

GC28-0755-0  
File No. S370-34

**Systems**

**OS/VS2 System Programming Library:  
Initialization and Tuning Guide**

VS2.03.807

**IBM**

This minor revision incorporates the following Selectable Unit:

Supervisor Performance #2 VS2.03.807

The Selectable Unit to which the information applies is noted in the upper corner of the page.

### **First Edition (July, 1976)**

This is a reprint of GC28-0681-2 incorporating changes released in the following Selectable Unit Newsletter:

GN28-2616 (dated May 28, 1976)

This edition applies to Release 3.7 of OS/VS2 and to all subsequent releases of OS/VS2 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/370 Bibliography*, GC20-0001, for the 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 book is effectively two manuals in one: a system initialization SRL and a preliminary tuning guide. These two “manuals” are combined because of the need for heavy cross referencing between performance discussions and the descriptions of parameters and parmlib members that affect performance. The book is a pioneer effort in both areas: how to initialize the system, and how to get improved system performance. For best results, you should read this manual before you prepare your sysgen deck.

Although it is basically an SRL, elements of logic are included for certain components: the System Resources Manager (SRM), the Auxiliary Storage Manager (ASM), and to some extent NIP and the Program Manager. The logic elements are provided either to clarify recommendations and “how it works” descriptions, or to provide a basis for possible internal modification. This is especially true of the chapter on the System Resources Manager.

The manual consists of six parts or chapters.

- Part 1: Introduction
- Part 2: System Initialization
- Part 2.1: Auxiliary Storage Management Initialization
- Part 3: The System Resources Manager
- Part 4: How to Use the System Activity Measurement Facility (MF/1)
- Part 5: System Performance Factors

Part 2, System Initialization, describes parmlib parameters and processes related to IPL, and certain system commands (START GTF, START MF1, SET IPS). It includes the meaning and use of each parameter, syntax rules, syntax examples, value ranges that are syntactically acceptable, default values, and performance notes where applicable. The chapter points the reader to extended discussions on related topics in Parts 3, 4, and 5.

Part 2.1 describes auxiliary storage management (ASM) with respect to controlling the system’s use of page and swap data sets. It explores the concepts of paging and swap operations, examines the algorithms used, and explains how these algorithms influence the choice of optimum data set sizes. Six examples in the chapter illustrate typical system tuning problems involving the concepts, algorithms, and data set sizes previously discussed. The chapter then makes some performance recommendations and concludes with a series of questions commonly asked about ASM and the answers to them.

Part 3 describes the types of control available through the system resources manager (SRM), the functions used to implement these controls, the concepts for using SRM parameters, and the parameters themselves. Part 3 also presents guidelines for defining these parameters and describes several sample specifications. Finally, Part 3 provides a list of potential problems along with guidelines for evaluating and adjusting the parameters.

Part 4 is a small “SRL” on the use of the System Activity Measurement Facility, MF/1. The chapter describes the parameters that control MF/1 reports and SMF records, discusses how MF/1 can be started (including the IBM-supplied cataloged procedure) and provides some possible uses for each of the system activity reports.

- TCAM Tuning Considerations
- TSO and Batch Service Trade-offs Via the IPS
- Miscellaneous Performance Guidelines

The performance factors information in Part 5, plus performance hints in other parts of the manual, are *preliminary* projections based on design analysis. That is, the information is based on the designers' best judgement of how the system should behave and the factors that should affect its performance. This preliminary performance information will be updated by Installation Newsletters after the manual is printed. The updates will provide tuning guidelines based on system *measurements*.

Several of the performance topics in Part 5 use an experimental technique: inclusion of customer questions asked on these topics, followed by the answers.

### Related Publications

The following manuals should be available for reference while you are reading this SRL:

<i>OS/VS System Management Facilities (SMF)</i>	GC35-0004
<i>OS/VS Utilities</i>	GC35-0005
<i>OS/VS2 Using OS Catalog Management with the Master Catalog: CVOL Processor</i>	GC35-0010
<i>OS/VS2 System Programming Library: System Generation Reference</i>	GC26-3792
<i>OS/VS2 Access Method Services</i>	GC26-3841
<i>OS/VS Virtual Storage Access Method Programmer's Guide</i>	GC26-3838
<i>OS/VS2 TCAM Programmer's Guide</i>	GC30-2041
<i>OS/VS System Programming Library: VTAM</i>	GC28-0688
<i>Operator's Library: OS/VS2 Reference (JES2)</i>	GC38-0210
<i>OS/VS2 System Programming Library: JES2</i>	GC23-0001
<i>OS/VS2 System Programming Library: JES3 System Programmer's Guide</i>	GC28-0608
<i>Operator's Library: OS/VS2 Reference (JES3)</i>	GC38-0226
<i>Operator's Library: OS/VS2 TCAM</i>	GC30-2046
<i>OS/VS2 Supervisor Services and Macro Instructions</i>	GC28-0683
<i>OS/VS2 System Programming Library: Job Management</i>	GC28-0627
<i>OS/VS2 System Programming Library: Supervisor</i>	GC28-0628
<i>OS/VS2 System Programming Library: TSO</i>	GC28-0629
<i>OS/VS2 System Programming Library: Storage Estimates</i>	GC28-0604
<i>OS/VS2 System Programming Library: Service Aids</i>	GC28-0674
<i>OS/VS2 JCL</i>	GC28-0692
<i>OS/VS Message Library: VS2 System Messages</i>	GC38-1002
<i>OS/VS Message Library: JES3 Messages</i>	GC38-1012
<i>OS/VS2 System Programming Library: Data Management</i>	GC26-3830
<i>OS/VS2 Conversion Notebook</i>	GC28-0689
<i>OS/VS Mass Storage System (MSS) Services for Space Management</i>	GC35-0012
<i>IBM System/360 and System/370 ASP Version 3 Asymmetric Multiprocessing System: System Programmer's Manual</i>	GH20-1292
<i>ASP Version 3 Operator's Manual</i>	GH20-1289

<i>Operator's Library: OS/VS2 TCAM</i>	GC30-2046
<i>OS/VS2 Supervisor Services and Macro Instructions</i>	GC28-0683
<i>OS/VS2 System Programming Library: Job Management</i>	GC28-0627
<i>OS/VS2 System Programming Library: Supervisor</i>	GC28-0628
<i>OS/VS2 System Programming Library: TSO</i>	GC28-0629
<i>OS/VS2 System Programming Library: Storage Estimates</i>	GC28-0604
<i>OS/VS2 System Programming Library: Service Aids</i>	GC28-0674
<i>OS/VS2 JCL</i>	GC28-0692
<i>OS/VS Message Library: VS2 System Messages</i>	GC38-1002
<i>OS/VS Message Library: JES3 Messages</i>	GC38-1012
<i>OS/VS2 System Programming Library: Data Management</i>	GC26-3830
<i>OS/VS2 Conversion Notebook</i>	GC28-0689
<i>OS/VS Mass Storage System (MSS) Services for Space Management</i>	GC35-0012

In this manual, any references made to an IBM program product are not intended to state or imply that only IBM's program may be used; any functionally equivalent program may be used instead. This manual has references to the following IBM program products:

RMF – Resource Measurement Facility



# Contents

Summary of Amendments . . . . .	9
<b>Part 1: General Introduction . . . . .</b>	<b>13</b>
<b>Part 2: System Initialization . . . . .</b>	<b>19</b>
Initialization Overview . . . . .	20
Descriptions of Individual PARMLIB Members . . . . .	41
COMMNDxx . . . . .	41
GTFPARM . . . . .	43
IEAABD00 . . . . .	48
IEAAPFxx . . . . .	52
IEAAP00 . . . . .	53
IEABLDxx . . . . .	56
IEADMP00 . . . . .	58
IEAFIXxx . . . . .	61
IEAIPSxx . . . . .	63
IEALOD00 . . . . .	72
IEALPxx . . . . .	73
IEAOPTxx . . . . .	75
IEAPAK00 . . . . .	78
IEASYSxx . . . . .	85
IKJPRM00 . . . . .	130
IRBMF1xx . . . . .	134
LNKLSTxx . . . . .	139
MVIKEY00 . . . . .	140
PARMTZ . . . . .	143
SMFPRMxx . . . . .	145
VATLSTxx . . . . .	151
<b>Part 2.1: Auxiliary Storage Management Initialization (VS2.03.807) . . . . .</b>	<b>158.1</b>
Page and Swap Operations . . . . .	158.1
Paging Operations and Algorithms . . . . .	158.1
Swap Operations and Algorithms . . . . .	158.2
Page and Swap Data Set Sizes . . . . .	158.3
Space Calculation Examples . . . . .	158.5
Example 1: Sizing the PLPA Page Data Set, Size of the PLPA Unknown . . . . .	158.6
Example 2: Sizing the PLPA Page Data Set, Size of the PLPA Known . . . . .	158.6
Example 3: Sizing the Common Page Data Set . . . . .	158.6
Example 4: Sizing the Duplex Page Data Set . . . . .	158.7
Example 5: Sizing Local Page Data Sets . . . . .	158.7
Example 6: Sizing Swap Data Sets . . . . .	158.8
Performance Recommendations . . . . .	158.8
Estimating Total Size of Paging Data Sets . . . . .	158.10
Using Measurement Facilities . . . . .	158.10
Adding More Paging Space . . . . .	158.11
Questions and Answers . . . . .	158.11

<b>Part 3: The System Resources Manager (VS2.03.807)</b>	159
Introduction	159
<b>Section 1: Description of the System Resources Manager (VS2.03.807)</b>	159
Objectives	160
Types of Control	161
Domain Control	161
Workload Control	162
Dispatching Priority Control	162
Throughput Control	162
Functions	163
Swapping	163
Dispatching Priority Control	164
Resource Use Functions	165
Enqueue Delay Minimization	166
Device Allocation	167
Prevention of Resource Shortages	167
Pageable Frame Stealing	169
<b>Section 2: Introduction to SRM Parameter Concepts (VS2.03.807)</b>	169
IPS Concepts	170
Domains	170
Service	173
Service Definition Coefficients	174
Service Rate	174
Performance Objective and Workload Level	176
Interval Service Value (ISV)	181
Response – Throughput Bias (RTB)	183
Performance Period	184
Performance Group	185
OPT Concepts	187
Resource Factor Coefficients (RFC)	187
Resources Manager Constant (RMC)	188
<b>Section 3: Guidelines and Examples (VS2.03.807)</b>	188
Defining Installation Requirements	189
Preparing an Initial IPS and OPT	191
Selecting Service Definition Coefficients	197
Defining Domains and Their Constraints	200
Selecting Dispatching Priorities	202
Defining Durations	202.1
Selecting ISV Values	202.1
Selecting Performance Objectives	202.2
The RTB Parameter and the Resource Factor Coefficients	202.3
Performance Groups	202.4
Default IPS	202.4
IPS Examples	202.6
Evaluating and Adjusting the IPS and OPT	202.8
Fixed Requirements Are Not Being Met	202.8
Optimizing the Use of Control Mechanisms	202.12
<b>Section 4: Installation Management Controls (VS2.03.807)</b>	202.15
Assigning Performance Group Numbers	202.15
Assigning Dispatching Priorities	202.16
Operator Commands Related to the SRM	202.16
<b>Section 5: SRM Parameters (VS2.03.807)</b>	202.17
IEAIPSxx Parameters	202.17
IEAOPTxx Parameters	202.22
<b>Section 6: SRM Constants (VS2.03.807)</b>	202.23



<b>Part 4: How to Use the System Activity Measurement Facility (MF/1)</b>	203
MF/1 Operation	203
MF/1 Options	205
Conflicts Between Options	207
Need for Careful Choice of Certain Parameters and Options	207
MF/1 Reports and SMF Records	209
CPU Activity Report	210
Channel Activity Report	212
I/O Device Activity Reports	215
Paging Activity Report	216
Workload Activity Report	220
Using MF/1	224
Using the CPU Activity Report	224
Using the Channel Activity Report	225
Using the Device Activity Report	226
Using the Paging Activity Report	227
Using the Workload Activity Report	228
<b>Part 5: System Performance Factors</b>	231
The Pageable Link Pack Area: Its Advantages and Uses	240
Recommendations	240
The VS2 Module Search Sequence	243
How the PLPA is Loaded	243
VIO Performance	246
Advantages	246
Disadvantages	248
VIO Performance Considerations	248
How to Specify VIO Data Sets	249
Questions and Answers	250
Device Allocation Performance	252
The Order in Which Allocation Requests Are Serviced	252
Guidelines for Improving Allocation Response	253
TSO Allocation Suggestions	255
Questions and Answers	256
VSAM Catalog Performance	257
Questions and Answers	260
How SMF Can Supplement MF/1	261
Comparison of Paging Rates for a Problem Program and the System	261
Comparison of I/O Activity for a Problem Program and the System	262
Comparison of Problem Program Service and Total Service	262
Determining Changes to the System Configuration	262
The Use of GTF to Track Sysevents	263
TCAM Tuning Considerations	294
Packaging the MCP to Minimize Page Fixes and Page Faults	294
Coding INTRO Operands to Minimize Fixed Pages	296
Ordering OPEN Macros to Minimize Fixed Pages for LCBs and SCBs	299
Miscellaneous Performance Guidelines	303
<b>Index</b>	305



**Figures**

Figure 2-1.	Parmlib Members: Relationships to IPL Parameters and Sysgen Parameters (Part 1 of 2)	25
Figure 2-2.	Characteristics of Parmlib Members (Part 1 of 8)	27
Figure 2-3.	Example of Adding and Replacing Parmlib Members by Means of the IEBUPDTE Utility	35
Figure 2-4.	Mutually Exclusive Options for GTFPARM	45
Figure 2-5.	Performance Objectives in IBM-Supplied IPS.	66
Figure 2-6.	Default Pack List for Time Sharing and Batch System	80
Figure 2-7.	Default Pack List for Batch System	81
Figure 2-8.	Functional Contents of Default Pack Lists (Part 1 of 3)	82
Figure 2-9.	Sysgen Parameters That Are Copied to the IEASYS00 Member	85
Figure 2-10.	Overview of IEASYSxx Parameters (Part 1 of 2)	87
Figure 2.1-1.	Page Data Set Values (VS2.03.807)	158.5
Figure 2.1-2.	Swap Data Set Values (VS2.03.807)	158.6
Figure 3-1.	Relationship between Workload Levels and Service Rates (VS2.03.807)	176
Figure 3-2.	Comparing the Slopes of Two Performance Objectives (VS2.03.807)	177
Figure 3-3.	The Effect of Different Performance Objective Slopes on Service	178
Figure 3-4.	Competition Controlled by Performance Objectives (VS2.03.807)	179
Figure 3-5.	Competition Controlled by Performance Objectives and ISVs (VS2.03.807)	182
Figure 3-6.	The Effect of RFCs on Swap Recommendation Values (VS2.03.807)	187
Figure 3-7.	A Use of Different Cut-off Levels (VS2.03.807)	202.3
Figure 3-8.	The Default IPS (VS2.03.807)	202.5
Figure 4-1.	CPU Activity Report	211
Figure 4-2.	Channel Activity Report	214
Figure 4-3.	Direct Access Device Activity Report	214
Figure 4-4.	Paging Activity Report	219
Figure 4-5.	Workload Activity Report (Part 1 of 2)	222
Figure 4-6.	Variation of System Workload Level with Time of Day	229
Figure 5-1.	Illustration of the Loading of the PLPA	245
Figure 5-2.	SRM Trace Record Format	263
Figure 5-3.	Descriptions of SYSEVENT Codes (Part 1 of 24)	265
Figure 5-4.	Sysevents Sorted by Address Space ID and by Time of Day (Part 1 of 2)	289
Figure 5-5.	Time Distribution Between Sysevents of Same Type	291
Figure 5-6.	Distribution of Time Needed to Complete a Swapout	292
Figure 5-7.	Instances of Real Frame Shortages, Distributed by Time of Day	293
Figure 5-8.	Distribution of Real Frame Shortages	293
Figure 5-9.	Available Buffer Units and CPBs According to UNITSZ Operand	299
Figure 5-10.	LCB Sizes by Terminal Type	300



The JES2 and JES3 Initialization sections and the JES2 Performance sections are no longer in this manual. For information about JES2 and JES3, refer respectively to *OS/VS2 System Programming Library: JES2*, GC23-0001, and *OS/VS2 System Programming Library: JES3*, GC28-0608.

Changes have been made throughout this publication to reflect a Service Update -- OS/VS2 Release 3.7. In addition to general editorial changes, technical updated topics include:

- How to Control Parmlib
- Descriptions of Individual Parmlib Members:
  - COMMNDxx
  - IEABLDxx
  - IEALPAxx
  - IEASYSxx
  - LNKLSTxx
  - MVIKEY00
  - VATLSTxx
- I/O Load Adjusting Routine
- Main Storage Occupancy Routine
- Page Replacement Routine
- Device Allocation Routine
- How the System Resources Manager is Controlled
- Performance Objectives
- SRM Constants Related to the Resource Use Routines
- MF/1 Options
- Guidelines for the Effective Use of Paging Data Sets
- Pageable Link Pack Area
- Device Allocation Performance
- VSAM Catalog Performance
- The Use of GTF to Track Sysevents

**Summary of Amendments  
for GC28-0681-1  
as Updated by GN28-2603  
VS2 Release 3**

This Technical Newsletter, a part of release 3 of OS/VS2, updates the initialization information for the Job Entry Subsystem 3 (JES3). In addition to general editorial changes, the following topics were technically updated:

- How to Control JES3 Initialization
- How JES3 Performs Initialization
- Creating an Initialization Data Set
- BUFFER
- CIPARM
- CLASS
- DEVICE
- DYNALLOC (new initialization card)
- JES3LIB
- MAINPROC
- OPTIONS
- PROC
- RESCTBLK (new initialization card)
- RJPTERM
- SELECT
- SETNAME
- STANDARDS
- SYSOUT

**Summary of Amendments  
for GC28-0681-1  
VS2 Release 3**

- Mass Storage System (MSS) adds a new member, MVIKEY00, to SYS1.PARMLIB. MSS also adds the PURGE parameter and causes changes to the VAL parameter in the IEASYSxx member of SYS1.PARMLIB.
- Multi-Access Spool adds three new JES2 initialization parameters: &CHKPT, QCONTROL, and Sn.
- JES3 Initialization is a new section that explains how to initialize JES3 and describes the JES3 initialization cards and their parameters.
- Vary Storage adds a new parameter, RSU, to the IEASYSxx member of SYS1.PARMLIB. RSU allows the installation to specify the number of storage units that will be available for storage reconfiguration in an MP system.
- Minor technical changes and additions were made throughout the publication.

**Note:** JES3 and Mass Storage System information contained in this publication is for planning purposes only until the products become available.

**Summary of Amendments  
for GC28-0681-1  
as Updated by GN28-2586  
VS2 Release 2**

**This Technical Newsletter provides:**

- **New IBM-supplied default Installation Performance Specifications (IPS).**
- **Changes to many of the SRM examples and explanations in “Part 3: How to Use the System Resources Manager” because of the new default IPS.**
- **Changes to the Auxiliary Storage Manager (ASM) slot sorting algorithms for paging data sets. Also, an expanded explanation of how the PAGE parameter is used by ASM.**
- **New tuning guidelines, some of which were previously printed in IBM Installation Newsletter Issue 74-09 (7/12/74).**
- **Corrections of minor technical and typographical errors.**





This manual discusses the following five general subject areas:

- System initialization
- Auxiliary storage manager
- The system resources manager
- The system activity measurement facility (MF/1)
- System performance factors

### System Initialization

System initialization is the part of system tailoring that takes place after sysgen. The tailoring results from the specification of system parameters, first at IPL and then later when certain operator commands are issued.

System tailoring is the overall process by which an installation selects its operating system. The process consists of the specification of system options through these mechanisms:

- System generation
- IPL-time selections
- Certain operator commands after IPL
- Implicit system parameters

IPL-time choices that help to tailor the system can come from several sources:

- Various types of IPLs.
- Operator entry of parameters from the console.
- SYS1.PARMLIB. This data set is one of the main sources of IPL-time parameters.
- The JES2 or JES3 initialization data set.

There are several general types of IPL:

- The first IPL after sysgen. This is a “cold” start for NIP.
- Any IPL at which the LPA is reloaded. From the NIP viewpoint this is a “cold” start.
- An IPL after power-up (“quick” start).
- An IPL after a system crash (“warm” start).

**Operator Entry of Parameters:** The operator responds to NIP’s SPECIFY SYSTEM PARAMETERS message after he IPLs the system. His response directs NIP, Master Scheduler Initialization, and other components to the desired parmlib members. The operator can enter either ENTER or END\* to default to the basic general parameter list IEASYS00, or enter SYSP=(aa,bb. . .) to select one or more alternate general parameter lists, such as IEASYS01, IEASYS02, etc. The alternate lists can supplement or partially override the basic list. The operator need not enter parameter values directly, except for those cases in which parameters are missing, are syntactically invalid, can’t be read, or must be supplemented to satisfy a special case. (An example of a special case would be the operator entry of the PAGE parameter to increase the amount of paging space on direct access.)

---

\*ENTER is used with the Model 158, END with the Model 168.

If an error occurs with certain parmlib members, the operator is prompted to manually enter one or more of the member's parameters. If the operator can't correct a parameter, he can use ENTER or END on the console. ENTER or END causes selection of system defaults if they exist. Most parameters have defaults, either as default parmlib members, or as coded values in system components. If a default doesn't exist, ENTER or END cancels the parameter. (The defaults are listed in the individual descriptions of parmlib members later in Part 2, System Initialization.)

An operator-entered parameter overrides the same parameter specified in parmlib member IEASYS00 or IEASYSxx, except for:

- A parameter in which operator intervention is prohibited (OPI=NO). In this case, the operator-entered parameter is ignored unless the parmlib parameter was invalid.
- The PAGE parameter. The page data-set names entered by the operator are added for the life of the IPL to those specified in either IEASYS00 or IEASYSxx. (For information on the PAGE parameter, refer to the description of member IEASYSxx in Part 2.)

**The Use of SYS1.PARMLIB:** SYS1.PARMLIB is read by NIP and Master Scheduler Initialization at IPL, and later by such components as the System Resource Manager, the TIOC, GTF, and MF/1, which are invoked by operator command.\* The purpose of parmlib is to provide many initialization parameters in a prespecified form in a single data set, and thus minimize the need for operator entry of parameters.

Parmlib contains both a basic or default general parameter list, IEASYS00, and possible alternate general parameter lists, called IEASYSaa, IEASYSbb, etc. Parmlib also contains specialized members, such as COMMNDxx, PARMTZ, IEALPaxx. The general parameter list(s) contain both parameter values and "directors". The directors (e.g., MLPA=01) point or direct NIP or Master Scheduler Initialization to one or more specialized members, such as IEALPA01.

---

\* The TIOC is the Terminal I/O Coordinator, whose parameters are described under member IKJPRM00. GTF is the Generalized Trace Facility, whose parameters are described under member GTFPARM. Lastly, MF/1 is the System Activity Measurement Facility, which is described in Part 4, "How to Use the System Activity Measurement Facility (MF/1)".

**System Tailoring Through Operator Commands:** After IPL, several operator commands provide additional system tailoring by directing particular groups of parameters to specific system components. The commands consist of:

- Stop and start JES2 (\$p JES2 and \$\$)
- START tcamproc
- MODIFY tcamproc
- SET IPS
- SETDMN
- START GTF
- START MF1
- START vtamproc

### The System Resources Manager

To a large degree, the control which an installation exerts over the functioning of the MVS system is exercised through the mechanism of the System Resources Manager (SRM). Part 3 describes the functioning of the SRM, the parameters which control its functioning, and guidelines for defining these SRM parameters.

The system resources manager (SRM) is a component in the MVS control program. The SRM has two principal objectives:

- First, to distribute the system's processing resources among individual address spaces in a way that satisfies the installation's response and turnaround time objectives.
- Second, to attempt to optimize the use of the system's CPU, storage and I/O resources by system users (address spaces). This is primarily a system throughput consideration.

The SRM uses four types of control to accomplish these objectives:

- Domain Control
- Workload Control
- Dispatching Priority Control
- Throughput Control.

These controls are implemented by combinations of the following seven functions:

- Swapping
- Dispatching Priority Control
- Resource Use Functions
- Enqueue Delay Minimization
- Device Allocation
- Prevention of Storage Shortages
- Pageable Frame Stealing.

