
**workload activity measurement 68**  
**workload activity reports 71**  
   performance group period report 71  
   performance group report 71-72  
   system summary report 72  
**workload level 59-66**  
   function of initiated jobs 63  
   MF/1 data in reports 71-72  
   tuning example 73-77  
   used to express service relationships between groups of users 59  
**workload management 54,56,58**  
   definition 159  
**WRITELOG command 127**  
**writer**  
   internal 156  
   OS 15,16  
**WTOBFRS system initialization parameter 36,31**  
**WTORPLY system initialization parameter 36,31**

**XDAP addressed to SYSIN/SYSOUT DCB 15,16**  
**XPT (external page table)**  
   definition 156

**2-channel switches 149**  
**3886 OCR device 116**



*Your views about this publication may help improve its usefulness; this form will be sent to the author's department for appropriate action. Using this form to request system assistance or additional publications will delay response, however. For more direct handling of such requests, please contact your IBM representative or the IBM Branch Office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Index Figures Examples Legibility

Cut or Fold Along Line

What is your occupation? \_\_\_\_\_  
Number of latest Technical Newsletter (if any) concerning this publication: \_\_\_\_\_  
Please indicate in the space below if you wish a reply.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. Elsewhere, an IBM office or representative will be happy to forward your comments.

Cut or Fold Along Line

**Your comments, please . . .**

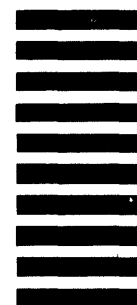
This manual is part of a library that serves as a reference source for system analysts, programmers, and operators of IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold

Fold

First Class  
Permit 81  
Poughkeepsie  
New York

**Business Reply Mail**  
No postage stamp necessary if mailed in the U.S.A.



Postage will be paid by:

International Business Machines Corporation  
Department D58, Building 706-2  
PO Box 390  
Poughkeepsie, New York 12602

Fold

Fold

OS/VS2 Planning Guide for Release 2 (S370-34)

Printed in U.S.A.

GC28-0667-1



**International Business Machines Corporation**  
**Data Processing Division**  
1133 Westchester Avenue, White Plains, New York 10604  
(U.S.A. only)

**IBM World Trade Corporation**  
821 United Nations Plaza, New York, New York 10017  
(International)