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

The System Activity Measurement Facility (MF/1) is an analysis tool which an installation may use to monitor selected areas of system activity, and to obtain feedback in the form of SMF records and/or formatted reports. MF/1 permits the gathering of information on the following classes of system activity, either individually or in combination:

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

- Paging activity
- Workload activity

With MF/1, an installation can monitor the utilization of individual CPU's, channels and devices, in order to identify system components whose overall utilization is exceptional. The installation can further identify periods of system activity during which the utilization of particular resources is exceptional. Finally, the installation can relate the service being provided to different classes of users to the specifications provided in the Installation Performance Specification (IPS).

MF/1 is a software monitoring tool. Thus it is limited to reporting on system activity as that activity is communicated to the system (for example, by the setting of flags). As a result of this indirect reporting, MF/1 can approach in accuracy, in its statistically sampled values, only the internal indications of external system activity. For example, if a CPU is disabled so that the freeing of a device (device-end interruption) cannot be communicated to the system, the device will appear busy for a longer period of time than it would if it were measured by a hardware measuring instrument.

MF/1 is always generated with the system, but its operation is completely optional. The system operator initiates MF/1 monitoring with the START command. MF/1 can also be started as a batch job. When MF/1 is not operating, it will cause little performance or storage overhead. When MF/1 is operating, the storage and performance overhead depend on the set of options that were specified.

In an installation using the Mass Storage System (MSS), Mass Storage System Trace Report programs are used to monitor the MSS. These programs are described in *OS/VS Mass Storage System (MSS) Services for Space Management*.

### **System Performance Factors**

Part 5 contains discussions of factors that affect system performance and discussions of certain tools that can measure performance. Each discussion includes guidelines and rationale. Some discussions also include questions and answers. Since MVS is significantly different from MVT, the discussions emphasize new aspects of the system, such as VIO, VSAM catalog, and the pageable link pack area. The performance information is based on design analysis; that is, on projections of how the system is supposed to work. Some of these ideas will change as system experience is gained.

The following performance topics are included in Part 5, in this order:

- The Pageable Link Pack Area: Its Advantages and Uses
- VIO Performance
- Device Allocation Performance
- VSAM Catalog Performance
- How SMF Can Supplement MF/1
- The Use of GTF to Track Sysevents
- TCAM Tuning Considerations
- Miscellaneous Performance Guidelines



## Part 2: System Initialization

This part of the book contains two sections:

- An overview of initialization
- Pamlib member descriptions

SYS  
INIT

# Initialization Overview

System Initialization is the part of system tailoring that takes place after sysgen. The tailoring results from the specification of system parameters, first at IPL and then later when certain operator commands are issued.

## System Tailoring

System tailoring is the overall process by which an installation selects its operating system. The process consists of the specification of system options through these mechanisms:

- System generation
- IPL-time selections
- Certain operator commands after IPL
- Implicit system parameters



## System Generation

System generation involves the specification of particular system options under a starter system or driver, before the desired system is available. Some system options are specified by means of parameters in sysgen macros, such as CTRLPROG, SCHEDULR, or DATASET. Some of the CTRLPROG, DATASET and SCHEDULR options become the initial contents of certain members of SYS1.PARMLIB, such as IEASYS00, IEAAPF00, and IEAAPP00. Other members of parmlib (IEAABD00, IEADMP00, IEABLD00, LNKLST00, and IEAPAK00) are copied directly from the APARMLIB data set to SYS1.PARMLIB. The initial contents of these members, although not determined by the system programmer at sysgen, can later be altered or enlarged through the use of the IEBUPDTE utility. (See Figure 2-3 for the complete list of members created and copied at sysgen.)

## System Tailoring at IPL Time

IPL-time choices that help to tailor the system can come from several sources:

- Various types of IPLs.
- Operator entry of parameters from the console.
- SYS1.PARMLIB. This data set is one of the main sources of IPL-time parameters.
- The JES2 initialization data set or JES3 initialization deck.
- Alternate nucleus selection.

## Types of IPL

There are several general types of IPL:

- The first IPL after sysgen. This is a “cold” start for NIP.
- Any IPL at which the LPA is reloaded. From the NIP viewpoint this is a “cold” start.
- An IPL after power-up (“quick” start).
- An IPL after a system crash (“warm” start).

**The First IPL After Sysgen:** At the first IPL after sysgen, NIP automatically loads the LPA from SYS1.LPALIB. The page data sets for this IPL are those named in parmlib member IEASYS00 (built at sysgen), plus any specified by the operator. These page data sets are those which were specified in sysgen DATASET macros that contained the PAGEDSN keyword or the DUPLEXDS,NAME keyword combination. The operator need not enter the CLPA (create link pack area) parameter in order to load the LPA, since at this first IPL the NIP program will do this automatically. However, he should enter FORMAT (meaning cold start) as a response to the SPECIFY OPTIONS message, since no job queues exist yet on the spool data set.

**An IPL at Which the LPA Is Reloaded:** The LPA need be loaded only at two times: at the first IPL after sysgen, when NIP loads it automatically, and at an IPL after the installation has added or modified one or more modules in SYS1.LPALIB, has tested the alteration, and now wants to put the replacement module(s) in the LPA. To reload the LPA from LPALIB, the operator would enter CLPA (create link pack area) as one of his responses to the SPECIFY SYSTEM PARAMETERS message. Such reloading of the LPA should not be a common occurrence. It should be done only when necessary, since the associated I/O slows down the IPL and because previously existing VIO data sets are deleted. (For further information on the loading of the LPA, see the CLPA parameter in the description of the IEASYSxx parmlib member, and the topic “The Pageable Link Pack Area: Its Advantages and Uses” in the System Performance Factors chapter.)

**An IPL After Power-Up:** The IPL after power-up can be called a “quick start”, since the LPA from the previous IPL can be used without reloading from LPALIB. Furthermore, VIO data sets can be purged, page data sets added, and swap data sets replaced. VIO data sets retained from the previous work shift can be purged, if the installation wishes, by the use of the CVIO (clear VIO) parameter. The operator, or a general parameter list of parmlib, can add additional page data sets by specifying the PAGE parameter and can replace swap data sets by specifying the SWAP parameter. (For information on the CVIO and PAGE parameters, see the description of the IEASYSxx parmlib member later in this chapter.)

**An IPL After a System Crash:** After a system crash the operator can “warm start” the system by entering or defaulting to the WARM parameter as a response to the SPECIFY OPTIONS message. Existing VIO data sets can be retained by the Auxiliary Storage Manager for continued use. Therefore, neither the operator nor his specified parmlib parameter list (IEASYSxx) would include the CVIO (Clear VIO) parameter. (The specification of one or more IEASYSxx members by the operator at IPL time is described in the next topic, “Operator Entry of Parameters”.)

*Note:* See *OS/VS2 System Programming Library: Supervisor* for information on the Power Warning Feature (PWF) pre-IPL options following a power disturbance.

## Operator Entry of Parameters

The operator responds to NIP's SPECIFY SYSTEM PARAMETERS message after he causes IPL. His response directs NIP, Master Scheduler Initialization, and other components to the desired parmlib members. The operator can enter either ENTER or END\* to default to the basic general parameter list IEASYS00, or enter SYSP=(aa,bb...) to select one or more alternate general parameter lists, such as IEASYS01, IEASYS02, etc. The alternate lists can supplement or partially override the basic list. The operator need not enter parameter values directly, except for those cases in which parameters are missing, are syntactically invalid, can't be read, or must be supplemented to satisfy a special case. (An example of a special case would be the operator entry of the PAGE parameter to increase the amount of paging space on direct access.)

If an error occurs with certain parmlib members, the operator is prompted to manually enter one or more of the member's parameters. (Figure 2-4 later in this overview lists the parmlib members that prompt for replacement of invalid or missing parameters.) If the operator can't correct a parameter, he can use ENTER or END on the console. ENTER or END causes selection of system defaults if they exist. Most parameters have defaults, either as default parmlib members, or as coded values in system components. If a default doesn't exist (and if a parameter is not required), ENTER or END cancels the parameter. (The defaults are listed in the individual descriptions of parmlib members later in this chapter.)

An operator-entered parameter overrides the same parameter specified in parmlib member IEASYS00 or IEASYSxx, except for:

- A parameter in which operator intervention is prohibited (OPI=NO). In this case, the operator-entered parameter is ignored (unless the parmlib parameter was syntactically invalid and is being corrected from the console).
- The PAGE parameter. The page data-set names entered by the operator are added for the life of the IPL to those specified in either IEASYS00 or IEASYSxx. (For information on the PAGE parameter, refer to the description of member IEASYSxx later in this chapter.)

---

\*ENTER is used with the Model 158, END with the Model 168.

## The Use of SYS1.PARMLIB

SYS1.PARMLIB is ready by NIP and Master Scheduler Initialization at IPL, and later by such components as the System Resource Manager, the TIOC, GTF, and MF/1, which are invoked by operator command.\* The purpose of parmlib is to provide many initialization parameters in a prespecified form in a single data set, and thus minimize the need for operator entry of parameters.

Parmlib contains both a basic or default general parameter list IEASYS00 and possible alternate general parameter lists, called IEASYSaa, IEASYSbb, etc. Parmlib also contains specialized members, such as COMMNDxx, PARMTZ, IEALPAxx. The general parameter list(s) contain both parameter values and "directors". The directors (e.g., MLPA=01) point or direct NIP or Master Scheduler Initialization to one or more specialized members, such as IEALPA01.

Member IEASYS00, the default general parameter list, is always read but its contents can be overridden and/or augmented by the alternate general parameter list(s). IEASYS00 can be further supplemented and/or partially overridden by operator-entered parameters. The IEASYSxx-lists are selected by the operator through the SYSP parameter at IPL. The specialized members can be named by the general parameter lists (IEASYS00 and IEASYSxx), or named by the operator at IPL. To IPL only with IEASYS00 and with the specialized lists it names, the operator would enter END or ENTER instead of SYSP=xx.

If the same parameter appears in both IEASYS00 and a specified alternate IEASYSxx list, the value in the alternate list overrides. In addition, a parameter value in a later specified IEASYSxx list overrides the same parameter in an earlier specified list. For example, assume that the operator enters R00, 'SYSP=(01,02)' in order to select the two parameter lists IEASYS01 and IEASYS02. Further assume that these two lists and IEASYS00 contain these values:

```
IEASYS00:    . . . ,MLPA=00,BLDL=00, . . .
IEASYS01:    . . . ,MLPA=(01,02) ,BLDL=01, . . .
IEASYS02:    . . . ,MLPA=03,SQA=10, . . .
```

From the above values, NIP accepts: MLPA=03,BLDL=01,SQA=10.

If a particular parameter or member is unavailable or incorrect, one of the following results takes place, depending on the particular member:

- A default value is used, or
- Processing of the parameter or the rest of the member is bypassed, or
- The operator is prompted to enter replacements for the invalid parameter(s), or to enter all the parameters in the member or to re-IPL, or to cancel the parameter or member by entering END or ENTER.

The handling of each member at a syntax error or read error is listed later in this overview in Figure 2-2.

---

\*The TIOC is the Terminal I/O Coordinator, whose parameters are described under member IKJPRM00. GTF is the Generalized Trace Facility, whose parameters are described under member GTFPARM. Lastly, MF/1 is the System Activity Measurement Facility, which is described in Part 4, "How to Use the System Activity Measurement Facility (MF/1)".

**How Parmlib Members Are Created:** Parmlib members are created in several ways:

- Some are unconditionally created at sysgen by being copied from the APARMLIB data set. They may later be changed or augmented by the installation through the use of the IEBUPDTE utility.
- Others are conditionally created at sysgen, if particular sysgen macros and keywords are specified.
- The remaining members can be explicitly created by the installation.

Figure 2-1 shows which parmlib members are created at sysgen, whether the creations are conditional, the names of any associated sysgen macros and keywords, and the IPL-time parameters that direct the reading system components to the desired specialized members. (See notes at bottom of figure.)

Member	Associated IPL-Time Parameter (in IEASYS00, IEASYSxx, or entered by the Operator)	Initially Built at Sysgen	Sysgen Macro and Parameter
COMMNDxx	CMD=xx	no	N/A
GTFPARM	none	no	N/A
IEAABD00	none	yes: default list is copied from APARMLIB.	N/A
IEAAPF00	APF=00	conditionally, if sysgen macro is specified	CTRLPROG APFLIB=
IEAAPFxx	APF=xx	no	N/A
IEAAP00	none	conditionally, if sysgen macro is specified.	DATASET ABEAPP= CHEAPP= EOEAPP= PCIAPP= SIOAPP=
IEABLD00	{ BLDL } { BLDLF } = 00	yes: default list is copied from APARMLIB, augmented via sysgen macro if it is specified.	CTRLPROG OPTIONS= BLDL
IEABLDxx	{ BLDL } { BLDLF } = xx	no	N/A
IEADMP00	none	yes: default list is copied from APARMLIB.	N/A
IEAFIX00	FIX=00	conditionally, if sysgen macro is specified.	DATASET RESIDENT=
IEAFIXxx	FIX=xx	no	N/A

**Notes:**

1. A "no" in the third column means the member is installation-created.
2. N/A means "not applicable".

Figure 2-1. Parmlib Members: Relationships to IPL Parameters and Sysgen Parameters (Part 1 of 2)

Member	Associated IPL-Time Parameter (in IEASYS00, IEASYSxx, or entered by the Operator)	Initially Built at Sysgen	Sysgen Macro and Parameter
IEAIPS00	IPS=00	yes: default list is copied from APARMLIB.	N/A
IEAIPSxx	IPS=xx	no	N/A
IEALOD00	None	no	N/A
IEALPAxx	MLPA=xx	no	N/A
IEAOPT00	OPT=00	no	N/A
IEAOPTxx	OPT=xx	no	N/A
IEAPAK00	none, although member is used only when CLPA is specified.	yes: default list is copied from APARMLIB, optionally augmented or altered by installation before IPL.	N/A
IEASYS00	none	yes, if sysgen macros are specified.	(see Figure 2-11 in IEASYSxx)
IEASYSxx	SYSP=xx, issued by the operator	no	N/A
IKJPRM00	none	no	N/A
IRBMF100	none	yes: default list is copied from APARMLIB.	N/A
IRBMF1xx	none	no	N/A
LNKLST00	LNK=00	yes: default list is copied from APARMLIB.	N/A
LNKLSTxx	LNK=xx	no	N/A
MVIKEY00	none	yes: default list is copied from APARMLIB.	N/A
PARMTZ	none	no, although time zone constant can be specified by the TZ keyword of CTRLPROG macro and placed in the CVT.	N/A
SMFPRM00	SMF=00	yes	N/A
SMFPRMxx	SMF=xx	no	N/A
VATLST=xx	VAL=xx	no	N/A

Figure 2-1. Parmlib Members: Relationships to IPL Parameters and Sysgen Parameters (Part 2 of 2)

**Overview of Parmlib Members:** Figure 2-2 lists all the parmlib members that are valid in MVS. The table briefly describes the purpose of each member, indicates whether the member was introduced in MVT, VS2 Release 1, or MVS, and lists additional categories of information for each of the parmlib members.

Member	When was member introduced?	Supplied by IBM	Required or Optional	Directly affects performance	Read at IPL or command	Allows listing of parameters at IPL or command	Response to Errors (N/A=not applicable)		
							Syntax Error	Read Error	Unsupported Parameters
COMMND00 (xx)	Commands to be issued by the control program immediately after Initialization. Also contains a parameter that controls prompting during TOD clock initialization. JES2 commands may <i>not</i> be included.								
	VS2-2	no	optional	no	IPL	no	Ignores invalid command name. TOD defaults to NOPROMPT.	No commands are processed.	N/A
Parameters to control GTF when GTF is started.									
GTFARM	VS2-2	yes	Optionally used. Built automatically at sysgen. Can be modified by installation.	Indirectly since SRM sysevent trace option is a tuning aid.	START GTF	Yes, automatically at START GTF.	On any error, prompts for all parameters.	Same as with syntax error.	N/A
Default parameters for an ABEND dump when a SYSABEND dd statement has been specified.									
IEAABD00	VS2-2	yes	Optional. However, if member is unavailable, ABEND dumps may not be possible without DUMPOPT lists. (Also, see GTFARM above for information on sysgen.)	no	IPL	no	Message lists valid parms that were accepted. Invalid parms are rejected.	Error message. Parameters are rejected.	N/A

Figure 2-2. Characteristics of Parmlib Members (Part 1 of 8)

Member	When was member introduced?	Supplied by IBM	Required or Optional	Directly affects performance	Read at IPL or command	Allows listing of parameters at IPL or command	Response to Errors (N/A=not applicable)		
							Syntax Error	Read Error	Unsupported Parameters
IEAAPF00 (xx)	Names of authorized program libraries.								
	VS2-2	Created option-ally at sysgen by APFLIB keyword of CTRL-PROG macro.	optional (LINKLIB and SVCLIB are always authorized.)	no	IPL	no	Prompts operator to re-specify bad parameters or cancel parms via END or ENTER key.	Same as with syntax error.	N/A
IEAAPF00	Names of authorized installation-written I/O appendage routines.								
	VS2-2	Created option-ally at sysgen by appendage keywords of DATA-SET macro	optional	no	IPL	no	Error message. Partial appendage name table is built, if possible.	Same as with syntax error.	N/A
IEABLD00 (xx)	Default list of modules in SYS1.LINKLIB (or in data sets concatenated to SYS1.LINKLIB) whose directory entries will go in a resident BLDL table. IEABLDxx is similar to IEABLD00 except that these may be multiple alternate lists.								
	MVT. Multiple members introduced in VS2-2.	IEABLD00-yes IEABLDxx-no	optional	yes	IPL	yes, if L option is specified with BLDL(F) parm in IEASY00(xx), or by the operator.	Prompts operator to re-specify bad parameters or cancel parms via END or ENTER key.	Same as with syntax error.	Error message. Obsolete module names are ignored.

Figure 2-2. Characteristics of Parmlib Members (Part 2 of 8)



Member	When was member introduced?	Supplied by IBM	Required or Optional	Directly affects performance	Read at IPL or command	Allows listing of parameters at IPL or command	Response to Errors (N/A=not applicable)		
							Syntax Error	Read Error	Unsupported Parameters
IEADMP00	Default parameters for an ABEND dump when a SYSUDUMP dd statement has been specified.								
	VS2-2	yes	Optional. However, if member is unavailable, ABEND dumps may not be possible without DUMPOPT lists. Built automatically at sysgen. Can be modified by installation.	no	IPL	no	Message lists valid parms that were accepted. Invalid parms are rejected.	Error message. Parm are rejected.	N/A
IEAFIX00 (xx)	Names of modules from SYS1.LPALIB, SYS1.SVCLIB, and SYS1.LINKLIB to be fixed in real storage for the life of the IPL.								
	VS2-1	IEAFIX00-yes IEAFIXxx-no	IEAFIX00 is created optionally at sysgen	yes	IPL	yes, if L option is spec'd with FIX parm in IEASY00(xx), or by the operator.	Prompts operator to re-specify bad parameters or cancel them via END or ENTER key.	Same as with syntax error.	Error message. Obsolete module names are ignored.

Figure 2-2. Characteristics of Parmlib Members (Part 3 of 8)

Member	When was member introduced?	Supplied by IBM	Required or Optional	Directly affects performance	Read at IPL or command	Allows listing of parameters at IPL or command	Response to Errors (N/A=not applicable)		
							Syntax Error	Read Error	Unsupported Parameters
IEAIG00	Resident ERP list in MVT. (Not supported in VS2 Release 2.)								
IEAIG00	Resident access method list. (Not supported in VS2 Release 2.)								
IEAIPS00 (xx)	Parameters of an installation performance specification that control the system resources manager.								
VS2-2	IEAIPS00- yes IEAIPSxx- no	IEAIPS00= req'd IEAIPSxx=opt.	yes	both IPL and SET IPS command.	yes, if L option is spec'd with IPS parm in IEASYS00(xx), or by the operator.	Prompts operator to re-specify alternate IPS member or to default to IEAIPS00. If no IEAIPS00, uses coded defaults.	Prompts operator to re-specify IPS=xx or to cancel parm via END or ENTER key.	N/A	
IEALOD00	Names of LPA modules whose directory information will be placed in real storage to avoid page faults involved in searching the PLPA directory.								
VS2-1	no	optional	yes	IPL	no	Module names before error are processed. Names after error are omitted.	Same as with syntax error.	Error message. Ignores obsolete module names.	
IEALPA00 (xx)	Names of reenterable modules from LINKLIB, SVCLIB, and LPALIB as a temporary addition (MLPA) to the PLPA.								
VS2-1	no	optional	yes	IPL	yes, if L option is spec'd with the MLPA parm in IEASYS00(xx), or by the operator.	Prompts operator to re-specify, or cancel parm via END or ENTER key.	Same as with syntax error.	Error message. Ignores obsolete module names.	

Figure 2-2. Characteristics of Parmlib Members (Part 4 of 8)

Member	When was member introduced?	Supplied by IBM	Required or Optional	Directly affects performance	Read at IPL or command	Allows listing of parameters at IPL or command	Response to Errors (N/A=not applicable)		
							Syntax Error	Read Error	Unsupported Parameters
Parameters that control resource management algorithms in the System Resource Manager.									
IEAOP00 (xx)	VS2-2	no	optional	yes	IPL	yes, if L option is spec'd with the OPT parm in IEASYS00(XX), or by the operator.	Error message. Default values substituted.	Prompts operator to select alternate member by respecifying OPT parm.	N/A
"Pack List" names of groups of modules in LPALIB that NIP will load between page boundaries to minimize page faults.									
IEAPAK00	VS2-1	Yes. Is automatically copied at sysgen. Can be modified by installation.	optional	yes	IPL	no	Bypasses the pack group that contains the error. Processes the next pack group.	The pack groups read before the error are processed. Other pack groups are omitted.	Error message. Ignores obsolete module names.
IEARSV00	Resident SVC list in MVT. (Not supported in VS2 Release 2.)								

Figure 2-2. Characteristics of Parmlib Members (Part 5 of 8)

Member	When was member introduced?	Supplied by IBM	Required or Optional	Directly affects performance	Read at IPL or command	Allows listing of parameters at IPL or command	Response to Errors (N/A=not applicable)		
							Syntax Error	Read Error	Unsupported Parameters
IEASY00 (xx)	System parameters that are valid responses to the SPECIFY SYSTEM PARAMETERS message. Multiple system parameter lists are valid. List is chosen by operator SYSP parameter. IEASY00, the default member, is initially built at syngen, although later modifiable by the installation.								
	VS2-1	yes. Is built at syngen from macros.	See Note 2.	See Note 3.	IPL	See Note 1. (L parm must be spec'd with parm or member.)	Prompts operator to re-specify, or cancel parm via END or ENTER key. Params processed before the error are retained.	Params processed before error are retained. Operator is asked to specify an alternate member. If he does, new parms override those retained.	Obsolete parms are treated as syntax errors.
<b>Notes:</b> 1. These parameters can be listed at IPL: APF, BLDL/BLDLF, DUMP, FIX, IPS, MLPA, SYSP, and OPT. 2. There are only two mandatory parms, PAGE and HARDCPY. Other parms have coded defaults. 3. Performance-oriented parms in IEASY00 (xx): APG, CMD, FIX, BLDL/BLDL, IPS, MLPA, OPT, REAL, RSU, WTOBERS, WTORPLY.									
IKJPRM00									
TIOC parameters that are used to control time sharing buffers.									
	MVT. Changed in VS2-2.	no	optional	yes	MODIFY toamproc command, if TS= START is spec'd in command.	no	Default value is substituted.	Same as with syntax error.	Obsolete parms ignored.

Figure 2-2. Characteristics of Parmlib Members (Part 6 of 8)

Member	When was member introduced?	Supplied by IBM	Required or Optional	Directly affects performance	Read at IPL or command	Allows listing of parameters at IPL or command	Response to Errors (N/A=not applicable)		
							Syntax Error	Read Error	Unsupported Parameters
IRBMF100 (xx) Parameters to control MF/1 data and reports.									
	VS2-2	IRBMF100-yes IRBMF1xx-no	IRBMF100= req'd to run MF/1 IRBMF1xx= opt.	Indirectly since MF/1 is a tuning aid.	START MF1 command.	Yes, if OPTION parm is spec'd.	Error message All accepted parms are listed. The operator is prompted to correct in-valid parms.	Error msg., and handles same as with syntax error.	N/A
LNKLIST00 (xx) List of data sets to be concatenated to SYS1.LINKLIB. LNKLISTxx is similar to LNKLIST00 except that these may be multiple alternate lists.									
	MVT Multiple members in VS2-2.	no	optional	no	IPL	Yes, if L option is spec'd with LNK parm in IEASYS00(XX), or by the operator.	Prompts operator to re-specify, or cancel parm via END or ENTER key.	No data sets are concatenated with LINKLIB. Operator can re-IPL to specify alternate LNKLIST member.	Error msg. Data sets not found will not be concatenated with LINKLIB.
MVIKEY00 Parameters to control MSSC data and messages.									
	VS2-3	yes. Is automatically copied at sysgen. Can be modified by installation.	required for MSS	yes	IPL	Yes, at IPL. If error occurs, default is taken.	Prompts operator to enter parm or re-IPL.	Same as with syntax error.	Same as with syntax error.

Figure 2-2. Characteristics of Parmlib Members (Part 7 of 8)

Member	When was member introduced?	Supplied by IBM	Required or Optional	Directly affects performance	Read at IPL or command	Allows listing of parameters at IPL or command	Response to Errors (N/A=not applicable)		
							Syntax Error	Read Error	Unsupported Parameters
<b>PARMTZ</b> Time zone constant: the value by which local time differs from Greenwich Mean Time.									
	VS2-1	no	optional	no	IPL	no	Uses sysgen value. If no value was spec'd at sysgen, default of 0 is used.	Same as with syntax error.	N/A
<b>SMFPRM00 (xx)</b> Parameters that define SMF options. Multiple members replace SMFDEFLT, which was used in MVT and VS2 Release 1.									
	MVT. Changed in VS2-2.	PRM00 yes PRMxx no	PRM00= req'd. PRMxx=opt. (See Note 4.)	Indirectly since SMF can supplement MF/1 as a tuning aid.	IPL	Yes, if OPI parm is spec'd as YES	Prompts operator to enter parm or re-IPL.	Same as with syntax error.	Same as with syntax error.
<b>VATLST00 (xx)</b> Volume attribute list that defines the 'mount' and 'use' attributes of direct access volumes. Multiple lists named VATLSTxx replace PRESRES.									
	MVT. Name and format changed in VS2-2.	no	optional	yes	IPL	No. Operator has option to get list only if error occurs.	Error msg. By-passes bad entry. Processes remaining entries.	Operator is given choice of processing remaining list(s) if multiple lists, or specifying new VATLST VATLSTxx member, or re-IPLing.	Old member (PRESRES) can't be processed.
<b>Note 4:</b> The installation must add DSV=2 or DSV=3 to the SMF default list SMFPRM00 so that the IEHUCAT utility will be able to update MVT catalogs.									

Figure 2-2. Characteristics of PamLib Members (Part 8 of 8)

**How to Control Parmlib:** To control parmlib and assure that it is manageable, you should consider the following problem areas and suggested solutions:

- Delete unsupported parameters and members. Since most components treat unsupported parameters from previous releases as syntax errors, you should probably remove the old parameters or build parmlib from scratch. This action will minimize the need for operator responses during an IPL. Furthermore, you can save space by removing unsupported members. Figure 2-2 shows which old members are not supported in MVS, and which changed members can involve unsupported parameters.
- Update parmlib with new and replacement members, as you gain familiarity with the new release. You can use the IEBUPDTE utility to add or replace members. Figure 2-3 illustrates the JCL for adding three new members and replacing two old members: IEABLD00, IEALOD00, IEAPAK00, IEASYS05, and IEASYS06. (Consult the *OS/VS Utilities* manual for further information on the use of IEBUPDTE.) To prevent excessive growth of parmlib, use the “compress” function of IEBCOPY to delete obsolete data.

---

```

//ADDLISTS      JOB      61938, 'R.L. WILSON'
//STEP          EXEC     PGM=IEBUPDTE, PARM=MOD
//SYSPRINT      DD       SYSOUT=A
//SYSUT1        DD       DSNAME=SYS1.PARMLIB, DISP=OLD
//SYSUT2        DD       DSNAME=SYS1.PARMLIB, DISP=OLD
//SYSIN         DD       DATA
./              ADD NAME=IEABLD00, LIST=ALL
./              NUMBER NEW1=01, INCR=02
SYS1.LINKLIB    IEFSD061, IEFSD062, IEFSD064, IEFSD104,
                IEFVM1, IEFWC000, IEFWD000, IEFW21SD,
                IEFW41SD, IEFW42SD, IEFXJ000
./              REPL    NAME=IEALOD00, LEVEL=01, SOURCE=1,
                LIST=ALL
./              NUMBER NEW1=10, INCR=100
                IEFAB400, IGG0325A, IGG0325H, IGC0003B,
                IGG0325B, IGG0325D, IGG0325E,
                IGG0325G, IFG0202J, IFG0202K, IFG0202L
./              REPL NAME=IEAPAK00, LEVEL=01, SOURCE=1, LIST=ALL
./              NUMBER NEW1=01, INCR=02
                (IEFAB400, IGG0325A, IGG0325H, IGC0003B),
                (IGG0325B, IGG0325D, IGG0325E,
                IGG0235G), (IFG0202J, IFG0202K, IFG0202L)
./              ADD NAME=IEASYS05, LIST=ALL
./              NUMBER NEW1=01, INCR=02
                MLPA=(00, 01),
                BLDL=00, SQA=2
./              ADD      NAME=IEASYS06, LIST=ALL
./              NUMBER NEW=01, INCR=02
                MLPA=(02, 03), BLDLF=(00, 01),
                SQA=1, FIX=(00, 01), OPI=NO
./              ENDUP
/*

```

**Note:** This example shows the format of IEBUPDTE statements, not the content of parmlib members.

---

**Figure 2-3. Example of Adding and Replacing Parmlib Members by Means of the IEBUPDTE Utility**

- Keep track of which parameters were specified at sysgen (through CTRLPROG and DATASET macros) and which parameters are included in particular parmlib members. This bookkeeping is necessary for two reasons: 1. The system doesn't keep track of parmlib members and their parameters. 2. The default general parameter list IEASYS00 is always read by NIP and Master Scheduler Initialization. The list's parameters can be overridden by the same parameters when they are specified in alternate general lists, such as IEASYS01, IEASYS02, etc. Furthermore, certain parameters, such as FIX, APF, and MLPA, direct the system to particular specialized members (in this example, IEAFIXxx, IEAAPFxx, and IEALPaxx). The installation should keep records of which parameters and which values are in particular members, and which general members point to which particular specialized members (COMMNDxx, IEALPaxx, etc.) A grid or matrix for such bookkeeping is very helpful.
- Allocate sufficient space for parmlib. The space must be a single extent. One way to estimate space is to count the number of characters, including blanks, (by counting 80-character records) in all members. Then add a suitable growth factor (e.g., 100–300%) to allow for future growth of alternate members. Consult Figure 2-2 to determine which members can have multiple alternates. To recapture space from obsolete members, use the “compress” function of IEBCOPY.

*Note:* It is better not to let members cross cylinder boundaries because some members are used during IPL. Those that are will generate I/O errors during IPL if they cross cylinder boundaries.

- Decide the volume and device that should hold parmlib. The volume could be demountable, although it must be mounted in order for the operator to start GTF, to start MF/1 as supplied with the system, to start time sharing by use of the MODIFY tcamproc command, or to try to specify a new installation performance specification by use of the SET IPS command. The volume must be cataloged, unless it is SYSRES. It could be placed on a slow or moderate speed device.
- Password protect the data set. Parmlib should be password protected for “write”. The purpose is to shield the appendage member (IEAAPPO0) and the Authorized Program Facility member (IEAAPFxx) from user tampering. Member IEAAPPO0 is particularly sensitive, since a user could add an appendage that would give his routine control in zero protection key and supervisor state.

**General Syntax Rules for the Creation of Members:** The following general syntax rules apply to the creation of most parmlib members. Exception to these rules are described under specific members later in this chapter. The general rules are:

- Record size is 80 bytes.
- Any columns between 1 and 71 may contain data.
- Columns 72 and 80 are ignored.
- Continuation is indicated by a comma followed by one or more blanks after the last entry on a record.
- Leading blanks are suppressed. A record therefore need not start at a particular column.
- Suffix member identifiers (such as LNK=A2) can be any alphanumeric combination.



### Causing an Alternate Nucleus Substitution

Another less common way to change the system at an IPL is to cause the IPL program to read a member of a nucleus data set that is different from IEANUC01, the default nucleus member. One reason for such a nucleus switch may be the need to apply a PTF. The operator process to cause the switch is described under “Loading a Secondary Nucleus” in the *Operator’s Library: OS/VS2 Reference (JES2)*.

### System Tailoring Through Operator Commands

After IPL, several operator commands provide additional system tailoring by directing particular groups of parameters to specific system components. The commands consist of:

- START tcamproc
- MODIFY tcamproc
- SET IPS
- SETDMN
- START GTF
- START MF1
- START vtamproc

### Starting TCAM

The operator can enter TCAM initialization parameters when he starts TCAM, as a response to the SPECIFY TCAM PARAMETERS message. TCAM issues the message if the system programmer accidentally or deliberately omitted one or more required parameters when he coded the INTRO macro for TCAM assembly. In response to the message, the operator can add to or modify existing TCAM parameters. (For information on the TCAM parameters, refer to *OS/VS2 TCAM Programmers Guide*.)

### Starting VTAM

VTAM start options can be entered by the network operator as parameters in the START command or they can be specified in a start option list. The start option list is stored in SYS1.VTAMLST and is specified by the LIST parameter in the START command. For information about VTAM initialization, refer to *OS/VS2 System Programming Library: VTAM*.

### Starting Time Sharing

The operator starts TCAM, then issues the following command in MVS in order to start time sharing:

```
MODIFY tcamproc, TS=START [,membername]
```

The optional parameter membername can specify an installation-defined parmlib member, or can default to the parmlib member IKJPRM00. In either case, the member is ready by the Terminal I/O Coordinator (TIOC). The TIOC uses the parameters mainly to control time-sharing buffers. (For additional information on TIOC parameters, see the description of member IKJPRM00 later in this chapter.)

### Using SET IPS to Change System Resources Manager Parameters

The SET IPS command causes the system resources manager to receive a specific installation performance specification (IPS). This IPS is a parmlib member (IEAIPSxx). By use of this command, the installation can dynamically change IPS parameters between IPLs. (For further information, refer to the IPS parameter in the description of the IEASYSxx member, and in the chapter entitled “The System Resources Manager”.)

### Using SETDMN to Change Domain Constraints

The SETDMN command provides a way for altering the constraint values on a domain’s multiprogramming level. The information from this command is valid only for the life of the IPL – it does not change fields in the IPS member.

## Starting GTF

When the operator starts the Generalized Trace Facility (GTF), the parameters are obtained from the PARM field of the START command and from a parmlib member. If the operator issues START GTF, the IBM-supplied cataloged procedure (named GTF) is read. The PROC statement of that procedure names GTFPARM as the member from which GTF will get its parameters. If, however, the installation wants to substitute another member in place of GTFPARM, the operator may enter the alternate member name with the MEMBER keyword of the START command. (For further information on GTF initialization parameters, see the description of member GTFPARM later in this chapter. For other information on starting or using GTF, refer to the GTF chapter in *OS/VS2 System Programming Library: Service Aids*.)

## Starting MF/1

The System Activities Measurement Facility (MF/1) is started by means of the START procname command. Parameters to control measurements, reports, and SMF records are taken from a merge of three sources, in this order:

1. PARM field of the START command
2. PARM field of the EXEC statement in the proc named by the command
3. MF/1 partitioned data set member (typically the IRBMF1xx member of parmlib)

If the IBM-supplied procedure (named MF1) is specified in the command, the MF/1 partitioned data set is SYS1.PARMLIB. The MF/1 member names must always be of the form IRBMF1xx. The default member IRBMF100 is supplied by IBM. (For further information on MF/1, its uses and parameters, see the part of this book entitled “How to Use the System Activity Measurement Facility (MF/1).”)

## Implicit System Parameters

Various system requirements, although not involving explicit parameters, affect the way the system performs. These system requirements may be considered as “implicit” parameters. They involve dd statements, data sets, hardware choices, etc. Some examples are:

- SYSABEND and SYSUDUMP dd statements. Without these statements, the parameters in parmlib members IEAABD00 and IEADMP00 and the dump option lists in ABEND macro instructions are useless, since an ABEND dump cannot be taken.
- SYSMANX and SYSMANY data sets must be allocated on direct access volumes and be cataloged. If this condition is not met, the Master Scheduler fails during IPL. This may be considered an implicit SMF parameter.
- Addition of new modules to SYS1.LPALIB through use of IEBCOPY or the Linkage Editor. This affects the size and usefulness of the LPA that is loaded by specification of the CLPA parameter at IPL.
- Choice of the device on which the LPA paging data sets will reside. This choice affects the speed at which LPA modules can be paged into real storage and thus system performance.

- Definition of page data sets by means of the DEFINE PAGESPACE command. The PAGE parameter, issued at IPL through parmlib and/or the operator, is meaningful only if the specified data sets have been previously formatted by the DEFINE PAGESPACE command. (See *OS/VS2 Access Method Services* for information on this command.)

**Warning:** NIP enforces a uniprocessor IPL when a system that has been generated without MP modules tries to initialize on MP or half duplex. The operator at IPL must ensure that at least 768K bytes of contiguous storage are dialed online, if the system was generated with ACRCODE=NO in the CTRLPROG macro. If the storage requirement is not met, NIP issues a message and may cause a wait state.

# Descriptions of Individual PARMLIB Members

The individual parmlib members, listed in alphabetic order, are described in the following topics.

## Member Name: COMMNDxx

**Status:** New for VS2 Release 2

### Use of the Member

COMMNDxx is an optional installation-created list of automatic commands to be internally issued by the system as part of Master Scheduler Initialization. COMMNDxx is useful for automatic entry of commands, such as START MF/1, that may be frequently issued at System Initialization. The member cannot be used to issue JES2 commands, since COMMNDxx commands are issued before JES2 is started. COMMNDxx also contains the TOD keyword. This keyword indicates whether messages from TOD Clock Initialization should be issued to prompt the operator. The installation can use this keyword to speed up the initialization process, if it feels that these messages are unnecessary. (See the NOPROMPT operand of the TOD keyword.)

The TRACE ON command may be placed in a COMMNDxx member, if OS Trace is desired after NIP processing (to cover log initialization, startup of GTF, and the latter part of JES2 initialization). OS Trace is turned off during GTF operation if TRACE ON was requested at IPL. TRACE OFF may be requested at the next IPL by operator command (before responding to the JES2 SPECIFY OPTIONS message), or via another COMMNDxx member.

**Caution:** The use of TRACE ON should not be encouraged, because it degrades the system and because it can't be stopped until the next IPL.

The following commands should not be put in COMMNDxx because they are console-oriented. They should be issued only on the console for which they apply:

Command	Abbreviation
CONTROL	K
MSGRT	MR
TRACK	TR
STOPTR	PT

**Warning:** Do not place JES2 commands in COMMNDxx, since JES2 is not started until after the COMMNDxx entries have been processed.

The default member COMMND00, if it exists, is read if CMD=xx is not included in the system parameter list (IEASYSxx) or is not specified by the operator. If Initialization can't find either the specified COMMNDxx member or COMMND00, processing continues without provision for automatic commands.

## COMMNDxx (continued)

**Parameter in IEASYSxx:**  
(or issued by the operator)

$$\text{CMD} = \left\{ \begin{array}{l} \text{aa} \\ (\text{aa}, \text{bb} \dots) \end{array} \right\}$$

The two-character identifier (aa, bb, etc.) is appended to COMMND to identify the COMMNDxx member(s) of parmlib. Multiple members can be specified.

*Note:* Commands issued from COMMNDxx do not show on the console. Therefore, the results of these commands appear on the console without the operator's seeing the command.

### Syntax Rules

The following rules apply to the creation of COMMNDxx by means of the IEBUPDTE utility:

- Enter only one command per card image. Enter the COM= keyword, followed by the command enclosed in apostrophes. For example, to start TCAM through use of the IBM-supplied PROC, enter COM='s TCAM'.
- Specify the TOD=keyword on a single card image. For example, TOD=PROMPT (This entry will cause prompting messages during TOD clock initialization.)
- Do not specify continuation on any card image.

**IBM-Supplied Defaults:** None

### Internal Parameters

Parameter	Meaning	Default Value
COM='command name'	The specified command will be issued by the system during Master Scheduler Initialization.	No commands will be issued.
TOD= $\left\{ \begin{array}{l} \text{PROMPT} \\ \underline{\text{NOPROMPT}} \end{array} \right\}$	Messages will (will not) be issued to the operator during TOD clock initialization  If NOPROMPT is specified, the system will prompt the operator to set the TOD clock only if the clock is not set, or in a multiprocessing system, if the TOD clocks are not synchronized.	NOPROMPT

**Member Name:** GTFARM (or an installation-supplied name)

**Status:** New for VS2 Release 2

### Use of the Member

GTFARM provides default or installation-defined trace options to control the Generalized Trace Facility (GTF). The member is read only when the operator (or an automatic command) issues START GTF. It is not used during System Initialization. (For use of the TRACE command to continue or discontinue OS Trace after IPL, see *Operator's Library: VS2 Reference (JES2)*.)

The procname of the START command can name an IBM-supplied cataloged procedure. The PROC statement of that procedure identifies GTFARM as the member from which GTF will get its trace parameters. If the installation wants to substitute another member in place of GTFARM, the operator may enter the replacement member name on the START command with the MEMBER keyword.

The IBM procedure, as supplied in SYS1.PROCLIB, contains these statements:

```
//GTF PROC      MEMBER=GTFARM
//IEFPROC EXEC  PGM=AHLGTF,PARM='MODE=EXT,DEBUG=NO,
                TIME=NO'
//IEFRDER DD   DSNAME=SYS1.TRACE,UNIT=SYSDA,
//             SPACE=(4095,20),DISP=(NEW,KEEP)
//SYSLIB DD    DSN=SYS1.PARMLIB(&MEMBER),DISP=SHR
```

For further analysis of this procedure, refer to the GTF chapter of *OS/VS2 System Programming Library: Service Aids*.

Since default options in GTFARM specify minimal data for only a limited number of traced events, you may wish to expand GTF capabilities through one of the following methods:

- Specify another member name via the MEMBER keyword on the START command.
- Change the trace options in GTFARM, by using the IEBUPDTE utility.
- Change the SYSLIB DD statement of the IBM procedure, in order to specify a different option list, which you create via IEBUPDTE.
- Retain the IBM procedure to handle default options, and write one or more alternate procedures, each specifying a different alternate parmlib member. You could design each member to contain GTF options useful under particular circumstances. Instruct the operator when to issue the START command for each procname.

GTF tries to read parameters from the specified parmlib member. If an error occurs in opening or reading the member, or if GTF detects a syntax error, it writes a diagnostic message to the operator, and requests him to SPECIFY TRACE OPTIONS, as if no GTF parmlib member were available. The operator therefore must have a complete list of desired GTF parameters available when he starts the facility.

## GTFARM (continued)

**Parameter in IEASYSxx:** None  
(or issued by the operator)

### Syntax Rules:

The following rules apply to the creation of a GTF parmlib member by means of the IEBUPDTE utility:

- Specify the TRACE keyword and its main options only on the first record. Do not place them on subsequent records. For example,

Record #1: TRACE=IOP,SVCP,SIO

This example requests the tracing of specific I/O interrupts, specific SVC interrupts, and all Start I/O operations.

- The second and subsequent records should contain only “prompting” keywords, such as IO= or SVC=. These keywords provide for detailed operands that indicate which I/O interrupts or which SVC interrupts should be traced. For example, the IOP and SVCP keywords in the Record # 1 example (above) must be followed by prompting records that name specific unit addresses and specific SVC numbers for which interrupts should be traced. As an example,

Record # 2: IO=(191,192,193),SVC=(1,2,3)

If the specific operands of any prompting keyword are missing, GTF does not prompt the operator initially. It accepts a general specification. For example, if IOP is specified in record # 1, and particular device addresses are not specified in a later record, GTF assumes that tracing of I/O interrupts is desired for *all* devices.

Later, when all records have been read, GTF issues message AHL103I TRACE OPTIONS SELECTED – IO (rather than IO=(191,192,193)). The operator can then respond to the accompanying message AHL125A RESPECIFY TRACE OPTIONS OR REPLY U by entering Rxx,IO=(191,192,193) to indicate the devices whose I/O interruptions should be traced.

- An END statement or an end-of-file must follow all prompting keywords.
- If you need to specify additional operands for the same keyword, restate the keyword and the additional operands in a subsequent prompting record. The previous examples, expanded to include additional SVC numbers and an END keyword, would appear like this:

Record # 1: TRACE=IOP,SVCP,SIO

Record # 2: IO=(191,192,193),SVC=(1,2,3)

Record # 3: SVC=(4,5,6,7,8,9,10),END



## GTFPARAM (continued)

- Certain trace options (prompting vs. non-prompting, general vs. specific) are mutually exclusive. (See Figure 2-4) Options listed in the same column of the table are mutually exclusive. If you select two or more options from the same column, the option listed highest in the column will take effect. For example, if you specify both SYSP and SIO from the first column, SYSP will take effect and SIO will be ignored.

**Note:** Options listed in each column are mutually exclusive.  
Preferential order is from top to bottom.

SYSM	SYSM	SYSM	SYSM	SYSM	SYSM	DSP	USR	TRC	PCI	SRM
SYSP	SYSP	SYSP	SYSP	SYSP	SYSP	These parameters are not mutually exclusive with other parameters.				
SYS	SYS	SYS	SYS	SYS	SYS					
SIO	IOP	SVCP	PIP	EXT	RR					
SIO	IO	SVC	PI							

Figure 2-4. Mutually Exclusive Options for GTFPARAM

## IBM-Supplied Defaults

When GTF is started by specifying the IBM-supplied cataloged procedure, the following options exist in GTFPARAM:

TRACE=SYSM,USR,TRC,DSP,PCI,SRM

These keywords will cause the following events to be recorded: SVC interruptions, I/O interruptions, PCI interruptions, program interruptions, external interruptions, dispatcher executions, Start I/O instructions, entries to the System Resource Manager, entries to recovery routines, events associated with GTF (TRC), and data passed to GTF via the GTRACE macro (USR). All keywords except USR result in minimal format trace entries. USR entries will be the length specified by the user in the GTRACE macro. Such entries may optionally be a maximum of 256 bytes, excluding the prefix.

## Internal Parameters

### Parameter

### Meaning and Use

SYS

This parameter requests comprehensive recording for six system events: I/O, SIO, SVC, program, external interruptions, and entry to recovery routines (RR). If any additional trace options are coded (e.g., SRM, DSP), trace entries for these options will also be comprehensive.

SYSP

This parameter produces results similar to that produced by SYS, except that GTF prompts for particular events (e.g., address of I/O device or SVC number). As with SYS, if any additional trace options are coded, trace entries for these options will also be comprehensive.

## GTFPARAM (continued)

Parameter	Meaning and Use
SYSM	<p>This parameter produces results similar to that for SYS, except that minimal trace data are recorded for the six events. If any additional trace options are coded (e.g., SRM, DSP), trace entries for these options will also be minimal, except for USR entries. USR entries are the length specified by the user in the GTRACE macro.</p> <p><b>NOTE:</b> Specification of either SYS or SYSM causes the following trace options to be ignored if specified, since they are included in SYS or SYSM: SIO, IO, SVC, PI, EXT,RR.</p>
SIO	<p>This parameter requests comprehensive recording for system SIO operations.</p>
SIOP	<p>This parameter is similar to SIO except that it requests GTF to prompt for the addresses of specific devices (e.g., 151, 152) for which SIO events should be recorded. Only the devices for which the operator replies, or the parmlib member specifies, will cause SIO entries in the trace.</p>
IO	<p>This parameter requests comprehensive recording of all non-PCI I/O interruptions. To obtain recording of PCI interruptions, specify PCI. Both options may be specified.</p>
IOP	<p>This parameter requests the same type of recording as does IO, except that GTF prompts for the addresses of specific devices whose I/O interruptions will be recorded.</p>
SVC	<p>This parameter requests comprehensive recording of all SVC interruptions.</p>
SVCP	<p>This parameter is similar to SVC, except that GTF will prompt for specific SVC numbers for which data is to be recorded.</p>
PI	<p>This parameter requests comprehensive recording for all program interruptions (1-19).</p>
PIP	<p>This parameter is similar to PI except that GTF prompts for the specific interruption codes for which data is to be recorded.</p>
EXT	<p>This parameter requests comprehensive recording for all external interruptions.</p>

## GTFPARAM (continued)

Parameter	Meaning and Use
RR	This parameter requests that uses of recovery routines (FRRs and STAE/ESTAE routines) be recorded. The trace record is created when the recovery routine returns control to the Recovery Termination Manager (RTM). Data is in comprehensive format except when the option is specified via SYSM.
DSP	This parameter requests recording for the local dispatching of units of work (service request block (SRB) and TCB). The option is not included in the specification of SYS or SYSM. It must be specified in addition to other parameters. The parameter produces comprehensive format except when SYSM is also specified. With SYSM, data is in minimal format.
PCI	This parameter requests that PCI interruptions be recorded in the same format as other requested I/O trace records. If specific I/O device addresses are specified through prompting records, the PCI interruptions will be recorded for the same devices. I/O tracing must be requested, since PCI cannot be specified without IO.
SRM	This parameter requests a trace entry each time that the System Resource Manager (SRM) is invoked. The option is not included in the specification of SYS or SYSM. It must be specified in addition to other parameters. Data is in comprehensive format except when SYSM is also specified. Comprehensive format includes the jobname.
TRC	This parameter requests that traced events include those related to GTF processing itself. If this parameter is not specified, GTF-related events will be excluded from the trace output.
USR	This parameter requests that user data passed to GTF via the GTRACE macro be recorded with the system data.
END	This parameter indicates the end of the prompting records. In the parmlib member an end of file serves the same purpose. Never include the END parameter in the first record (the record that contains the TRACE parameters), since GTF regards such occurrence as an error.

**Member Name:** IEAABD00 — (See also IEADMP00)

**Status:** New for VS2 Release 2

### Use of the Member

IEAABD00 contains IBM defaults and/or installation assigned parameters for ABDUMP, for use when an ABEND dump is written to a SYSABEND data set. (For ABDUMP parameters associated with a SYSUDUMP data set, see member IEADMP00.)

ABDUMP parameters for a dump to be taken to a SYSABEND data set may be specified in any of three ways:

- A parameter list pointed to by the DUMPOPT keyword of an ABEND macro instruction.\* The list can be built by using the list form of the SNAP macro. (See *Supervisor Services and Macro Instruction* for details regarding the ABEND and SNAP macros.)
- Default options specified in IEAABD00. These options control the content of the ABEND dump if the programmer doesn't include the DUMPOPT keyword and its associated parameter list with his ABEND, CALLRTM, or SETRP macro. If the DUMPOPT keyword is included, the specified options are merged with those in IEAABD00.
- Temporary override options that an operator can specify with a CHNGDUMP command (see cautionary statement below).

The reader should note that options specified by the CHNGDUMP command are distinctly different from those provided by the DUMPOPT list or the IEAABD00 options. Either of these lists can provide a reasonable dump, selected either by the application programmer or the installation system programmer. The CHNGDUMP command, however, must be issued very carefully, in order not to nullify dump options. Parameters in a CHNGDUMP command are *overriding*; that is, they temporarily replace *all* dump options specified in both IEAABD00 and all ABEND macro instructions. *Only* those parameters specified in CHNGDUMP are effective. Parameters not specified in CHNGDUMP are temporarily nullified, even though they still exist in IEAABD00 or in ABEND DUMPOPT lists. The overrides prevail until the operator issues another CHNGDUMP command, specifying the DEL keyword, or until the system is reinitialized (whichever occurs first).

As an example of how CHNGDUMP can accidentally wipe out desired ABDUMP options, assume that IEAABD00 contains the following options:

```
SDATA=(SQA,CB,ENQ),PDATA=(PSW,REGS,SA,ALLPA)
```

Further assume that the operator is told to add TRT to the SDATA options, in order to dump the trace table, and to add SPLS to the PDATA options in order to dump user storage acquired for the failing task. The operator, having only a superficial grasp of the command, enters:

```
CHNGDUMP SET,SYSABEND,SDATA=(TRT),PDATA=(SPLS)
```

---

\*An ABEND dump can also be requested by a CALLRTM or SETRP macro. See *OS/VS2 System Programming Library: Supervisor*.

## IEAABD00 (continued)

The operator thinks he has added the TRT and SPLS parameters to an incomplete list and that he has expanded the list by two parameters. Actually, the operator has set up effectively a new parameter list containing only *two parameters*: SDATA=TRT and PDATA=SPLS. He has unwittingly overridden and therefore nullified SDATA=(SQA,CB,ENQ),PDATA=(PSW,REGS,SA,ALLPA). He has also overridden any DUMPOPT parameters specified by programmers for use with their ABEND macro instructions. (For syntax information on the CHNGDUMP command refer to *Operator's library: VS2 Reference (JES2)*.)

The ABDUMP Initialization routine reads IEAABD00 to get ABDUMP parameters. If during Initialization, IEAABD00 is found to be invalid or can't be located, the operator is notified. No prompting occurs. If both valid and invalid options are included in the member, or a syntax error is encountered, a message lists the valid options that were accepted before the error occurred.

**Parameter in IEASYSxx:** None  
(or specified by the operator)

## Syntax Rules

The following rules apply to the replacement of IEAABD00 by means of the IEBUPDTE utility:

- There are two keywords, SDATA and PDATA. Each keyword is followed by a string of operands separated by commas and enclosed in parentheses. A single operand does not need parentheses.

### *Examples:*

SDATA=(SQA,CB,ENQ,TRT) or SDATA=ALLSDATA  
PDATA=(PSW,REGS,SA,ALLPA,SPLS) or PDATA=ALLPDATA

- Normally both parameters (i.e., SDATA=operands and PDATA=operands) can fit on one card image. If, however, continuation is needed, use a comma followed by a blank.

### *Example:*

SDATA=(SQA,CB,ENQ,TRT),  
PDATA=(PSW,REGS,  
SA,ALLPA,SPLS)

## IEAABD00 (continued)

### IBM-Supplied Defaults

The following defaults are placed in IEAABD00 by IBM:

SDATA=(LSQA,CB,ENQ,TRT),PDATA=(ALLPA,SPLS)

These options request a dump of the following areas:

- LSQA, including subpools 229 and 230
- formatted control blocks for the task
- formatted enqueue control blocks for the task
- GTF trace or supervisor incore trace  
(See explanation under TRT parameter)
- modules listed on the link pack area queue and the job pack area queue, and active SVC modules related to the failing task
- user storage allocated for the task

### Internal Parameters

#### Parameter

#### Meaning and Use

SDATA=

ALLSDATA All the following options are automatically specified.

The following parameters request dump of specific SDATA areas, as indicated:

NUC	Control program nucleus. SQA, LSQA and the PSA are included.
SQA	The system queue area.
LSQA	Local system queue area for the address space, including subpools 229 and 230.
SWA	Scheduler work area used for the failing task.
CB	Control blocks related to the failing task.
ENQ	Enqueue control blocks (QCBs and QELs) related to the failing task.
TRT	GTF or supervisor trace table depending on whether the ABEND occurs after or during IPL, and whether the TRACE ON command was issued at IPL. If the ABEND occurs during NIP or Master Scheduler Initialization, the supervisor trace is displayed. Otherwise, the GTF trace is displayed, provided that GTF has been started. If GTF is not running, and the TRACE ON command was issued at IPL, the Supervisor trace table is displayed. (For the use of the TRACE command, see <i>Operator's Library: VS2 Reference (JES2)</i> .)

## IEAABD00 (continued)

Parameter	Meaning and Use
PDATA=	
ALLPDATA	All the following options are automatically specified.
	<b>The following parameters request dump of specific PDATA areas, as indicated:</b>
PSW	Program status word at entry at ABEND.
REGS	Contents of general registers at entry to ABEND.
SA or SAH	SA requests save area linkage information and a backward trace of save areas. This option is automatically selected if ALLPDATA is specified.  SAH requests only save area linkage information.
JPA	Contents of the job pack area (module names and contents) that relate to the failing task.
LPA	Contents of the LPA (module names and contents) related to the failing task. Includes active SVCs related to the failing task.
ALLPA	Contents of both the job pack area and the LPA, as they relate to the failing task, plus SVCs related to the failing task.
SPLS	User storage subpools (0–127) related to the failing task.

## Member Name: IEAAPFxx

Status: New for VS2 Release 2

### Use of the Member

IEAAPFxx is a list of program library names (dsnames) and corresponding volume serial numbers that require APF authorization. (APF means the Authorized Program Facility.) SYS1.LINKLIB and SYS1.SVCLIB are automatically authorized. SYS1.LPALIB, however, is not automatically authorized, since it is closed at the end of NIP processing and is not required until the next IPL at which time the LPA is to be reloaded.

If an installation wants IEAAPFxx, it must explicitly create the member via the IEBUPDTE utility. The default member IEAAPF00, however, may be optionally created at sysgen through the use of the APFLIB keyword of the CTRLPROG macro.

**Warning:** Be careful when you list a library in IEAAPF00 or IEAAPFxx that is concatenated to other libraries. If any of the concatenated libraries is not authorized, *all* the concatenated libraries will become unauthorized when they are opened.

**Parameter in IEASYSxx: APF=xx**  
(or specified by the operator)

The two-character identifier xx is appended to IEAAPF to identify the IEAAPFxx member. If the APF parameter is not specified, only SYS1.LINKLIB and SYS1.SVCLIB will be authorized (i.e., placed in the APF table).

### Syntax Rules

The following rules apply to the creation of IEAAPFxx by means of the IEBUPDTE utility:

- Place only one library name and corresponding volume serial number on a record (card image).
- Duplicate data set names are valid.
- On each record, first enter the library name, then one or more blanks, then the volume serial number.
- To continue to another record, place a comma after the volume serial number. Omit this comma on the last record.

#### *Example*

```
first record:   LIB087   614703,  
second record: LIB122   705650
```

### IBM-Supplied Default

If neither IEAAPFxx nor IEAAPF00 exists, SVCLIB and LINKLIB are authorized, plus libraries concatenated to LINKLIB via the LNKLST00 or LNKLSTxx member of parmlib. These data sets are always included in the table of authorized program libraries.

**Internal Parameters:** Not applicable.



## Member Name: IEAAPPO0

**Status:** New for VS2 Release 2

### Use of the Member

IEAAPPO0 contains the names of authorized installation-written I/O appendage routines. These appendages, when listed, can be used by any unauthorized user program. Otherwise, only programs authorized under APF or running under system protection key (0-7) may use the EXCP appendages. If your installation does not use EXCP appendages, you need not create IEAAPPO0.

IEAAPPO0 can be built during the sysgen process, if the installation specifies any of the appendage keywords of the DATASET macro. The possible keywords and the associated appendage types are:

SIOAPP	Start I/O appendages
CHEAPP	channel end appendages
EOEAPP	end-of-extent appendages
PCIAPP	PCI appendages
ABEAPP	abnormal end appendages

NIP accesses SYS1.PARMLIB, reads the list of appendage names in IEAAPPO0, builds an appendage name table, and sets a pointer to the table in the CVT. On each subsequent OPEN, the appendage name table will be examined, instead of IEAAPPO0. If the EXCP caller is not in system protection key or the job step is not authorized, Open verifies that the caller's appendage names are listed. If the names can't be found, Open issues a 913 ABEND. If, however, the caller is authorized, Open loads the appendages without inspecting the list.

IEAAPPO0 can be created at sysgen, with all appendage names specified, even though all appendages have not yet been created. The IEBCOPY step of sysgen will, in this case, issue a diagnostic message, but will not fail the step. The installation thus has the ability to "prime" IEAAPPO0 with appendage names whose modules can be created later. To create or alter IEAAPPO0 after sysgen, use the IEBUPDTE utility. Then re-IPL.

### Rules for Specifying Appendages with the DATASET Macro at Sysgen

- Use the appendage keywords (e.g., SIOAPP=) for user appendages that are to be used by unauthorized problem programs. The full name of an appendage is eight characters long. The first six characters must be IGG019. The last two characters, specified as the operands of the appendage keywords, can range from WA to Z9. Example: CHEAPP=XZ,ZZ,Z6.
- Specify LPALIB or SVCLIB as the system library parameter.
- Do not use the MEMBERS= keyword (unless the appendages are used exclusively by authorized programs). The MEMBERS keyword would prevent sysgen from building IEAAPPO0.

## IEAAP00 (continued)

- Each appendage keyword can list up to 84 appendage-name suffixes (e.g., XZ).
- Omit a particular appendage keyword if there are no appendages of that type. For example, do not specify EOEAPP if end-of-extent appendages are not desired.
- Omit all appendage keywords, if there are to be no user-written appendages.

### *Syntax Example for DATASET Macro:*

The following example shows how the sysgen DATASET macro can be used to specify appendage names:

```
DATASET  LPALIB
          PDS=MYLIB (user data set from which the appendage modules
                    are to be copied to LPALIB)
          MEMBERS=(IGG019XX,IGG019WA)
          SIOAPP=(XY)
          CHEAPP=(XZ,ZZ,ZY)
```

In this example six appendages are copied to LPALIB. The two appendages, IGG019XX and IGG019WA, are copied to LPALIB but are not listed in IEAAP00. These two are to be used only by authorized programs. The other four appendages, a Start I/O and three channel-end appendages, are copied to LPALIB and are listed in IEAAP00.

## Syntax Rules

The following rules apply to the construction of IEAAP00 by means of the IEBUPDTE utility:

- IEAAP00 can contain up to five entries, each entry containing names of a particular type of appendage (e.g., abnormal-end or Start I/O). You need not necessarily use all five types of entries.
- Each entry consists of an appendage-type name, followed by a list of suffixes of that type, separated by commas. The appendage-type name can start in any column. For example:  

```
SIOAPP      WA,W1
```

This entry specifies two Start I/O appendages.
- Indicate continuation, as with most parmlib members, by means of a comma followed by at least one blank. The next record can start in any column.

## IEAAPPO0 (continued)

### *Syntax Example:*

Here is an example of a complete IEAAPPO0 member:

SIOAPP	Y1,Y2,
EOEAPP	X1,W2,X3,X4,X5,X6,
PCIAPP	X3

Note in this example that there are no channel-end appendages and none for abnormal end. Routine IGG019X3 is used as both end-of extent and PCI appendage.

### **IBM-Supplied Default**

If IEAAPPO0 doesn't exist, only IGG019E4 (channel end/abnormal end appendage for Interactive Terminal Facility) is placed in the table of authorized appendage routines.

**Internal Parameters:** Not applicable.

**Member Name: IEABLDxx** (Resident BLDL List)

### Status

The purpose of IEABLD00 is unchanged from MVT, although initial contents from sysgen has changed. IEABLDxx, permitting multiple alternate members, was introduced with VS2 Release 1.

### Use of the Member

IEABLDxx contains the names of load modules in SYS1.LINKLIB, or any data set concatenated to SYS1.LINKLIB, whose data set directory entries NIP will place in a pageable or fixed BLDL table. (In MVT, member IEABLD00 contained module names for either LINKLIB or SVCLIB. In MVS many SVC modules are included in the link pack area.)

NIP builds a table of data set directory entries for use by LINK, LOAD, ATTACH, and XCTL macro instructions. The Program Manager can fetch a requested load module from LINKLIB to a paging data set without issuing a BLDL macro to search the data set directory, if a resident BLDL entry exists for the module. The fetch in this case takes less time than if a BLDL macro had to be issued.

The created BLDL table will exist in either pageable or fixed storage (but not in both) depending on the system parameter specified, BLDL or BLDLF. Alternately, you may specify a fixed BLDL table, without naming its contents, by specifying the BLDL option of the CTRLPROG macro at sysgen. With either method, you create the contents of the table by means of the IEBUPDTE utility.

There are several ways to reduce fetch time for frequently used modules. One way is to select a fixed BLDL table instead of a pageable one. This choice reduces the number of page faults, although at a slight cost in real storage (60 bytes per directory entry). Additional I/O time can be saved by placing frequently and moderately used reentrant and refreshable load modules in the pageable or fixed link pack area. Another technique is to add modules to the link pack area extension (MLPA) for the life of the current IPL. (For information on creating and extending the LPA, refer to the CLPA and the MLPA parameters in the description of member IEASYSxx. Additional performance information is provided in the topic "The Pageable Link Pack Area: Its Advantages and Use" in the System Performance Factors chapter.)

**Parameter in IEASYSxx: BLDLF=xx or BLDL=xx**  
(or specified by the operator)

NIP appends the two alphanumeric characters xx to IEABLD to form the member name IEABLDxx. BLDLF specifies that NIP is to create the resident BLDL table in fixed (real) storage. BLDL, on the other hand, specifies that NIP is to create the resident BLDL table in virtual storage and that the table is to be paged.

## IEABLDxx (continued)

The two keywords BLDLF and BLDL are mutually exclusive. If you specify both parameters, BLDLF will be accepted and BLDL ignored. In this case, NIP issues an informational message. If you specify neither parameter, NIP does not try to find an IEABLDxx member. Only IEABLD00 will be used, and it will be pageable (unless the BLDL parameter of the CTRLPROG macro was specified at sysgen).

### Syntax Rules

Use the general syntax rules listed in the introduction to this chapter. In addition, list the load module names in the same order as they appear in the data set directory, that is, in alphameric order, separated by commas. You may include either major or alias names.

### IBM-Supplied Default

The IEABLD00 member is always created at sysgen. The member contains either of two versions, depending on whether TSO is requested at sysgen. Both versions remain in parmlib, named IEABLDBA (for batch only) and IEABLDTS (for mixed TSO and batch). The sysgen-selected version is renamed IEABLD00 and is copied into parmlib from APARMLIB.

#### Batch Version (IEABLDBA)

```
SYS1.LINKLIB    HEWL,IEWL,IFOX00,IFOX01,IFOX02,IFOX03,IFOX04,
                IFOX05,IFOX06,IFOX11,IFOX21,IFOX31,IFOX51,
                IFOX61,IFOX62,LINKEDIT,LOADER
```

#### TSO/Batch Version (IEABLDTS)

```
SYS1.LINKLIB    HEWL,IEWL,IFOX00,IFOX01,IFOX02,IFOX03,IFOX04,
                IFOX05,IFOX06,IFOX11,IFOX21,IFOX31,IFOX51,
                IFOX61,IFOX62,LINKEDIT,LOADER,
                ALLOC,ALLOCATE,E,EDIT,LINK,LOGOFF,LOGON,
                SUBMIT,TEST
```

*Note 1:* HEWL and IEWL are aliases for the Linkage Editor. IFOXxx names the Assembler XF.

*Note 2:* LNKLST00 by default at sysgen time contains SYS1.LINKLIB. The TSO modules are in SYS1.CMDLIB which must be concatenated with SYS1.LINKLIB via a LNKLST member.

**Internal Parameters:** Not applicable.

**Member Name: IEADMP00**

**Status:** New for VS2 Release 2

### **Use of the Member**

IEADMP00 contains IBM defaults and/or installation parameters for ABDUMP, for use when an ABEND dump is written to a SYSUDUMP data set. (For ABDUMP parameters associated with a SYSABEND data set, see description of member IEAABD00.)

ABDUMP parameters for a SYSUDUMP data set may be specified in any of three ways:

- A parameter list pointed to by the DUMPOPT keyword of an ABEND macro instruction.\* The list can be built by using the list form of the SNAP macro. (See ABEND and SNAP macros in *OS/VS2 Supervisor Services and Macro Instructions* for details.)
- Default options contained in IEADMP00. These options control the content of the ABEND dump if the programmer does not include the DUMOPT keyword and its associated parameter list with his ABEND, CALLRTM, or SETRP macro. If the DUMPOPT keyword is included, the specified options are merged with those in IEADMP00.
- Temporary override options that an operator can specify with a CHNGDUMP command. (See warning statement and example of misuse of CHNGDUMP in the description of member IEAABD00. For format and general usage information on CHNGDUMP, refer to *Operator's Library: OS/VS2 Reference (JES2)*.)

During IPL an information message will notify the operator if IEADMP00 is invalid or can't be found. No prompting of the operator will occur. If the member contains both valid and invalid parameters, an information message will indicate the valid options that were accepted before the error occurred.

**Parameter in IEASYSxx:** None  
(or specified by the operator)

### **Syntax Rules**

Same as for member IEAABD00. See the description of that member.

---

\*An ABEND dump can also be invoked by the CALLRTM and SETRP macros. (For details, see *OS/VS2 System Programming Library: Supervisor*.)

## IEADMP00 (continued)

### IBM-Supplied Defaults

The following defaults are automatically placed in IEADMP00 by sysgen.

SDATA=(CB,ENQ,TRT),PDATA=(ALLPA,SPLS)

These options request a dump of the following areas:

- formatted control blocks for the task
- formatted enqueue control blocks for the task
- modules on the link pack area queue and the job pack area queue, and SVC modules related to the task
- user storage allocated for the task
- GTF trace or supervisor incore trace (see explanation under TRT parameter)

### Internal Parameters

#### Parameter

#### Meaning and Use

SDATA=

ALLSDATA      All the following options are automatically specified.

The following parameters request dump of specific SDATA areas, as indicated:

NUC	Control program nucleus. SQA, LSQA and PSA are included.
SQA	The system queue area.
LSQA	Local system queue area for the address space, including subpools 229 and 230.
SWA	Scheduler work area used for the failing task.
CB	Control blocks related to the failing task.
ENQ	Enqueue control blocks (QCBs and QELs) related to the failing task.
TRT	GTF or supervisor trace table depending on whether the ABEND occurs after or during IPL, and whether the TRACE ON command was issued at IPL. After Master Scheduler Initialization, GTF data is displayed if GTF has been started. If GTF is not running, and the TRACE ON command was issued at IPL, the supervisor trace table is displayed. (For the use of the TRACE command, see <i>Operator's Library: OS/VS2 Reference (JES2)</i> .)

## IEADMP00 (continued)

Parameter	Meaning and Use
-----------	-----------------

PDATA=

ALLPDATA	All the following options are automatically specified.
----------	--

The following parameters request dump of specific PDATA areas, as indicated:

PSW	Program status word at entry to ABEND.
-----	--

REGS	Contents of general registers at entry to ABEND.
------	--

SA or SAH	SA requests save area linkage information and a backward trace of save areas. This option is automatically selected if ALLPDATA is specified.
-----------	---

SAH requests only save area linkage information.

JPA	Contents of the job pack area that relate to the failing task. These include module names and contents.
-----	---

LPA	Contents of the LPA related to the failing task. These include module names and contents. Also includes active SVCs related to the failing task.
-----	--

ALLPA	Contents of both the job pack area and the LPA, as they relate to the failing task, plus SVCs related to the failing task.
-------	--

SPLS	User storage subpools (0–127) related to the failing task.
------	--



**Member Name:** IEAFIXxx (the “fix” list or fixed LPA list)

**Status:** Introduced with VS2 Release 1

### Use of the Member

IEAFIXxx contains the names of modules from LPALIB, SVCLIB, and LINKLIB which will be temporarily fixed in real storage. The duration of this residence is the life of the current IPL. The member may be used to temporarily add or replace SVC or ERP routines that already exist in the pageable LPA, or which should be fixed to improve system performance. Like the temporary modules chosen through the MLPA option, fixed LPA modules may not be automatically reactivated by a “quick-start” IPL. That is, the fixed LPA can be reestablished only by respecification of the FIX parameter at the quick-start IPL. (A quick-start IPL is one at which the CLPA parameter is not specified.)

Since fixed modules are not paged, I/O time and paging overhead can be saved by placing in the fixed LPA moderately used modules from SYS1.LPALIB. When a module is requested, the Program Manager searches the list of fixed routines before it examines the LPA directory on auxiliary storage. The price for this performance improvement is the reduction in real storage available for paging old jobs and starting new jobs. Therefore, the fixed LPA should not be made too large, if real storage is relatively small (2 megabytes or less). Remember that frequently referenced pages will tend to remain in real storage even when they are not fixed.

You may, however, use the fixed LPA to buy reduced page-fault overhead at the expense of some real storage. Such tradeoff would be desirable with a system that tends to be CPU bound but which has sufficient real storage. You can implement the tradeoff by placing moderate-usage LPALIB modules in the fixed LPA (via parmlib member IEAFIXxx), instead of in the pageable LPA. High usage PLPA modules probably need not be fixed, since they are referenced frequently enough to remain in real storage anyway. Since now less real storage will be available for pageable programs, the System Resource Manager will swap out address spaces that would otherwise occupy real core, awaiting CPU availability.

Modules specified in IEAFIXxx are loaded and fixed in the order in which they are named in the member, and are packed without respect to page boundaries. The modules' CDEs are placed on the active LPA queue for easy access by the Program Manager. (*Note:* To keep search time within reasonable limits, do not allow the fixed link pack area to become excessively large.) If the first load module of a type 3 or 4 SVC routine is specified, the SVC table is updated as required.

Fixed LPA modules may optionally be specified at sysgen by means of the RESIDENT keyword of the DATASET macro. Be careful not to specify the same module name in both the RESIDENT and the MEMBERS keywords, since the two keywords are mutually exclusive. These modules will be listed in member IEAFIX00. To specify this list at IPL, the IEASYSxx member or the operator should include FIX=00 as a system parameter.

## IEAFIXxx (continued)

*Note:* There is a maximum size that the fixed area of real storage can occupy. The maximum space taken up by the nucleus, the fixed BLDL table, the fixed LPA, RMS, page frame table, etc. cannot exceed half of real storage, up to a maximum of 2 megabytes. (For information on how to compute the space occupied by fixed areas of real storage, refer to the *Storage Estimates* manual.)

**Parameter in IEASYSxx:**       $\text{FIX} = \left\{ \begin{array}{l} \text{aa} \\ (\text{aa}, \text{bb}, \dots) \end{array} \right\}$   
(or specified by the operator)

The two alphanumeric characters aa, bb, etc. are appended to IEAFIX to form the name of one or more IEAFIXxx members of SYS1.PARMLIB. NIP defaults to no fixed LPA, if you fail to specify the option in one of the following ways:

- FIX keyword included in the IEASYSxx member.
- FIX keyword entered by the operator at IPL.
- RESIDENT keyword of DATASET macro specified at sysgen.

## Syntax Rules

These rules apply to the creation of IEAFIXxx through the IEBUPDTE utility:

- The first record should start with the name of the data set from which modules will be taken (SYS1.LINKLIB, SYS1.SVCLIB, or SYS1.LPALIB), followed by at least one blank.
- The module names from the specified data set follow the blank(s). Names are separated by commas. They need not be in collating sequence. Names may be either major or alias names.
- To continue the list, place a comma and at least one blank after the last module name on all records except the last. Omit the data set name on following record(s) until the data set name changes.
- Place the next data set name at the start of a new record, followed by at least one blank and a new string of module names.
- Do not place a comma after the last module name of the last data set.

### *Syntax Example*

1st record	SYS1.LINKLIB	IKJPARS,IKJPARS2,IKJSCAN, IKJEFD00,IKJDAIR,
2nd record	SYS1.SVCLIB	IGC0009C,
3rd record	IGC09301, IGC09302, IGC09303	

**IBM-Supplied Defaults:** None

## Internal Parameters

This category is not applicable, since module names, not parameters, may appear in this member.

**Member Name: IEAIPSxx**

| **Status:** Introduced with VS2 Release 2

**Use of the Member**

| Used by the system resources manager (see chapter entitled “The System Resources Manager”.)

**Parameter in IEASYSxx: IPS=xx**  
(or specified by the operator)

| The two-character identifier xx is appended to IEAIPS to specify one of several possible parmlib members from which the system resources manager will obtain its parameters. (For additional information, see “IPS” in the description of the IEASYSxx member, and Part 3: “The System Resources Manager.”)

**Syntax Rules**

| See “Part 3: The System Resources Manager”.

**This blank leaf represents pages 64 - 71 which were deleted by Supervisor Performance #2.**



**Member Name:** IEALOD00 (the “Load List”)

**Status:** Introduced with VS2 Release 1

### **Use of the Member**

IEALOD00 contains a list of names of the more frequently used modules of the pageable LPA. For each name on the list, NIP builds a contents directory entry (CDE) on the active LPA queue in fixed storage. After Initialization, when one of the listed modules is requested, the Program Manager searches the active LPA queue, finds the module name, and thus avoids a search of the LPA directory in pageable storage. Since paging overhead is reduced, performance is improved. Frequently and moderately used LPA modules are reasonable candidates for this member.

*Note:* Modules in the fixed LPA and modules in the LPA extension (MLPA) also can be found without a search of the LPA directory. See the descriptions of the IEAFIXxx and IEALPAXx members. For further performance improvements, refer to the IEABLDxx and IEAPAK00 member descriptions, and to the topic “The Pageable Link Pack Area: Its Advantages and Uses” in the Performance Factors chapter.

**Parameter in IEASYSxx:** None  
(or entered by the operator)

### **Syntax Rules**

The following rules apply to the creation of IEALOD00 by means of the IEBUPDTE utility:

- Separate names of modules with commas.
- Indicate continuation by a comma after the last module on a record.
- Module names may be either major names or alias names, but they should be the same names that appear in SYS1.LPALIB.

**IBM-Supplied Defaults:** None

**Internal Parameters:** Not applicable.

**Member Name: IEALPAXx** (Modified LPA List or LPA Extension)

**Status:** Introduced in VS2 Release 1

### Use of the Member

IEALPAXx contains the names of reenterable modules that NIP will load from LINKLIB, SVCLIB, and LPALIB as a temporary extension to the existing pageable LPA. The extension is temporary in that the modules will remain in the paging data sets and be listed on the active LPA queue only for the life of the current IPL. They may *not* be automatically quick-started, i.e., reinstated without respecification of the MLPA parameter. Note that both the LPA extension and the fix list modules (those named in IEAFIXxx) avoid a search of the LPA directory by the Program Manager when one of the modules is requested. The LPA extension, unlike the fix list, however, contains *pageable* modules which behave in most respects like LPA modules.

You may use IEALPAXx to temporarily add or replace SVC or ERP routines. Another possible application would be the testing of replacement LPA modules that have been altered by PTFs.

Modules that have been replaced via IEALPAXx are not physically removed from the pageable LPA or from the LPA directory. They are, however, logically replaced since when one of them is requested, the Program Manager searches the active LPA queue (a chain of CDEs), finds the name of the temporary replacement, and does not examine the LPA directory which contains the name of the replaced module.

If the first load module of a type 3 or 4 SVC routine is added or replaced, the SVC table is updated as required.

**Parameter in IEASYSxx:**       $MLPA = \left\{ \begin{matrix} aa \\ (aa, bb. . .) \end{matrix} \right\}$   
(or specified by the operator)

The two alphameric characters xx are appended to IEALPA to specify the name of one or more modified-LPA list members of parmlib. Since the modified LPA is not a permanent addition to the LPA, you should probably specify MLPA=xx,etc. in an alternate system parameter list (IEASYSxx) and not place the parameter in IEASYS00. Alternately, you may have the operator enter the parameter from the console.

The following is an example of the use of an alternate system parameter list to specify the MLPA parameter:

NIP issues the message IEA101A SPECIFY SYSTEM PARAMETERS FOR  
RELEASE 02.00.VS2

## IEALPaxx (continued)

The operator responds with SYSP=01 to specify the system parameter list IEASYS01.

IEASYS01 contains: . . . , MLPA=00, . . .

NIP reads parmlib member IEALPA00.

## Syntax Rules

The following rules apply to the creation of IEALPaxx by means of the IEBUPDTE utility:

- Place the data set name (e.g., SYS1.SVCLIB) first on the initial record, followed by at least one blank.
- After the blank, list the modules to be loaded, separated by commas. You may use all columns except 72 through 80.
- Indicate continuation of the list by a comma after the last name on all records, except the last.
- Do not include the data set name on continued records.
- Place a *new* data set name at the beginning of a record.
- You may use either major or alias names, or both.
- End the last record with one or more blanks.

### *Syntax Example*

1st record	SYS1.LINKLIB	IKJPARS,IKJPARS2,IKJSCAN, IKJEFD00,IKJDAIR,
2nd record	SYS1.SVCLIB	IGC0009C,
3rd record	IGC09301, IGC09302, IGC09303	

**IBM-Supplied Defaults:** None

**Internal Parameters:** Not applicable.



**Member Name: IEAOPTxx**

| **Status:** Introduced with VS2 Release 2

**Use of the Member**

IEAOPTxx contains three parameters that affect swapping decisions by the system resources manager (SRM). Two parameters (called resource factor coefficients or RFC) are used to weight the recommendations produced by the CPU load balancing and I/O load balancing routines. The third parameter (a resource manager constant or RMC) is called the Enqueue Residence Value (ERV). It determines the minimum period, in terms of CPU execution time, during which the SRM attempts to avoid swapping an address space that is enqueued on a system resource for which there is contention. For additional information, see “Part 3: The System Resources Manager”.

The specification of the entire IEAOPTxx member is optional. Likewise, the specification of each category in the member and of each parameter within a category is optional.

**Parameter in IEASYSxx: OPT=xx**  
(or specified by the operator)

The two-character identifier xx is appended to IEAOPT to specify one of several possible parmlib members that can contain parameters used by the resource algorithms of the system resources manager. (For additional information, see “OPT” in the description of the IEASYSxx member.)

**Syntax Rules**

| See “Part 3: The System Resources Manager”.

**This blank leaf represents pages 76 - 77 which were deleted by Supervisor Performance #2.**



**Member Name:** IEAPAK00 (LPA pack list)

**Status:** Introduced with VS2 Release 1

### Use of the Member

IEAPAK00 contains the names of groups of modules in SYS1.LPALIB that are executed together or in sequence. (As an example, the load modules of a type-4 SVC routine are called sequentially and executed as a group.) The member is used only during a "cold" start (CLPA specified), when the PLPA is loaded from LPALIB.

NIP uses this member to determine the order in which all modules listed in IEAPAK00 are to be loaded from SYS1.LPALIB into the pageable LPA. These modules are packed together, if possible, on a single page. The purpose is to reduce page faults. The LPA can greatly contribute to page faults, since it is highly used. The IEAPAK00 list can significantly reduce page faults (and disk arm movement).

Each group ideally should not exceed 4K bytes in size. If a group exceeds 4K, the module in the group that causes 4K to be exceeded, and all later modules in the same group, will be loaded at the next page boundary in the LPA. In contrast, NIP loads other modules (those not listed in IEAPAK00) in size order, the largest modules first, then the smaller modules. Unused spaces within page boundaries are filled, if possible, with modules smaller than 4K. (For information on how the LPA is loaded, and LPA performance information, see the topic "The Pageable Link Pack Area: Its Advantages and Uses" in the System Performance Factors chapter. See module sizes in *OS/VS2 System Programming Library: Storage Estimates*.)

There are no alternate members for IEAPAK00. If the member does not exist, NIP continues processing without it.

You should select link pack area programs for inclusion in the system pack list to reduce the number of page faults from the pageable link pack area and thereby enhance system performance. The affinity of programs for each other and the size of the programs determine which should be selected for a pack list entry. Program affinity means that one program will usually reference another program when the first program is invoked. By putting programs that reference each other into the same pack list entry, and thereby on the same page, extra page faults are avoided, because the programs are always in real storage together.

Very large programs which are to be put into the link pack area should be link edited so that modules which have affinity\* for other modules are placed within the same page. The linkedit ORDER statement can be used to group CSECTs into pages. This process will accomplish the same goal as the pack list entries, since page faults during execution will be reduced.

---

\*The modules are either executed at the same time or call each other.

## **IEAPAK00 (continued)**

### **Parameter in IEASYSxx** (or specified by the operator)

None, although the member is used only when the CLPA parameter is specified, or at the first IPL after sysgen.

### **Syntax Rules**

The following rules apply to the creation of IEAPAK00 by means of the IEBUPDTE utility:

- The member consists of “groups” or entries containing load module names. Each group is enclosed in parentheses. For example: (IGG019CM, IGG019CN, IGG019CO, IGG019CP, IGG019CR).
- The modules listed in each group should not exceed 4K bytes in total size.
- Separate modules within each group by commas.
- Separate each group from the next by a comma after the closing parenthesis.
- Do not use alias module names. NIP processes only major names.
- All named modules must be reentrant, since LPA modules must have this attribute.

### **IBM-Supplied Default**

A minimum LPA pack list is copied at sysgen. The list is supplied as four alternate members in the APARMLIB data set:

IEAPAKTS — time sharing and batch system (without VTAM)  
IEAPAKBA — batch-only systems (without VTAM)  
IEAPAKBV — batch-only systems with VTAM  
IEAPAKTV — time sharing and batch systems with VTAM

After sysgen, the SYS1.PARMLIB data set contains all four members plus IEAPAK00. IEAPAK00 is a copy of the member that represents the generated system. For example, IEAPAK00 is the same as IEAPAKTS if the system is generated for TSO without VTAM, or is the same as IEAPAKBA if the system is generated for batch only without VTAM, and so forth. The system will use only member IEAPAK00. The other four members are available in SYS1.PARMLIB for possible future use by the installation.

*Note:* The default pack lists are general purpose lists that may not satisfy all installations. Through tracing and other system measurements you may decide on changes that best suit your particular needs.

## IEAPAK00 (continued)

Figures 2-6 and 2-7 list the contents of two of the IBM-supplied default pack lists: IEAPAKTS and IEAPAKBA. The two lists are identical except for the TSO entries at the bottom of Figure 2-6. Figure 2-8 defines the functional module groups that comprise the pack list entries in Figures 2-6 and 2-7. Circled numbers cross relate the module groups with the related list entries.

The VTAM IBM-supplied default pack lists, IEAPAKBV and IEAPAKTV, are extended versions of IEAPAKBA and IEAPAKTS respectively, with the addition of the VTAM modules. IEAPAKBV and IEAPAKTV, along with guidelines for repackaging their contents, are described in *OS/VS2 System Programming Library: VTAM*.

---

```
./ .NAME=IEAPAKTS,LIST=ALL      TSO      IEAPAK00
```

---

(1) (IRBMFTCH,IRBMFEVT,IRBMFEDV,IRBMFECH,IRARMWAR,IRARMSET),  
(1A) (IGG019CM,IGG019CN,IGG019CO,IGG019CP,IGG019CR),  
(4) (IGG019KU,IGG019LI,IGG019BI,IGG019BM), (6) (IGG0199F,IGG0199G,IGG0199W,  
IGG0198L), (10) (IGG019DK,IGG019BB), (26) (IGC0005B,IGC0205B,IGC0505B),  
(27) (IGC0605B,IGC0705B,IGC0805B), (28) (IGC0G05B,IGC0G95B,IGC0I05B),  
(29) (IGC0J05B,IGC0H05B,IGC0W05B,IGC0U05B), (30) (IGC0K05B,IGC0L05B,IGC0P05B),  
(31) (IGC0S05B,IGC0M05B,IGC0V05B), (32) (IGC0N05B,IGC0R05B,IGC0T05B),  
(36) (IGC0006C,IGC0106C,IGC0206C), (37) (IGC0D06C,IGC0F06C,IGC0G06C),  
(38) (IGC0H06C,IGC0A06C), (38A) (IGC0N06C,IGC0Q06C,IGC0S06C), (39) (IGC0008F,  
IGG0860A,IGG0860B,IGG0860C), (40) (IGG0860D,IGG086AE,IGC0008H), (41) (IGC0008B,  
IGC0108B), (42) (IGC0208B,IGC0308B,IGG019P7,IGG019P8,IGG019P9), (43) (IEEVDEV,  
IECVIOPM), (44) (IGC0008A,IGG08101,IGG08102), (45) (IGG08103,IGG08104,IGC0010E),  
(46) (IGC0009H,IGC0109H), (48) (IGC0008C,IEFU83),  
(49) (IGC0011,IGC40110,IGC10110,IGC11110,IGC12110),  
(50) (IGC20110,IGC21110,IGC22110,IGC23110), (51) (IEEVMNT1,IEEPRWI2,IEEPRTN),  
(52) (IEFJDSNA,IEFJJTRM,IEFJRASP,IEFJRECM,IEFJSDTN,IEFJSREQ,IEFIRECM),  
(53) (IEFVGM1,IEFVGM78,IEFVGM2,IEFVGM3,IEFVGM4,IEFVGM5), (54) (IEFVGM6,  
IEFVGM7,IEFVGM70,IEFVGM19,IEFVGM76,IEFVGM8,IEFVGM17,IEFVGM9),  
(55) (IEFVGM10,IEFVGM11,IEFVGM12,IEFVGM13,IEFVGM14,IEFVGM15,IEFVGM16,  
IEFVGM18),  
(58) (IGG0196S,IGG019T3,IGG019T4,IGG019T5,IGG019T6,IGG019T7,IGG019T8, TSO  
IGG019TX,IGG019TY,IGG019TZ)

*Note:* The circled numbers are used as references to the functional module groups listed in Figure 2-8. These numbers are *not* contained in the pack list itself.

Figure 2-6. Default Pack List for Time Sharing and Batch System

## IEAPAK00 (continued)

---

./ NAME=IEAPAKBA,LIST=ALL BATCH (NON-TSO) IEAPAK00

① (IRBMFTCH,IRBMFEVT,IRBMFEDV,IRBMFECH,IRARMWAR,IRARMSET),  
①A (IGG019CM,IGG019CN,IGG019CO,IGG019CP,IGG019CR),  
④ (IGG019KU,IGG019LI,IGG019BI,IGG019BM), ⑥ (IGG0199F,IGG0199G,IGG0199W,  
IGG0198L), ⑩ (IGG019DK,IGG0198B), ②⑥ (IGC0005B,IGC0205B,IGC0505B),  
②⑦ (IGC0605B,IGC0705B,IGC0805B), ②⑧ (IGC0G05B,IGC0G95B,IGC0I05B),  
②⑨ (IGC0J05B,IGC0H05B,IGC0W05B,IGC0U05B), ③⑩ (IGC0K05B,IGC0L05B,IGC0P05B)  
③① (IGC0S05B,IGC0M05B,IGC0V05B), ③② (IGC0N05B,IGC0R05B,IGC0T05B),  
③⑥ (IGC0006C,IGC0106C,IGC0206C), ③⑦ (IGC0D06C,IGC0F06C,IGC0G06C),  
③⑧ (IGC0H06C,IGC0A06C), ③⑧A (IGC0N06C,IGC0Q06C,IGC0S06C), ③⑨ (IGC0008F,  
IGG0860A,IGG0860B,IGG0860C), ④⑩ (IGG0860D,IGG086AE,IGC0008H),  
④① (IGC0008B,IGC0108B), ④② (IGC0208B,IGC0308B,IGG019P7,IGG019P8,IGG019P9),  
④③ (IEEVDEV,IECVIOPM), ④④ (IGC0008A,IGG08101,IGG08102), ④⑤ (IGG08103,  
IGG08104,IGC0010E), ④⑥ (IGC0009H,IGC0109H), ④⑧ (IGC0008C,IEFU83),  
④⑨ (IGC0011,IGC40110,IGC10110,IGC11110,IGC12110),  
⑤⑩ (IGC20110,IGC21110,IGC22110,IGC23110), ⑤① (IEVMNT1,IEEPRWI2,IEEPRTN),  
⑤② (IEFJDSNA,IEFJJTRM,IEFJRASP,IEFJRECM,IEFJSDTN,IEFJSREQ,IEFIRECM),  
⑤③ (IEFVGM1,IEFVGM78,IEFVGM2,IEFVGM3,IEFVGM4,IEFVGM5), ⑤④ (IEFVGM6,  
IEFVGM7,IEFVGM70,IEFVGM19,IEFVGM76,IEFVGM8,IEFVGM17,IEFVGM9),  
⑤⑤ (IEFVGM10,IEFVGM11,IEFVGM12,IEFVGM13,IEFVGM14,IEFVGM15,IEFVGM16,  
IEFVGM18)

*Note:* The circled numbers are used as references to the functional module groups listed in Figure 2-8. These numbers are *not* contained in the pack list itself.

---

Figure 2-7. Default Pack List for Batch System

## IEAPAK00 (continued)

Note: Numbers 2, 3, 5, 7-9, 11, 12-25, 33-35, 47, 56, 57, and 59-62 have been intentionally omitted, since their modules are no longer included in the PAK list.

---

<b>①</b> <u>MF/1</u> IRBMFTCH IRBMFEVT IRBMFEDV IRBMFECH IRARMWAR IRARMSET	<b>②6</b> <u>Checkpoint/Restart</u> IGC0005B IGC0205B Restart IGC0505B
<b>①A</b> <u>Translation Tables</u> IGG019CM TTY,ASCII IGG019CN Burroughs, Friden, NCR IGG019CO IGG019CP IGG019CR	<b>②7</b> <u>Checkpoint/Restart</u> IGC0605B IGC0705B IGC0805B
<b>④</b> <u>BDAM-BSAM</u> IGG019KU BDAM CE Appendage IGG019LI BDAM Check IGG019BI BSAM Check IGG019BM BSAM EOE Appendage	<b>②8</b> <u>Checkpoint/Restart</u> IGC0G05B IGC0G95B IGC0I05B
<b>⑥</b> <u>SAM – SI</u> IGG0199F IGG0199G IGG0199W IGG0198L	<b>②9</b> <u>Checkpoint/Restart</u> IGC0J05B IGC0H05B IGC0W05B IGC0U05B
<b>⑩</b> <u>SAM SI</u> IGG019DK IGG0198B	<b>③0</b> <u>Checkpoint/Restart</u> IGC0K05B IGC0L05B IGC0P05B
	<b>③1</b> <u>Checkpoint/Restart</u> IGC0S05B IGC0M05B IGC0V05B
	<b>③2</b> <u>Checkpoint/Restart</u> IGC0N05B IGC0R05B IGC0T05B

---

Figure 2-8. Functional Contents of Default Pack Lists (Page 1 of 3)



**IEAPAK00 (continued)**

<p><b>36</b> <u>Checkpoint</u> IGC0006C IGC0106C IGC0206C</p>	<p><b>37</b> <u>Checkpoint</u> IGC0D06C IGC0F06C IGC0G06C</p>	<p><b>38</b> <u>Checkpoint</u> IGC0H06C IGC0A06C</p>	<p><b>38A</b> <u>Checkpoint</u> IGC0N06C IGC0Q06C IGC0S06C</p>	<p><b>46</b> <u>Protect, Overlay, Identify</u> IGC0009H IGC0109H Protect</p>	<p><b>48</b> <u>SMF</u> IGC0008C IEFU83</p>
<p><b>39</b> <u>ATLAS</u> IGC0008F IGG0860A IGG0860B IGG0860C</p>	<p><b>40</b> <u>ATLAS</u> IGG0860D IGG086AE IGC0008H</p>	<p><b>41</b> <u>IEDASDR</u> IGC0008B IGC0108B</p>	<p><b>42</b> <u>IEDASDR</u> IGC0208B IGC0308B IGG019P7 IGG019P8 IGG019P9</p>	<p><b>49</b> <u>SVC 110</u> IGC0011 IGC40110 IGC10110 IGC11110 IGC12110</p>	<p><b>50</b> <u>SVC 110</u> IGC20110 IGC21110 IGC22110 IGC23110</p>
<p><b>43</b> <u>MP Reconfiguration</u> IEEVDEV Device Subroutine Mainline IECVIOPM IOS Operational Path Test</p>	<p><b>44</b> <u>SETPRT-3211-IMAGELIBs</u> IGC0008A IGG08101 IGG08102</p>	<p><b>45</b> <u>SETPRT</u> IGG08103 IGG08104 IGC0010E</p>	<p><b>51</b> <u>Start/Mount</u> IEEVMNT1 IEEPRWI2 IEEPRTN</p>	<p><b>52</b> <u>JES2 Initiator</u> IEFJDSNA IEFJJTRM IEFJRASP IEFJRECM IEFJSDTN IEFJSREQ IEFIRECM</p>	<p><b>53</b> <u>Interpreter</u> IEFVGM1 IEFVGM78 IEFVGM2 IEFVGM3 IEFVGM4 IEFVGM5</p>
				<p><b>54</b> <u>Interpreter</u> IEFVGM6 IEFVGM7 IEFVGM70 IEFVGM19 IEFVGM76 IEFVGM8 IEFVGM17 IEFVGM9</p>	

Figure 2-8. Functional Contents of Default Pack Lists (Page 2 of 3)

**IEAPAK00 (continued)**

---

<b>55</b> <u>Interpreter</u>	<b>58</b> <u>TSO Terminal I/O Controller (TIOC)</u>
IEFVGM10	IGG0196S
IEFVGM11	IGG019T3
IEFVGM12	IGG019T4
IEFVGM13	IGG019T5
IEFVGM14	IGG019T6
IEFVGM15	IGG019T7
IEFVGM16	IGG019T8
IEFVGM18	IGG019TX
	IGG019TY
	IGG019TZ

---

**Figure 2-8. Functional Contents of Default Pack Lists (Page 3 of 3)**

**Internal Parameters:** Not applicable.

**Member Name: IEASYSxx (System Parameter List)**

**Status:** Introduced with VS2 Release 1

**Use of the Member**

The system programmer can specify system parameters through a combination of three sources: sysgen macros, IEASYSxx members of parmlib, and operator responses to the SPECIFY SYSTEM PARAMETERS message. As many as twelve system parameters may be specified through sysgen macros and automatically copied to the IEASYS00 member of parmlib. Only one of these parameters, the PAGEDSN keyword of the DATASET macro, is mandatory at sysgen. Other parameters, as well as these nine, may be placed in the IEASYS00 member or in an alternate system parameter list (IEASYSxx), to provide a fast initialization with little or no operator intervention.

Figure 2-9 shows which sysgen parameters are always placed in IEASYS00, which parameters are optionally placed in IEASYS00, and which NIP parameters are equivalent to these sysgen parameters.

<b>Sysgen Macro and Parameter</b>	<b>Equivalent Initialization Parameter</b>	<b>Related Parmlib Member (if applicable)</b>	<b>Comments on Sysgen Parameter</b>
<b>CTRLPROG</b>			
CSA=	CSA		Parameter is copied to IEASYS00 only if parameter is specified at sysgen.
OPTIONS=BLDL	BLDL=00 or BLDLF=00	IEABLD00	BLDL=00 or BLDLF=00 is always copied to IEASYS00. Default contents of IEABLD00 are always copied to IEABLD00 from APARMLIB.
PAGNUM	PAGNUM=		Parameter is copied to IEASYS00 only if specified at SYSGEN.
REAL=	REAL=		Parameter is copied to IEASYS00 only if parameter is specified at sysgen.
SQA=	SQA=		Parameter is copied to IEASYS00 only if parameter is specified at sysgen.
VRREGN=	VRREGN=		Parameter is copied to IEASYS00 only if parameter is specified at sysgen.
<b>DATASET</b>			
DUPLEXDS,NAME=	DUPLEX=		Parameter is copied to IEASYS00 only if specified at SYSGEN.
PAGEDSN=	PAGE=		The specification of at least three DATASET macros with PAGEDSN keyword is a required parameter at sysgen, without which the sysgen cannot complete. This specification causes sysgen to place the PAGE parameter and the specified dsnames into IEASYS00. The dsnames specified with PAGEDSN can be changed or added to at IPL.
RESIDNT=	FIX=00	IEAFIX00	FIX=00 is copied to IEASYS00. The member name specified in the RESIDNT keyword is placed in IEAFIX00.
SWAPDSN=	SWAP=		Parameter is copied to IEASYS00 only if specified at SYSGEN.
<b>SCHEDULR</b>			
HARDCOPY=	HARDCPY=		Default value is SYSLOG. Sysgen places this operand in IEASYS00 if HARDCOPY is not specified.

Figure 2-9. Sysgen Parameters that Are Copied to the IEASYS00 Member

## **IEASYSxx (continued)**

The installation has the choice of placing all or some system parameters in the IEASYS00 member, and other parameters, or other values of the same parameters, in one or more alternate IEASYSxx members. IEASYS00 is the likely place to put installation defaults or parameters that will not change from IPL to IPL. The system programmer can add to or modify sysgen-created parameters in this member by means of the IEBUPDTE utility. The alternate IEASYSxx member(s), in contrast, should contain parameters that are subject to change, possibly from one work shift to another.

Use of the IEASYS00 member can minimize operator intervention at IPL. Since IEASYS00 is read automatically, the operator can respond to SPECIFY SYSTEM PARAMETERS by replying END or ENTER or 'U', and need not enter parameters unless an error occurs and prompting ensues.

If a parameter list other than IEASYS00 is desired at a particular IPL, the operator must indicate the alternate list by replying SYSP=xx in response to the SPECIFY SYSTEM PARAMETERS message. (See the SYSP parameter description in this section for additional information.)

The use of system parameter lists in parmlib thus offers two main advantages:

- They shorten and simplify the IPL process by allowing the installation to preselect system parameters.
- They provide flexibility in the choice of system parameters. The operator can specify SYSP=xx to select one of several alternate lists.

The system parameters that sysgen places in IEASYS00 are adequate to initialize the system, although additional parameters are desirable. Figure 2-9 lists the sysgen parameters that are copied to IEASYS00.

## **Overview of IEASYSxx Parameters**

The following list briefly defines all system parameters that can be placed in an IEASYSxx or IEASYS00 member (or specified by the operator). Detailed discussions of these parameters are provided in later sections of the IEASYSxx topic.

*Note:* PAGE and HARDCPY are the only mandatory parameters that have no internal defaults. They must be specified.

**IEASYSxx (continued)**

<b>Parameter</b>	<b>Use of the Parameter</b>
APF	Names the parmlib member (IEAAPFxx) that contains authorized data set names.
APG	Specifies the priority value of the automatic priority group for use by the System Resource Manager.
BLDL/BLDLF	Names the parmlib member (IEABLDxx) that lists module names that are to be placed in a pageable or fixed BLDL table. The choice of BLDL or BLDLF determines which table NIP will build.
CLPA	Tells NIP to load the link pack area with the modules contained in SYS1.LPALIB. Also purges VIO data sets that were used in the previously initialized system. Thus, CLPA implies CVIO.
CMD	Completes the name of the parmlib member (COMMNDxx) that contains commands to be issued internally during Master Scheduler Initialization. COMMNDxx also controls prompting during TOD clock initialization.
CSA	Specifies the size of the common service area in multiples of 1K bytes.
CVIO	Deletes previously used VIO data sets from the paging space. This parameter is automatically included when CLPA is specified.
DUMP	Specifies whether SYS1.DUMP data sets for SVC Dump are to be on direct access device(s) or tape, and names the unit address(es) if tape is to be used. This parameter can also indicate that <i>no</i> SYS1.DUMP data sets are to be made available for SVC dumps.
DUPLEX	Specifies a duplex page data set name or overrides the existing duplex data set name. This parameter is ignored on quick starts or warm starts (non-CLPA IPLs).
FIX	Completes the name of one or more parmlib members (IEAFIXxx) that contain names of modules from SVCLIB, LINKLIB, and LPALIB that are to be placed in a fixed LPA that lasts for the life of the IPL.
HARDCPY	Specifies a hard copy log and indicates the types of commands, responses, and messages that will appear on the log device.
IPS	Completes the name of the parmlib member (IEAIPSxx) from which the system resources manager (SRM) will obtain the installation performance specification.
LNK	Completes the name of one or more parmlib members (LNKLSTxx) that contain names of data sets that are to be concatenated to LINKLIB.
LOGCLS	Specifies the JES output class for the log data sets.
LOGLMT	Specifies the maximum number of WTLs (messages) for a log data set. When the limit is reached, the data set is scheduled for sysout processing.
MAXUSER	Indicates the maximum number of address spaces that can be located by means of the address space vector table (ASVT). Sets a limit on the number of concurrent address spaces.
MLPA	Completes the name of one or more parmlib members (IEALPAxx) that name modules to be placed in a temporary LPA extension.
NUCMAP	Specifies that the Dynamic Support System (DSS) resource initialization module (RIM) should build a new map of the nucleus in the SYS1.DSSVM data set, and overlay the old map, if one exists.
OPI	Indicates whether the operator is to be allowed to override particular parameters, or all parameters, contained in IEASYSxx.
OPT	Completes the name of a parmlib member (IEAOPTxx) that contains tuning parameters to be used by the resource algorithms of the System Resource Manager.

Figure 2-10. Overview of IEASYSxx Parameters (Part 1 of 2)

**IEASYSxx (continued)**

Parameter	Use of the Parameter
PAGE	Gives the names of newpage data sets to be used as additions to or replacements for existing page data sets. The first-named data set is intended for exclusive placement of the PLPA. The second-named data set is for exclusive placement of MLPA and CSA. Replacement is possible only if the parameter is placed in IEASYSxx, and the operator selects this member by entering SYSP=xx. The PAGE parameter, when specified by the operator, can only add temporarily to a parmlib page data set list.
PAGNUM	Specifies the number of page and swap data sets that can be added to the system after IPL by the PAGEADD command.
PURGE	Demounts all Mass Storage System volumes.
REAL	Specifies the maximum amount of real storage, in 1K blocks, that can be allocated for concurrent ADDRSPC=REAL jobs.
RSU	Specifies the number of storage units that will be available for storage reconfiguration in an MP system.
SMF	Specifies a parmlib member (SMFPRMxx) from which SMF will obtain its parameters.
SQA	Gives the number of additional 64K segments of the virtual system queue area to be created at IPL.
SWAP	Specifies the names of new swap data sets to be used as additions to or replacements for existing swap data sets.
SYSP	Specifies one or more alternate system parameter lists (IEASYSxx) that are to be read by NIP in addition to IEASYS00. SYSP may be specified only by the operator.
VAL	Names one or more parmlib members (VATLSTxx) that contain "mount" and "use" attributes of direct access devices.
VRREGN	Gives the default real-storage region size for an ADDRSPC=REAL job step that does not have a REGION parameter in its JCL.
WTOBFRS	Specifies the number of buffers that the Write-to-Operator (WTO) routines will use for operator messages.
WTORPLY	Specifies the number of operator reply elements to be used by the Write-to-Operator-with-Reply (WTOR) routines.

Figure 2-10. Overview of IEASYSxx Parameters (Part 2 of 2)

**Changes to Initialization Parameters**

The following lists briefly describe MVT system parameters that are no longer supported, and VS2 Release 1 parameters that have changed or that are no longer supported.

## 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 MVS because the Initiator modules are contained in the pageable link pack area.





## **IEASYSxx (continued)**

### **MOD**

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

### **RAM**

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

### **RERP**

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

### **RSVC**

In MVS, 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 Manager determines the acceptable time slices.

**The following MVT system parameters are not supported in MVS and have no comparable MVS functions:**

### **ALTSYS**

Dynamic device reconfiguration (DDR) for the system residence device is not supported.

### **HRAM**

MVT hierarchy support is not needed in MVS.

### **HSVC**

MVT hierarchy support is not needed in MVS.

### **MPS**

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

## **Changed MVS System Parameters**

### **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.DUMPxx data sets can be specified in MVS.

**IEASYSxx (continued)****PAGE**

Values specified by the PAGE parameter have changed for MVS, because paging space is made up of preformatted VSAM data sets, at least three of which must be defined and formatted either before or during system generation. The first-named data set is intended to hold the PLPA. The second-named data set is for exclusive placement of common areas excluding PLPA (MLPA and CSA). The third and subsequent-named data sets are for non-duplexed areas (that is, private address spaces and VIO data sets). The PAGE parameter no longer specifies a particular volume or unit to contain the link pack area.

**REAL**

In MVS, the value specified by the REAL parameter represents the total amount of ADDRSPC=REAL 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 MVS functions.

**LSQACEL**

This parameter is not needed in MVS because quickcells for the local system queue area are controlled by the Virtual Storage Manager.

**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 the Virtual Storage Manager.

**TMSL**

This parameter is not needed because the System Resource Manager determines 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 each time-sharing user is assigned to an individual virtual address space.

**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.

## IEASYSxx (continued)

### TRACE

The system trace option cannot be specified in IEASYS00 or IEASYSxx in MVS. It is activated automatically by the system control program during IPL. The system trace is continued after IPL if the TRACE ON command is issued at IPL before the operator responds to the JES2 SPECIFY OPTIONS message.

### TSOAUX

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

### Parameter Specified by the Operator: SYSP=xx

The operator specifies this parameter to indicate an alternate system parameter list to be used by NIP in addition to IEASYS00. (See SYSP in “Internal Parameters” later in this section.)

### Syntax Rules

The following rules apply to the creation of IEASYSxx by means of the IEBUPDTE utility:

- Use columns 1 through 71 of each record for parameters. Leading blanks are acceptable.
- Do not use columns 72 through 80, since NIP ignores these columns.
- Separate parameters with commas.
- Indicate continuation by a comma, followed by at least one blank.
- Begin subsequent records in any column.
- Enclose multiple subparameters in parentheses. The number of subparameters is unlimited.

#### *Syntax Example:*

```
. . .DUMP=(TA,282),MLPA=(00,01,02,03,L),BLDLF=02
```

### IBM-Supplied Defaults

The default member IEASYS00, initially created at sysgen, always contains at least BLDL=00 or BLDLF=00, HARDCPY, and PAGE. Additionally, particular parameters of the CTRLPROG, DATASET, and SCHEDULR sysgen macros are copied to IEASYS00, if the installation specifies them during system generation. (See Figure 2-11 for details.)

### Internal Parameters

The IEASYSxx parameters are listed alphabetically and are individually described. These parameters may optionally be issued by the operator, although such manual issuance would slow the IPL.

**NOTE:** “Value range”, if applicable, means the syntactically acceptable range of values, not necessarily a range of values reasonable for function or performance. The “associated parmlib member” refers to the parmlib member that is named by the parameter. For example, IEAAPF01 is named by the APF=01 parameter in IEASYSxx, or entered by the operator.

## **IEASYSxx (continued)**

**Parameter:** APF=xx

**Meaning and Use:** The two alphanumeric characters xx are appended to IEAAPF to form the name of parmlib member IEAAPFxx. This member lists the data set names and volume serial numbers of authorized data sets. SYS1.LINKLIB and SYS1.SVCLIB are automatically included as authorized data sets. SYS1.LPALIB is not automatically authorized, since it is used only by NIP.

The default parmlib member IEAAPF00 is created at sysgen if the system programmer specifies the APFLIB keyword of the CTRLPROG macro.

**Value Range:** Not applicable.

**Default Value:** SYS1.LINKLIB and SYS1.SVCLIB alone are authorized if APX=xx is not specified. In addition, any libraries concatenated to SYS1.LINKLIB are also authorized.

**Associated Parmlib Member:** IEAAPFxx

**IEASYSxx (continued)****Parameter:** APG=nn

**Meaning and Use:** The one or two-digit number nn specifies the priority value of the automatic priority group (APG). Job steps which do not include the DPRTY keyword in their EXEC statements are automatically placed in the APG. DPRTY is specified as DPRTY=(value1,value2). Job steps for which value1 = APG priority will also be in the APG group. Job steps for which value1 is greater than APG will have a higher priority than APG-group job steps. Job steps for which value1 is less than APG will have a lower priority than APG-group job steps. (Value2 is not used for APG. See *OS/VS2 JCL* for a description of value2.) For further information, see "Part 3: The System Resources Manager."

**Value Range:** 0–13**Default Value:** 7**Associated Parmlib Member:** None

**IEASYSxx (continued)**

**Parameter:** BLDLF=xx or BLDL=xx

**Meaning and Use:** The two alphameric characters xx are appended to IEABLD to form the name of a parmlib member. This member contains a list of module names for which NIP builds a list of BLDL entries in the resident BLDL table.

The installation specifies BLDLF if it wishes the BLDL table to be fixed in real storage. Otherwise, it specifies BLDL and the table is pageable. The two keyword options are mutually exclusive. If by error, both BLDL and BLDLF are specified, NIP rejects BLDL, accepts BLDLF, and issues a warning message.

*Note:* At a relatively small expense in real storage, the BLDLF parameter can improve performance by reducing the number of page faults.

**Value Range:** Not applicable

**Default Value:** BLDLF=00 or BLDL=00, since IEASYS00 is always read and contains this default.

**Associated Parmlib Member:** IEABLDxx (See description of this member.)

**IEASYSxx (continued)****Parameter: CLPA (Create Link Pack Area)**

(See also the **MLPA** parameter for temporary additions to the LPA, and the **CVIO** parameter for the deletion of VIO data sets.)

**Meaning and Use:** This parameter causes NIP to load the LPA with all modules contained in SYS1.LPALIB. Modules listed in IEAPAK00 are packed together, preferably in one-page groups. (See description of IEAPAK00.) Modules not in the pack list (IEAPAK00) are loaded in size order, large modules first, then smaller modules to fill unused space. (For further information, refer to the topic “The Pageable Link Pack Area: Its Advantages and Uses” in the System Performance Factors chapter.)

The LPA resides on external page storage. Only one LPA may exist in paging space, although the same LPA is duplexed on two separate data sets for error recovery if a duplex data set is specified. Since the LPA is a read-only area, all LPALIB modules should be reenterable and refreshable.

CLPA should be specified after the installation has modified LPALIB and wishes to reload the LPA with new or changed modules.

*Note:* CLPA also implies CVIO, so that old VIO data sets are automatically purged. (See the description of CVIO for further information.)

The CLPA parameter is not needed at the *first* IPL. NIP detects the cold start condition internally, noting that the LPA has not been loaded and that VIO data sets do not exist.

If CLPA is not specified, NIP tries to find a usable LPA in the existing page data sets. If NIP is successful, a Quick Start occurs. In Quick Start, the Auxiliary Storage Manager (ASM) obtains the quick start record that describes the existing LPA. It then reestablishes the old LPA. The old LPA may be reused for any number of system initializations, as long as CLPA is not specified. However, page data sets that contain the last used LPA must be mounted. If they are not, the operator is asked to mount them. If he bypasses mounting, ASM Initialization requests a different page data set and forces a “cold” start. NIP then reestablishes the LPA as it does when CLPA is specified. In this cold start, both the previously established LPA *and* existing VIO data sets are logically deleted from paging space.

The fixed LPA, the LPA extension, and the resident BLDL table, however, are not automatically reused in a Quick Start. They must be respecified. Existing VIO data sets are also retained, unless the CVIO or CLPA parameter is specified or forced. (See description of CVIO parameter.)

If CLPA is specified and an LPA already exists on a paging data set, NIP frees the existing LPA and updates the quick start record to reflect the new LPA. NIP loads the LPA from LPALIB, as previously described.

## IEASYSxx (continued)

### Parameter: CLPA (continued)

After NIP has constructed the LPA, it reads and processes the IEALOD00 member of parmlib. This member contains the names of LPA modules that the installation uses frequently. NIP creates contents directory entries (CDEs) for these modules on the active LPA queue. This queue can reduce page faults, because the Program Manager can avoid a search of the LPA directory when one of the modules on this queue is requested. (See the description of the IEALOD00 member, and the topic “The Pageable Link Pack Area: Its Advantages and Uses” in the System Performance Factors chapter.)

*Note:* If you wish to specify a new time interval for the Missing Interrupt Handler (MIH), you must ZAP the affected csect and re-IPL, specifying CLPA. The time interval is the time between checks by MIH for pending conditions. Possible pending conditions include: device ends, channel ends, swaps of Dynamic Device Reconfiguration (DDR), and MOUNT commands.

IBM-supplied csect IGFINTVL provides a time interval of three minutes. To change the interval, you should use the AMASPZAP program to modify the csect. The JCL and the required control statements are described in *OS/VS2 System Programming Library: Supervisor*. The changed time interval will take effect after the next IPL at which the modified csect is loaded. To load the csect, you must specify the CLPA parameter at the IPL. (Keep in mind, however, that CLPA implies CVIO, and existing VIO data sets will be deleted. It's therefore best to do the IPL at a power-on, rather than during a work shift.)

**Value Range:** Not applicable

**Default Value:** Not applicable

**Associated Parmlib Member:** None



## IEASYSxx (continued)

**Parameter:**    **CMD** =  $\left\{ \begin{array}{l} \mathbf{aa} \\ (\mathbf{aa,bb. . .}) \end{array} \right\}$

**Meaning and Use:** The two alphameric characters (e.g., 01, 0B) specify one or more COMMNDxx members of parmlib. The installation can specify multiple members. Each member can contain automatic operator commands that the installation wants executed during Master Scheduler Initialization. Examples of such commands are those that start GTF and TCAM. JES2 commands are not accepted, since automatic commands are processed before JES2 is started.

If the CMD parameter is not specified, the COMMND00 member is used if it exists. If COMMND00 does not exist or cannot be read, Initialization continues without any internally issued commands, and without prompting for TOD clock initialization.

**Value Range:** Not applicable

**Default Value:** CMD=00

**Associated Parmlib Member:** COMMNDxx (See description of this member.)

**IEASYSxx (continued)****Parameter:** CSA=nnnn

**Meaning and Use:** This parameter specifies the maximum size of the virtual common service area (CSA) in multiples of 1K (1024 bytes), rounded to the next segment (64K) boundary.

Example: CSA=200 could provide a CSA of 256K bytes ( $200 \div 64 = 3$  segments + 8K) so that the boundary between the CSA and the user area would be on a segment boundary.

The CSA is an address range in each address space that is used for common system functions (i.e., functions not related to a particular address space). For example, the system allocates buffers for LOG and SMF, as well as work queue elements and reply buffers, from the CSA. The CSA is duplexed for recoverability, as are the other common areas, PLPA and MLPA, when a duplex page data set is being used.

In selecting a value for the CSA parameter, choose a value somewhat larger than you think you need. There are two reasons:

- If the Virtual Storage Manager runs out of SQA, it will try to obtain space from the CSA. If it cannot do so, it will place the system in a disabled wait state.
- A large CSA size will reserve space for future LPA growth. Such growth would be hampered if users were allowed to obtain very large private areas. A large CSA specification effectively limits the maximum private area that a user job can acquire.

**Notes:**

1. You can also specify the CSA parameter by means of the CSA keyword of the CTRLPROG macro at sysgen. In this case, the specified CSA value is automatically placed in member IEASYS00.
2. For storage overviews and for a method of calculating CSA, refer to *OS/VS2 System Programming Library: Storage Estimates*.

**Value Range:** 0–9999

**Default Range:** 100 (This means 128K, since the value is rounded up to the next segment boundary.)

**Associated Parmlib Member:** Not applicable

**IEASYSxx: (continued)**

**Parameter:** CVIO (Clear VIO)

**Meaning and Use:** This parameter specifies that all VIO data sets are to be deleted from page space. A typical application would be the purging of VIO data sets when the system is re-IPLed after a previous end-of-day (EOD).

*Note:* If you wish the Auxiliary Manager to purge and reinitialize *both* VIO data sets *and* the LPA, specify CLPA. CLPA always implies CVIO.

If neither CVIO nor CLPA is specified, VIO data sets will be retained for restart processing. Such restart would be possible for some data sets after a temporary system failure. Reuse of VIO data sets occurs independently of a warm start of the job entry subsystem. That is, if neither CLPA nor CVIO is specified, the Auxiliary Storage Manager reestablishes the VIO data sets that were checkpointed before the system failure. This action, of course, does not ensure that the job entry subsystem will reuse these data sets.

When neither CVIO nor CLPA is specified, one or more volumes that contain VIO pages may possibly not be mounted. The Auxiliary Storage Manager (ASM) requests the operator to mount the missing volume(s). If the operator doesn't mount all the requested volumes, ASM deletes all VIO data sets, just as if CVIO or CLPA had been specified. The operator receives a message that indicates that CVIO has been forced.

**Value Range:** Not applicable

**Default Value:** None

**Associated Parmlib Member:** None

IEASYSxx (continued)

Parameter: DUMP =  $\left\{ \begin{array}{l} \text{NO} \\ \text{DASD} \\ \text{L} \\ (\text{TA}, \text{unitaddr1}, \text{unitaddr2} \dots) \end{array} \right\}$

**Meaning and Use:** This parameter specifies whether SYS1.DUMP data sets for SVC Dump are to be on direct access device(s) or tape, and names the unit address(es) if tape is to be used. Optionally, the parameter can specify (via NO) that no SVC dump data sets are to be made available. (All three options – DASD, L, and TA – may be specified with a single DUMP parameter, if so desired.) SVC Dump options are not included in parmlib. However, the installation can specify the options, if it so desires, by means of the CHNGDUMP operator command.

**Operand Descriptions:**

NO

specifies that no dump data sets will be available for SVC Dump.

DASD

specifies that the currently cataloged SYS1.DUMPnn data sets (if any) on permanently resident direct access volumes, are to be used. This operand is the default if the DUMP parameter is omitted.

L

is used with DASD to specify that cataloged SYS1.DUMPnn data sets are to be listed on the operator's console, with the status of empty or full. *Empty* means the data set is unused or is reusable; *full* means the data set is used and is ready to be printed. A message prompts the operator for the names of full data sets that he doesn't want to print. SVC Dump will reuse these data sets. The operator can also specify in his reply the unit addresses of additional tape units on which dump data can be written.

(TA,unitaddr1,unitaddr2. . .)

specifies that the tape unit at unitaddr is to be used as an SVC Dump data set. If both this operand and the DASD operand are specified, the tape unit(s) are added to the available direct access dump data sets. If DASD is not specified, only the named tape units are made available.

**Examples of Valid DUMP Statements:**

DUMP=NO  
DUMP=DASD  
DUMP=(DASD,L)  
DUMP=(DASD,L,(TA,282,283))  
DUMP=(TA,282)

## IEASYSxx (continued)

### Parameter: DUMP (continued)

**How Dump Data Sets Are Used:** Dump data sets may be all on tape, all on direct access devices, or on a combination of devices. Direct access data sets must be preallocated and cataloged. Eligible device types consist of unlabeled 2400-series tape, 9-track (or tape compatible with 2400-series), and any of the following direct access devices:

- 2314
- 2319
- 3330
- 3330-1
- 2305-1
- 2305-2
- 3340/3344
- 3350

As many as ten dump data sets may be allocated. Direct access data set names must be in the form SYS1.DUMPnn, in which nn may be digits 00 to 09. Since each direct access data set will contain only one dump, the installation should allocate it with sufficient space for the maximum size SVC dump it expects. The dump size depends on the SDATA options that are specified in the console DUMP command, or on the overriding options that are specified in the CHNGDUMP command. (For information on these commands, refer to *Operator's Library: OS/VS2 Reference (JES2)*.)

**Requirements for Dump Data Sets:** Dump data sets must meet the following requirements:

- A data set can reside on only one volume; that is, a data set can't span across two or more volumes.
- Each direct access data set can contain only one dump.
- Direct access data sets must be on permanently resident volumes and be allocated, and cataloged. (For further information on the permanently resident attribute, see the description of the VATLSTxx member of parmlib.)
- DD statements for direct access data sets must not specify a secondary quantity in the SPACE parameter.

**Processing of Dump Data Sets:** The status and type of each dump data set is maintained in an internal table by SVC Dump. The data sets are processed in the order in which they are specified as operands of the DUMP parameter. If, for example, the parameter statement specifies DUMP=(TA,283),DASD,(TA,285), the first table entry would be for (TA,283). This would be followed by entries for cataloged DASD data sets on permanently resident volumes, in the order SYS1.DUMP00 to SYS1.DUMP09. Only ten data sets are processed. The last table entry is for (TA,285), unless the limit of ten data sets has already been reached.

## **IEASYSxx (continued)**

### **Parameter: Dump (continued)**

Each table entry reflects the data set status, empty or full. An empty data set is available for use by SVC Dump. A full data set can either be printed via AMDPRDMP, or optionally reused for a new dump. The data set is reused if the operator replies to message IEA877A by entering the data set name or tape unit address. For example, the operator replies to message IEA877A: R 00, 'DA=(xx,yy),(TA,tt,uuu)'. The characters xx,yy are the last two digits of two direct access data sets, named SYS1.DUMPxx and SYS1.DUMPyy. They are currently full but are to be reused. The characters tt,uuu are the unit addresses of two additional tape drives that are to be used for dump data sets.

The criteria that determines whether a dump data set is empty or full depends on the device type, DASD or tape. A tape is always considered empty. A DASD data set is empty only if the first record is "end of data". Otherwise, the data set is considered full.

SVC Dump examines the data sets, as listed in the table, in sequence. If a data set is empty, SVC Dump uses it and marks the table entry accordingly. If a data set appears full, it is marked as being used and is skipped. When all data sets have been marked in use, SVC Dump reads the first record of each DASD data set to see if it has been emptied (printed) by AMDPRDMP. If so, the first record is "end of data". SVC Dump updates the table to reflect the current status, then reexamines the data sets, starting at the beginning of the table.

**Value Range:** Not applicable

**Default Value:** DASD

**Associated Parmlib Member:** None

## IEASYSxx (continued)

**Parameter:** DUPLEX=dsname

**Meaning and Use:** The DUPLEX parameter allows the user to specify a paging data set to be used to hold a secondary copy of all common area system pages (all pages written to the PLPA or common page data sets). Only one duplex data set may be specified. The duplex parameter is valid only on a cold IPL (CLPA). On either quick or warm IPLs, ASM ignores the DUPLEX parameter and attempts to use the same duplex page data set specified on the cold IPL. If no duplex page data set was then specified, or if it is otherwise unavailable, duplexing is suspended. This parameter follows all of the normal override rules for system parameters.

**How the Duplex Page Data Set Is Specified:** On a CLPA IPL, the duplex page data set is specified from one of two sources: 1) IEASYS00 or IEASYSxx, or 2) the operator-specified DUPLEX parameter. On a non-CLPA IPL, the duplex page data set is specified from only a single source: the temporary page activity reference table (TPARTBLE). An outline of these specifications is as follows:

- CLPA IPLs –
  - *The DUPLEX parameter in IEASYS00:* This data set name is specified along with the DUPLEXDS<sup>1</sup> keyword of the DATASET macro during system generation.
  - *The DUPLEX parameter in IEASYSxx,* an alternate system parameter list: If the operator selects this list (by using the SYSP parameter), the DUPLEX parameter in IEASYSxx overrides the DUPLEX parameter in IEASYS00.
  - *The DUPLEX parameter specified by the operator in the current IPL:* This DUPLEX parameter specification is *not* merged with that specified by IEASYSxx, as is the case with the PAGE parameter specification, but overrides those specifications. The operator specification lasts until the next cold IPL.
- Non-CLPA IPLs –
  - *TPARTBLE:* The temporary page activity reference table occupies the first 8K of the PLPA page data set. It contains previously specified page data sets – including the previous duplex page data set – which will be used for the current IPL.

---

<sup>1</sup>The DUPLEXDS keyword causes the DUPLEX parameter to be placed in IEASYS00.

**IEASYSxx (continued)**

Before an IPL, the page data set specified through the DUPLEX parameter, or through the sysgen DATASET<sup>1</sup> macro, must have been allocated, cataloged in the system master catalog, and preformatted in VSAM format with track overflow. This data set is a normal page data set used for duplexing. The data set can be formatted by using the DEFINE PAGESPACE processor of access method services as described in *OS/VS2 Access Method Services*.

**Syntax Example for the DUPLEX Parameter:**

Example 1: DUPLEX=dsname

When specified on a cold IPL, this statement causes ASM to use the specified data set for duplexing the common area pages.

Note that neither UNIT nor VOLSER is specified. Because all data sets used as paging data sets must be cataloged, ASM initialization does not need externally specified volume serial numbers. The operator may either premount the necessary volumes or await a mount message.

When the duplex data set gets full, ASM issues a message stating that duplexing has been suspended. Once suspended, duplexing can only recommence after a subsequent IPL. (See “Part 2.1: Auxiliary Storage Management Initialization” space requirement guidelines for duplex data sets.)

**Value Range:** Not applicable.

**Default Value:** None.

**Associated Parmlib Member:** None.

---

<sup>1</sup>For information about creating page data sets with the sysgen DATASET macro, refer to *OS/VS2 System Programming Library: System Generation Reference*.



### IEASYSxx (continued)

Parameter:  $FIX = \left\{ \begin{array}{l} aa \\ (aa,bb... ) \end{array} \right\}$

**Meaning and Use:** This parameter specifies one or more IEAFIXxx members of parmlib. The two alphameric characters are appended to IEAFIX to name the member(s). The member(s) contain names of modules from SVCLIB, LINKLIB, and LPALIB that are to be fixed in real storage as a temporary fixed LPA. The fixed LPA modules are active only for the life of the current IPL and may not be automatically reinstated by a Quick Start. You must respecify the FIX parameter in subsequent IPLs if you want to reinstate the fixed LPA.

*Note:* The FIX parameter may also be specified by using the RESIDENT keyword of the DATASET macro at sysgen. In this case, FIX=00 is automatically placed in member IEASYS00, and the library specified by the RESIDENT keyword is listed in parmlib member IEAFIX00. (For additional information on the FIX option, refer to the description of the IEAFIXxx member. For information on the other temporary LPA option, see the descriptions of the MLPA parameter and the IEALPAXx member.)

**Value Range:** Not applicable

**Default Value:** FIX=00, if the installation specified the RESIDENT keyword of the DATASET macro at sysgen.

**Associated Parmlib Member:** IEAFIXxx

## IEASYSxx (continued)

### Parameter:

$$\text{HARDCPY} = \left( \left\{ \begin{array}{l} \text{SYSLOG} \\ \text{address} \end{array} \right\}, \left\{ \begin{array}{l} \text{ALL} \\ (\text{route code 1, route code 2. . .}) \end{array} \right\}, \left\{ \begin{array}{l} \text{CMDS} \\ \text{INCMDS} \\ \text{STCMDS} \\ \text{NOCMDS} \end{array} \right\} \right)$$

**Meaning and Use:** This parameter specifies a hardcopy log, which may record messages, operator commands, and system commands and responses. The parameter is new to IEASYSxx. It was formerly only a parameter of the SCHEDULR sysgen macro. Now it may be specified at either or both sysgen and Initialization.

A hardcopy log is *required* in either of two cases:

- The system has more than one active console,  
or
- The system has an active graphic console.

*Note:* In either of these two cases, routing codes 1 through 4, and 7, 8, and 10 need not be specified, since the system will automatically assign them.

### Operand Descriptions:

#### SYSLOG

specifies the system log as the hardcopy log.

*Note:* If the SYSLOG specification is in effect and the subsystem reaches the WTO buffer limit before completing its processing, a WAIT condition in which hardcopy messages cannot appear on SYSLOG can occur because the subsystem is waiting for message buffers. If this condition does occur, the operator should issue a VARY command to direct hardcopy to another device.

#### address

is the unit address of a hardcopy console that has at least output capability.

*Note:* One of these operands must be specified. If SYSLOG was specified at sysgen in the HARDCOPY parameter of the SCHEDULR macro, the unit address must be specified at Initialization.

The following device types are eligible for the unit address operand:

1052-7	3286-1, 2
1443-N1	1403
3210	2740-1
3215	3211
3213	3284-1, 2

#### ALL

specifies that all WTO and WTOR messages issued with routing codes will be displayed on the hardcopy log.

## IEASYSxx (continued)

### Parameter: HARDCPY (continued)

(route1,route2. . .)

is a list of numbers ranging from 1 to 15 that designate the routine codes of messages that the hardcopy log is authorized to receive. The list must be separated by commas and enclosed in parentheses. If a hardcopy log is required, any of the following routing codes are automatically assigned and therefore need not be specified: 1–4, 7, 8, 10. (See the previous description of conditions under which a hardcopy log is required.)

### CMDS

specifies that operator commands, and system commands and responses, will be displayed. This operand is the default for the various command operands. It is automatically assigned by the system, if there is an active graphics console or there is more than one active console. In either case, a commands operand other than CMDS is ignored.

### INCMDS

specifies that operator commands, system commands, and in-line responses will be displayed.

### STCMDS

specifies that operator commands, system commands, in-line responses, and static status displays will be written out.

### NOCMDS

specifies that no commands or responses are to be displayed.

**Value Range:** Not applicable

**Default Value:** The full set of HARDCPY operands can be in IEASYS00 if the HARDCOPY keyword of the SCHEDULR macro was specified at sysgen. If HARDCOPY was omitted at sysgen, the default in IEASYS00 is only SYSLOG. In this case, you must specify a unit address in the HARDCPY parameter at IPL.

**Associated Parmlib Member:** None

**IEASYSxx (continued)****Parameter:** IPS=xx

**Meaning and Use:** This parameter specifies the particular installation performance specification (IPS) that will be used by the system resources manager.

The two alphameric characters, represented by xx, are appended to IEAIPS to form the name of an IEAIPSxx member of parmlib.

If the member can't be found or contains invalid specifications, the initialization routine of the system resources manager prompts the operator to specify an alternate member, i.e., respecify IPS=xx. If the operator chooses to reply END or ENTER, the System Resource Manager tries to use the default member IEAIPS00. If IEAIPS00 is invalid or unavailable, the System Resource Manager uses an internal set of IPS values, called the "skeleton IPS". The skeleton IPS avoids service rate distinctions among any jobs. It is merely a stopgap IPS intended to permit the completion of IPL.

The operator can select a new IPS (i.e., indicate that the system is to run under the control of an alternate IEAIPSxx member) between IPLs by issuing the SET IPS command. When the SET IPS command is processed, any ongoing transactions that are associated with performance groups whose definitions have changed in the new IPS, will be associated with the first performance group period of the new performance group (that is, the previous transaction is ended, and a new transaction is started, before processing continues). If the performance group with which ongoing transactions were previously associated is not defined in the new IPS, they will be associated with the appropriate default performance group (1 for batch users, 2 for TSO users). The operator can also change the PERFORM parameter (thus the performance group) of an ongoing job or TSO session by issuing the RESET job name, PERFORM=nnn command. (Refer to *Operator's Library: OS/VS2 Reference (JES2)* for detailed syntax information on these commands.

**Value Range:** Any two-character alphameric combination.

**Default Value:** IPS=00. The IEAIPS00 default member, supplied by IBM, can be modified by the installation. The use and contents of this default member are described under "Default IPS" in "Part 3: The System Resources Manager".

**Associated Parmlib Member:** IEAIPSxx

**IEASYSxx (continued)**

**Parameter:** LNK=  $\left\{ \begin{array}{l} \text{aa} \\ (\text{aa,bb} \dots) \end{array} \right\}$

**Meaning and Use:** This parameter specifies particular LNKLSTxx member(s) of parmlib. The two alphameric characters, represented by xx, are appended to LNKLST to form the name of the LNKLSTxx member(s).

The LNKLSTxx member(s) list data sets that are to be concatenated to SYS1.LINKLIB. (For information on the use, contents, and syntax of this member, see the description of member LNKLSTxx.)

**Value Range:** Any two alphameric characters.

**Default Value:** LNK=00, causing selection of LNKLST00.

**Associated Parmlib Member:** LNKLSTxx

**IEASYSxx (continued)**

**Parameter:** LOGCLS=x

**Meaning and Use:** This parameter specifies the JES output class for the log data sets. A log data set is queued to this class when its WTL limit has been reached. (The limit is specified by the LOGLMT initialization parameter.)

*Example:* LOGCLS=L

In this example, the current log data set is queued to output class L when the limit on the number of WTLs has been reached.

If the specified LOGCLS value is invalid, or an I/O error occurs while the IEASYSxx member is being read, Master Scheduler Initialization prompts the operator for a replacement LOGCLS value. If prompting is forbidden (i.e., the OPI operand was specified), the default value A is assigned.

(For the other log parameter, see LOGLMT.)

**Value Range:** A single alphabetic or numeric character: A–Z or 0–9.

**Default Value:** A, which represents output class A.

**Associated Parmlib Member:** None

## IEASYSxx (continued)

**Parameter:** LOGLMT=nnnnnn

**Meaning and Use:** This parameter specifies the maximum number of WTLs (i.e., messages) allowed for each log data set. The value is used by Log processing to determine when a log data set should be scheduled for sysout processing by JES. When the value is reached, Log processing issues a simulated WRITELOG command to allocate and open a new log data set, and to close and free the current log data set.

*Example:* LOGLMT=004852

In this example, when 4,852 WTLs have been issued to a log data set, the data set is scheduled for sysout processing on the output class specified by the LOGCLS parameter. Log processing then switches log data sets.

If the specified value is invalid or an I/O error occurs while the IEASYSxx member is being read, Master Scheduler Initialization prompts the operator for a replacement LOGLMT value. If prompting is forbidden (i.e., the OPI operand was specified), the default value of 500 is assigned.

(For the other log parameter, see LOGCLS.)

**Value Range:** 000000–999999

**Default Value:** 500

**Associated Parmlib Member:** None

**IEASYSxx (continued)**

**Parameter:** MAXUSER=nnnn

**Meaning and Use:** This parameter specifies the maximum number of concurrent jobs and started tasks that the installation wishes to allow. The number includes time sharing jobs, batch jobs, started system tasks, the Master Scheduler, and JES2. The number determines the maximum number of entries in the address space vector table. (The address space vector table is used to locate the various address space control blocks.)

The MAXUSER value sets the limit on the number of concurrent jobs, by limiting the number of concurrent address space control blocks.

MAXUSER must be at least one more than the sum of: the number of JES2 batch initiators; the number of subsystems; the maximum number of logons; the maximum number of started tasks, excluding initiators. In general, the installation should set the MAXUSER value as high as the largest expected number of concurrent jobs.

**Value Range:** 0–9999. The MAXUSER value determines the extent of the search of the address space vector table during job step timing. An installation could reduce this search time by reducing its MAXUSER value, if possible.

**Default Value:** 256

**Associated Parmlib Member:** None



### IEASYSxx (continued)

Parameter: MLPA=  $\left\{ \begin{array}{l} \text{aa} \\ (\text{aa,bb. . .}) \end{array} \right\}$

**Meaning and Use:** This parameter specifies one or more IEALPAXx parmlib members, which list modules to be added to the pageable LPA, as a temporary LPA extension. Each member lists the names of modules to be loaded from LINKLIB, SVCLIB, and LPALIB.

The installation can use this parameter to temporarily update an existing LPA at a Quick Start (i.e., without creating a new LPA through the CLPA parameter). The added modules are temporary in that they remain as an LPA extension only for the life of the current IPL. The temporary modules may not be *automatically* reinstated by a Quick Start. That is, the MLPA parameter must be specified again in the next IPL to reinstate the LPA extension.

If the installation wants to retain the temporary modules as a permanent part of the LPA, it should use the IEBCOPY utility or the Linkage Editor to place the modules on SYS1.LPALIB, and specify the CLPA parameter at a future IPL to load LPALIB into the LPA.

(For additional information on the MLPA option, see IEALPAXx. For information on the other temporary LPA option, see the FIX parameter and the IEAFIXxx member.)

**Value Range:** Any two alphanumeric characters, repeated if desired.

**Default Value:** None. If MLPA is not specified, no LPA extension is created.

**Associated Parmlib Member:** IEALPAXx

**IEASYSxx (continued)**

**Parameter: NUCMAP**

**Meaning and Use:** This parameter specifies that NIP's DSS\* resource initialization module (RIM) should create a new map of the nucleus in the SYS1.DSSVM data set, and overlay the old nucleus map, if one exists.

The SYS1.DSSVM data must exist. NIP's DSS RIM issues a LOCATE, MOUNT, and OPEN for the data set. If the data set can't be found, NIP goes into a wait state.

NUCMAP must be specified if *all* the following conditions exist:

- The nucleus has been modified, perhaps by a PTF, and
- A nucleus map already exists from a previous IPL, and
- The nucleus load module name has not changed.

The parameter need not be specified again at a future IPL, unless the nucleus is again modified without a change in load module name. NIP's DSS RIM *automatically* creates a new nucleus map (i.e., NUCMAP need *not* be specified) if:

- The name of the nucleus load module changes between IPLs,  
or
- No nucleus map exists because this is the first IPL after sysgen.

**Value Range:** Not applicable.

**Default Value:** None

**Associated Parmlib Member:** None

---

\*Dynamic Support System, an FE debugging tool.

**IEASYSxx (continued)**

**Parameter:** OPI=  $\left\{ \begin{array}{c} \text{YES} \\ \text{NO} \end{array} \right\}$

**Meaning and Use:** This parameter specifies whether the operator is to be allowed to override system parameters contained in IEASYSxx members of parmlib. The YES operand allows operator overrides. The NO operand causes overrides to be ignored. If, however, NIP detects an invalid parameter in an IEASYSxx member in which OPI=NO applies, NIP ignores the OPI specification and prompts the operator.

OPI may be specified only in an IEASYSxx member; it may not be specified by the operator.

OPI may be specified either for individual system parameters or for the entire set of parameters.

*Examples:*

IEASYSAA: MLPA=(00,01),SQA=(10,OPI=NO)

IEASYSBB: MLPA=(00,01),SQA=10,OPI=NO

For IEASYSAA, the operator can override MLPA values but not the SQA value. For IEASYSBB, however, the operator can override neither MLPA nor SQA values.

**Value Range:** Not applicable

**Default Value:** YES

**Associated Parmlib Member:** Not applicable

**IEASYSxx (continued)**

**Parameter:** OPT=xx

**Meaning and Use:** This parameter specifies a parmlib member that contains tuning parameters used by the resource algorithms of the System Resources Manager (SRM). The two alphameric characters, represented by xx, are appended to IEAOPT to form the name of the IEAOPTxx member.

If the parameter is not specified, internal default values are used but no message is issued. If an invalid parameter is detected in IEAOPTxx, a message is issued to the operator\*, and the SRM uses internal defaults for the parameters normally furnished in IEAOPTxx. (See description of member IEAOPTxx in “Part 3: The System Resources Manager” for the default values.) The operator can either accept use of the defaults, or re-IPL to specify a replacement OPT value.

If the SRM can't find the specified member, it prompts the operator to supply a new value for OPT.

**Value Range:** Any two alphameric characters.

**Default Value:** None. If OPT is not specified, the SRM uses internal defaults. (See IEAOPTxx for the internal default values.)

**Associated Parmlib Member:** IEAOPTxx

---

\*Message IEA874I is issued.

## IEASYSxx (continued)

Parameter: PAGE=  $\left\{ \begin{array}{l} \text{dsname} \\ (\text{dsname1,dsname2,} \dots [\text{,L}]) \\ (\text{,L}) \end{array} \right\}$

**Meaning and Use:** This parameter allows the installation to name page data sets as additions to or replacements for existing page data sets. The maximum number of page data sets is 64, which includes the page data set specified by the DUPLEX parameter. To replace page data sets, the parameter must be in IEASYSxx, and the operator must select this member by entering SYSP=xx. To *add* a parmlib data set list, it is necessary only that the operator specify the PAGE parameter at the console.

If the “L” keyword is specified – either in parmlib or from the operator’s console – ASM generates a list of all the page data sets that the initialization routines have opened. This list is then printed at the operator’s console.

ASM interprets the sequence of data set names specified as follows:

- The first named data set on the list is used as the PLPA page data set. This data set contains all of the link pack area pages.
- The second named data set in the list is used as the common page data set. This data set contains all of the common area pages that are not LPA pages.
- The third and all subsequently named data sets are used as local page data sets. These data sets contain all the system pages (including VIO pages) that are considered neither PLPA nor common data set pages.

When defining page data sets, you must ensure that the desired PLPA page data set is the first entry in the data set list, in both IEASYS00 and IEASYSxx. For IEASYS00, this means that the desired data set name must be in the first sysgen DATASET macro containing the PAGEDSN keyword.

During ASM initialization, there are no checks on the sizes of user-supplied data sets. For this reason, the user must define data set sizes prudently. Also, there must be at least three page data sets available for IPL: the PLPA page data set, the common page data set, and at least one local page data set.

The data set intended for PLPA should contain enough space for the entire PLPA. If the entire PLPA cannot fit on this data set, ASM puts the excess on the common page data set. Likewise, if the common page data set gets full, its excess goes to the PLPA page data set. In the interest of good performance, however, you should make the common page data set big enough to prevent its “spilling over” to the PLPA page data set (except in cases forced by error situations). For specific data set size and placement recommendations, see “Part 2.1: Auxiliary Storage Management Initialization”.

**IEASYSxx (continued)**

**How Page Data Sets Are Specified:** Page data sets are specified by a merging of information from three sources: 1) IEASYS00 or IEASYSxx; 2) operator-issued PAGE parameter; and 3) the temporary page activity reference table (TPARTBLE). The system merges this information as follows:

- *From the PAGE parameter in IEASYS00:* These data set names were specified in the PAGEDSN keyword of the DATASET macro during system generation.<sup>1</sup>
- *From the page parameters in IEASYSxx (an alternate system list):* If the operator selects this list by using the SYSP parameter, the PAGE parameter in IEASYSxx overrides the PAGE parameter in IEASYS00. The overriding parameter can either increase or decrease the number of data sets defined during system generation.
- *From the PAGE parameter specified by the operator in the current IPL:* The system merges this page specification with that in either IEASYS00 or IEASYSxx, but not both. The operator specification of page data sets lasts only for the life of the new IPL, unless the data sets are also used for PLPA and/or VIO data sets.
- *From the temporary page activity reference table (TPARTBLE):* The system uses information from the TPARTBLE on quick starts and warm starts (IPLs that do not specify the CLPA parameter). The TPARTBLE contains the names of the page data sets used during the previous IPL, at which time the initialization routines wrote the TPARTBLE to the first record of the PLPA page data set. On subsequent quick starts or warm starts the information in TPARTBLE is thereby accessible to ASM initialization.

*Note:* Two other conditions are prerequisite for certain warm start or quick start situations:

- 1) The local page data sets from the previous IPL must be mounted for *all* warm starts, to make VIO slots available. Otherwise, ASM forces a quick start instead.
- 2) The common page data set from the previous IPL must be mounted for *both* quick starts and warm starts if NIP's writing of the PLPA to the PLPA page data set during the previous cold start resulted in spilling some of the PLPA records into the common page data set.

In general, page data sets specified by any means must have been allocated, cataloged in the system's master catalog, and preformatted in VSAM format (with track overflow) before an IPL can start. You can format the data sets by using the DEFINE PAGESPACE processor of access method services (refer to *OS/VS2 Access Method Services* for information about the formatting process). If you specify the data sets during sysgen, however, you need not explicitly use the DEFINE processor because the DATASET macro uses DEFINE itself to format the data sets for you.

<sup>1</sup>For guidance on using the DATASET macro instruction to create page data sets, refer to *OS/VS2 System Programming Library: System Generation Reference*.

**IEASYSxx (continued)****Syntax Examples for the PAGE Parameter:**

Example 1: The following statements each specify one page data set.

```
PAGE=dsname
PAGE=(dsname)
```

Example 2: The following statement specifies three page data sets.

```
PAGE=(dsname1,dsname2,dsname3)
```

Dsname1 holds the LPA pages, dsname2 holds the common pages, and dsname3 holds the private area pages.

Example 3: The following statement specifies *n* page data sets.

```
PAGE=(dsname1,dsname2, . . . ,dsname3-n)
```

Dsname1 holds the LPA pages, dsname2 holds the common pages, and dsname3 through dsnamen all hold private area pages.

**Notes:**

- 1) If the operator specifies the PAGE parameter, ASM initialization adds (but does not replace) the data sets as specified. IEASYS00 or IEASYSxx are used as the first named data sets. To ensure that the operator-specified data sets are used for the PLPA and common page data sets, it is necessary to use an IEASYSxx member that does not specify page data sets.
- 2) It is unnecessary to specify either UNIT or VOLSER because all page data sets must be cataloged. ASM initialization therefore does not need externally specified volume serial numbers. The operator may either premount volumes or await a mount message.

**Minimum Paging Space:** ASM enforces minimum requirements for paging space. If the requirements are not satisfied, ASM is forced to terminate the IPL or, later, the system. Nevertheless, the use of minimum paging space is inadvisable because it can result in poor performance.

*Minimum* requirements are as follows:

- There must be at least a PLPA, a common, and a local page data set to prevent failure of IPL.
- If no duplex page data set is used (the duplex parameter), the PLPA and common page data sets must be able to hold the total combination of PLPA and common pages. Eight megabytes of space divided between these data sets is sufficient as long as no errors occur. If the PLPA and common page data sets spill back and forth because of insufficient space, severe performance degradation can result.

**IEASYSxx (continued)**

If a duplex page data set is used and is large enough to hold all of the PLPA and common pages (the only duplexed areas), the system will be able to continue even if the PLPA and common page data sets are full. Nevertheless, this situation is inadvisable because it defeats the purpose of duplexing and may severely degrade performance.

- Local page data sets are used to hold all private area and VIO pages. The amount of storage necessary varies with each system and can be calculated using the guidelines in “Part 2.1: Auxiliary Storage Management Initialization” in this publication.

**Page Space Shortage:** Two warning messages appear when the system resources manager (SRM) detects a shortage of page space, the first when 70% of the available local paging space has been allocated, and the second when 85% has been allocated. The SRM reacts to the situation by preventing the creation of new address spaces. That is, new “start initiator” commands (\$SInn), LOGONs, MOUNT commands, and START commands for system tasks that execute in their own address spaces are therefore ineffective. Upon receipt of these messages, it may be possible to add paging space to the system dynamically by using the PAGEADD operator command. (Refer to *Operator’s Library: VS2 System Commands* for information about using PAGEADD, and to related information about the PAGNUM parameter here in this publication.) For these situations, the installation should keep some preformatted, cataloged VSAM paging data sets available. The data sets can be formatted by using the DEFINE PAGESPACE processor of access method services (see *OS/VS2 Access Method Services*).

If these data sets are not readily available, it may be possible to modify key ASM constants as described in “Part 2.1: Auxiliary Storage Management Initialization”. When the page space usage has been decreased below 70% utilization, the SRM informs the operator that there is no longer a shortage.

**Value Range:** Not applicable.

**Default Value:** None.

**Associated Parmlib Member:** None.

**Parameter:** PAGNUM=(pp,s)

**Meaning and Use:** This parameter allows the installation to specify the total number of additional page and swap data sets that can be dynamically added to the system via the PAGEADD operator command. These numbers represent the total number of data sets that can be added during the life of the IPL, regardless of whether they are added one at a time or all at once.

ASM supports as many as 64 page data sets and 25 swap data sets. If the number of data sets currently defined to the system and the additional number specified by this command exceed these limits, ASM ensures that the maximum supported values are not exceeded by truncating the excess specification. One should use this parameter with caution because ASM must reserve SQA space for each page or swap data set that can be dynamically allocated.



**IEASYSxx (continued)**

**How Page Number Values Are Obtained:** The PAGNUM parameter is specified in one of two sources: 1) IEASYS00 or IEASYSxx, or 2) by the operator-issued PAGNUM parameter. An outline of these specifications is as follows:

- *The PAGNUM parameter in IEASYS00:* This page and swap number value was specified in the PAGNUM keyword of the CTRLPROG macro during system generation.
- *The PAGNUM parameter in IEASYSxx, an alternate system parameter list:* If the operator selects this list (by using the SYSP parameter), the PAGNUM parameter in IEASYSxx overrides the PAGNUM parameter in IEASYS00.
- *The PAGNUM parameter specified by the operator in the current IPL:* This PAGNUM specification overrides the specification in IEASYS00 or IEASYSxx. The operator specification lasts only for the life of the IPL.

**Syntax Example for the PAGNUM parameter:**

PAGNUM=(pp,s)

This specification causes ASM to allow for the addition of *pp* page data sets and *s* swap data sets.

**Value Range:** Valid *pp* values are 0-99 so long as the value specified plus the number of page data sets specified does not exceed 64. If the combination exceeds 64, the value used is that which makes a sum of 64 rather than the specified value. Valid *s* values are 0-9 so long as the value specified plus the specified number of swap data sets does not exceed 25. If the combination exceeds 25, the value used is that which makes a sum of 25.

**Default Value:** 1,1.

**Associated Parmlib Member:** None.

**IEASYSxx (continued)**

**Parameter: PURGE**

**Meaning and Use:** This parameter specifies that all Mass Storage System (MSS) volumes currently mounted for this system are to be demounted. PURGE is required when different system configurations have been generated with varying numbers of MSS volumes. If the MSS volumes mounted for this system do not reflect the exact configuration shown in this system's UCBs, some volumes may be inaccessible to the system. Using PURGE to demount all existing volumes forces NIP at I/O initialization to mount the correct volumes for the current system configuration.

**Value Range:** Not applicable.

**Default Value:** None. (If not specified, no MSS volumes are demounted.)

**Associated Parmlib Member:** None.

## IEASYSxx (continued)

**Parameter:** REAL=nnnn

**Meaning and Use:** This parameter specifies the maximum amount of real storage, in 1K blocks, that can be allocated concurrently for ADDRSPC=REAL jobs. The value is rounded to a multiple of 4K bytes.

The REAL parameter may also be specified in the CTRLPROG macro at sysgen.

*Syntax Example:* REAL=150       $150 \div 4 = 37\text{-}1/2$  pages. Rounding to the next page boundary yields 38 pages, or a value of 152K. This statement allows up to 152K (152 x 1024) bytes to be allocated to an ADDRSPC=REAL job.

### Notes:

1. If possible, avoid a large value for the REAL parameter, since a large value degrades system performance, even when no REAL regions are allocated.
2. If REAL is specified as zero, no ADDRSPC=REAL job will be allowed to run. Furthermore, OLTEP will not run, since it requires a REAL region of 76K. Thus, for all practical purposes, 76 is the minimum REAL value.

(For storage layouts and general information on real storage usage, refer to *VS2 Release 2 Storage Estimates*.)

**Value Range:** 0-9999. The operand can be from one to four digits. (See Note 2 above.)

**Default Value:** 76. (This means a default value of 76K.)

**Associated Parmlib Member:** None

## IEASYSxx (continued)

**Parameter:** RSU=xx

**Meaning and Use:** This parameter specifies the number of processor storage units that are required for storage reconfiguration in a multi-processing system. The frames in these storage units are not to be used for long-term storage and will be designated the non-preferred area.

During IPL, the operating system assigns the nucleus and V=R areas to the low end of storage and assigns the current SQA to the high end of storage. The system then defines the reconfigurable storage units starting with the first available (online) high end storage unit after the SQA. Offline storage units are skipped during the calculation; when they are varied online after IPL they will automatically be available for storage reconfiguration.

After the system has defined the reconfigurable storage units, it defines the remainder of the processor storage units as the preferred area for long-term frames.

*Note:* While jobs are processing, short-term pages are assigned to any available processor storage frames in either non-preferred or preferred areas. Long-term pages are assigned only to storage frames in the preferred area. However, when the condition occurs that a long-term page requires storage space but the preferred area is full, the system performs one of the following:

- *Immediate steal* – If a short-term page that has not been changed is using a frame in preferred storage, the system removes the page and assigns the new long-term page to the vacated space.
- *Dynamic expansion* – If all of the frames in the preferred area are being used for pages that are not eligible for “immediate steal”, the system looks for a frame in the non-preferred area. When it finds a frame, it converts the entire processor storage unit from non-preferred to preferred.

Dynamic expansion lowers the number of reconfigurable storage units designated by the RSU parameter. Therefore, message IEA9881 is issued at the system console to inform the system operator. The operator can then issue the Display Matrix (D M) command to determine which units are still available for reconfiguration.

**Syntax Example:** RSU=4. Four storage units will be available for storage reconfiguration.

**Value Range:** 0-99. The minimum number of reconfigurable storage units required to meet installation subsystem requirements should be specified. If a larger value is specified than can be satisfied, the maximum possible is defined without an error indication. (See the *OS/VS2 Release 3 Guide*, GC28-0700, for information on determining subsystem requirements.)

**Default Value:** 0. If the RSU parameter is omitted or specified as 0, all processor storage units are available for preferred storage.

**Associated Parmlib Member:** None.

## **IEASYSxx (continued)**

**Parameter:** SMF=xx

**Meaning and Use:** This parameter specifies the parmlib member, SMFPRMxx, from which SMF will obtain its options. The two alphanumeric characters, represented by xx, are appended to SMFPRM to name the member. NIP saves the name until SMF is initialized.

(For detailed information on SMF parameters, see member SMFPRMxx. For suggestions on how to use SMF to supplement the Measurement Facility MF/1, see the topic “How SMF Can Supplement MF/1” in the chapter titled “System Performance Factors”.)

**Value Range:** Any two alphanumeric characters.

**Default Value:** 00 (This specifies SMFPRM00, the IBM-supplied default parmlib member.)

**Associated Parmlib Member:** SMFPRMxx

## IEASYSxx (continued)

**Parameter:** SQA=nnn

**Meaning and Use:** This parameter specifies to the Virtual Storage Manager the maximum size of the virtual system queue area. The virtual system queue area is fixed in real storage as it is used. The three digits, represented by nnn, indicate the number of 64K blocks (segments) to be added to the system's minimum virtual SQA of 3 segments, or 192K bytes.

SQA may also be specified by means of the SQA keyword of the CTRLPROG macro at sysgen.

*Note:* During a Quick Start (i.e., when the CLPA parameter is not specified), NIP determines if the currently specified SQA value is equal to the previously specified (cold start) value. If it is not, NIP issues an informational message and uses the cold-started SQA value. (See the CLPA parameter for a comparison between *cold start* and *quick start*.)

*Syntax Example:* SQA=4 This value indicates to the Virtual Storage Manager that its maximum virtual allocation for SQA should be: 192K + (4 x 64K) = 448K.

**SQA Space Shortage:** If the Virtual Storage Manager runs completely out of SQA, it tries to allocate space from the common service area (CSA). If it can't obtain space from the CSA because the reserved CSA area is too small, it places the system in a disabled wait state. To avoid this possibility, the installation should specify extra space in the CSA parameter, beyond that obtained from the storage formula. (For calculations, storage layouts, and general storage information, see *OS/VS2 System Programming Library: Storage Estimates*.)

The Virtual Storage Manager keeps track of the remaining virtual SQA, and informs the System Resources Manager, via a SYSEVENT macro, when the total of available virtual SQA plus available virtual CSA has reached two threshold values. The two values are currently six pages, or 24K, and two pages, or 8K. The System Resources Manager reacts to the situation by issuing an "SQA shortage" message at each of the two thresholds. At the upper threshold (24K) it also inhibits the creation of new address spaces by disallowing Start Initiator commands (\$S Innn), LOGON commands, MOUNT commands, and START commands for system tasks that require their own address spaces, such as START TCAM or START MF1.

**Value Range:** 0–999 (one to three digits)

### **IEASYSxx (continued)**

*Note:* During NIP processing, if a large number of paging data sets have been specified via the PAGE= parameter, or if many 2305 paging data sets are active, the minimum three segments of SQA may be exhausted before NIP can process the SQA= parameter. If this situation occurs, you can change the content of the half-word at NVTNVSQA in module IEAVNIP0 by means of the AMASPZAP service aid (see *OS/VS2 System Programming Library: Service Aids*, GC28-0674) to reflect an increase in the initial SQA allocation. For example, suppose that the content of NVTNVSQA is X'0003'. To add one additional segment of SQA to the initial SQA size available to NIP processing, change the NVTNVSQA field content to X'0004'. (If you increase the content of NVTNVSQA, you should correspondingly decrease the size represented by the SQA= parameter.) To find the location of the NVTNVSQA field, consult listings of the source code or a representation of them on microfiche.

**Default Value:** 1 (This means 192K + 64K, or a default size of 256K.)

**Associated Parmlib Member:** None

**IEASYSxx (continued)**

**Parameter:** SWAP= { dsname  
(dsname1,dsname2 . . .) }

**Meaning and Use:** This parameter allows the installation to name swap data sets as replacements for existing swap data sets for a maximum of 25 swap data sets in all. This parameter follows the normal override rules for system parameters. In other words, any swap data set specified in a parmlib member IEASYSxx, or specified by the operator, completely overrides the previously specified data sets.

ASM swap data sets are optional. If none are supplied, the LSQA pages normally sent to a swap data set will be directed to a local page data set. Similarly, if swap data sets become full during system operation, the LSQA pages will be directed to a local page data set. Nevertheless, it is advisable for sufficient swap space to be always available to ASM to prevent possible performance degradation. Additional information and recommendations are in “Part 2.1: Auxiliary Storage Management Initialization”.

**How Swap Data Sets Are Specified:** Swap data sets are specified from one of two sources: 1) the IEASYS00 or IEASYSxx parmlib member, and 2) the operator-issued SWAP parameter.

- *The SWAP parameter in IEASYS00* – These data set names were specified in the SWAPDSN keyword of the DATASET macro during sysgen.
- *The SWAP parameter in IEASYSxx* (an alternate system parameter list) – If the operator selects this list (by issuing the SYSP parameter), the SWAP parameters in IEASYSxx override the SWAP parameter in IEASYS00. The overriding parameters can either increase or decrease the number of data sets specified during sysgen.
- *The SWAP parameter specified by the operator in the current IPL* – This SWAP parameter specification does not merge with those specified by IEASYSxx, as is the case with the PAGE parameter specification, but instead overrides those specifications. The operator specification last only for the life of the IPL.

Before an IPL, SWAP data sets specified through SWAP, or through the DATASET<sup>1</sup> macro at sysgen, must have been allocated, cataloged in the system master catalog, and preformatted in VSAM format, without track overflow. (Unlike page data sets, swap data sets use multi-tracking instead of track overflow.) You can use the DEFINE PAGESPACE processor of access method services to format the data sets; refer to *OS/VS2 Access Method Services* for information about how to use this processor. For information about using the DATASET macro to create swap data sets, refer to *OS/VS2 System Programming Library: System Generation Reference*.

<sup>1</sup>The SWAPDSN keyword of the DATASET macro causes the SWAP parameter to be placed in IEASYS00.



## IEASYSxx (continued)

**Syntax Examples for the SWAP Parameter**

Example 1: SWAP=dsname  
—or—  
SWAP=(dsname)

Either of these statements specifies one swap data set.

Example 2: SWAP=(dsname1,dsname2)

This statement specifies two swap data sets.

Example 3: SWAP=(dsname1,dsname2, . . . ,dsnamen)

This statement specifies *n* swap data sets.

Note that neither UNIT nor VOLSER is specified. Since all swap data sets must be cataloged, ASM initialization does not need externally specified volume serial numbers. The operator may either premount volumes or await a mount message.

**Minimum Swap Space:** ASM neither checks for a swap data set nor requires a minimum size if one is specified. If no swap data set is specified, or if swap data set space is exhausted, ASM forces further swap LSQA pages to a local page data set. For this reason, one must consider local page data set sizes.

**Swap Space Shortage:** If swap data set space becomes unavailable during system processing (as evidenced by message ILR009I, “SWAP DATA SET BAD”), it may be possible to add swap data sets with the PAGEADD operator command. Information about PAGEADD is in *Operator’s Library: OS/VS2 MVS System Commands*. Related information is also available here in the *Initialization and Tuning Guide* under the “PAGNUM Parameter” topic and in “Part 2.1: Auxiliary Storage Management Initialization”.

For situations where swap data sets may be added, the installation may keep preformatted, cataloged VSAM swap data sets available.

**Value Range:** Not applicable.

**Default Value:** None.

**Associated Parmlib Member:** None.

### IEASYSxx (continued)

**Parameter:** SYSP =  $\left\{ \begin{array}{l} \text{aa} \\ (\text{aa}, \text{bb} \dots) \end{array} \right\}$

*Note:* This parameter may be specified only by the *operator*. It cannot validly be specified in a system parameter list. It is included here for the sake of completeness.

#### Meaning and Use:

The SYSP parameter specifies that one or more alternate system parameter lists (e.g. IEASYS01, IEASYS02, etc.) are to be read by NIP in addition to the default list IEASYS00. The two alphameric characters, represented by aa, are appended to IEASYS to name the alternate list(s). Any number of lists are valid. The specification cannot be prohibited by the OPI parameter.

A system parameter value specified in an operator-selected alternate list overrides the value for the same parameter, if it is specified in IEASYS00. (IEASYS00 is always read.) NIP accepts parameters in alternate lists specified in SYSP in priority order from right to left. This means that a parameter defined in a list specified later in SYSP overrides the same parameter defined in a list specified earlier in SYSP.

#### Example:

The operator responds to SPECIFY SYSTEM PARAMETERS by entering:

```
R00, 'SYSP=(01, 02)'
```

Assume that the two specified members contain the following parameters:

```
IEASYS01: MLPA=(00, 01), BLDL=00, SQA=8
```

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

NIP would accept MLPA=02, BLDL=00, and SQA=10, in addition to other parameters contained in IEASYS00.

**Value Range:** Any two alphameric characters.

**Default Value:** 00 (This specifies IEASYS00.)

**Associated Parmlib Member:** IEASYSxx

### IEASYSxx (continued)

**Parameter:** VAL=  $\left\{ \begin{array}{l} \text{aa} \\ (\text{aa}, \text{bb}, \dots) \end{array} \right\}$

**Meaning and Use:** This parameter specifies the VATLSTxx member or members of parmlib. They are new members in MVS, although similar in function to PRESRES in MVT and VS2 Release 1. Volume Attribute Processing reads this member or members during Initialization to obtain *mount* and *use* attributes for direct access volumes. The *mount* and *use* attributes are set in the UCBs whose volume serial numbers are listed in the VATLSTxx member(s). If multiple members are specified, the lists are read in the order in which they appear in the VAL parameter. If a particular volume serial number appears on more than one entry, the volume attributes specified in the last entry for that volume serial will be accepted. If the volume serial does not duplicate another entry, but is for an MSS volume (see the 'device type' parameter in VATLSTxx description), then the last entry with that particular MSS unit address will be used to set the volume attributes for the volume.

(For additional information, refer to the topic that describes the VATLSTxx member.)

**Value Range:** Any two alphameric characters.

**Default Value:** 00 (This means that VATLST00, if it exists, will be read.)

**Associated Parmlib Member:** VATLSTxx

**IEASYSxx (continued)**

**Parameter:** VRREGN=nnnn

**Meaning and Use:** This parameter specifies the default real-storage region size for an ADDRSPC=REAL job step that does not have a REGION parameter in its JCL. The numerical value of the operand (nnnn) indicates the real-storage region size in 1K-byte blocks. (VRREGN may also be specified by the CTRLPROG macro at sysgen.)

**Note:** The following VRREGN values will prevent an ADDRSPC=REAL job step from running if it omits a REGION parameter:

- VRREGN value that is greater than the value of the REAL parameter, or
- VRREGN value of zero.

(For further information on the reserved area of real storage, refer to *OS/VS2 System Programming Library: Storage Estimates*.)

**Value Range:** 0–9999

**Default Value:** 64 (This means a default REGION size of 64K.)

**Associated Parmlib Member:** None

## IEASYSxx (continued)

**Parameter:** WTOBFRS=nnnn

**Meaning and Use:** This parameter specifies the number of buffers that the Write-to-Operator (WTO) routines will use for operator system (not JES2) messages. The Write-to-Operator buffers are assigned to fixed storage. The parameter was *formerly* a keyword of the SCHEDULR sysgen macro.

**Note:** If an insufficient number of buffers are specified, throughput may degrade because some messages that need a reply will have to wait for buffers to become available. Choose a value significantly higher than 20 (the default) to avoid an early wait state soon after IPL.

**Value Range:** 20–9999

**Default Value:** 20 If the value is less than 20, the specification is ignored, a message is issued, and a value of 20 buffers is assigned.

**Associated Parmlib Member:** None

**IEASYSxx (continued)**

**Parameter:** WTORPLY=nnn

**Meaning and Use:** This parameter specifies the number of operator reply elements to be used by the Write-to-Operator-with-Reply (WTOR) routines. The reply elements are used to queue outstanding replies not yet received from the operator. The parameter was formerly the REPLY parameter of the SCHEDULR sysgen macro.

**Value Range:** 5 – 100

**Default Value:** 5

**Associated Parmlib Member:** None

## Member Name: IKJPRM00

### Status

Changed from MVT and VS2 Release 1. The member formerly contained time sharing control parameters, driver parameters, and terminal I/O coordinator (TIOC) parameters. It now contains only TIOC parameters. The time-sharing control parameters and driver parameters need not necessarily be removed, since the TIOC will ignore them. However, your system bookkeeping will be simpler, if you delete obsolete parameters.

### Use of the Member

IKJPRM00 is an optional member that contains installation-defined TIOC parameters used mainly to control time sharing buffers. The TS and DRIVER parameters, specifiable under MVT and VS2 Release 1, are no longer valid. They are now fixed internal parameters of the System Resources Manager. IKJPRM00 is used only during TIOC initialization and does not participate in System Initialization.

If the installation uses time sharing, the system programmer may optionally construct this member by use of the IEBUPDTE utility. (See example of IEBUPDTE statements in chapter introduction.) The default values, listed under "Internal Parameters" later in this section, are internal constants of the TIOC program. You may override a default value by placing the same parameter into the member.

IKJPRM00, or an alternate member name, may be specified by the operator as an optional parameter of the MODIFY tcamproc command. The command starts time sharing under MVS. The command syntax consists of:

```
MODIFY tcamproc,TS=START [,membername]
```

Membername can be either defaulted to IKJPRM00, or specified as the name of an installation-defined alternate. If the operator omits the member name, the system looks for member IKJPRM00 when time sharing is started. (For additional information on the use of the MODIFY tcamproc command, see *Operator's Library: OS/VS2 Reference (JES2)* and *Operator's Library: OS/VS2 TCAM.*)

TIOC Initialization tries to obtain parameters by reading the specified parmlib member. Special processing occurs if errors are encountered. If parmlib can't be allocated or opened, an information message is issued, and default parameters are used.\* If the specified member can't be found in parmlib, another message is issued and TIOC Initialization terminates. In this case, the operator should reenter the MODIFY tcamproc command, either specifying the correct member name or omitting the member name. If the name is omitted, TIOC Initialization tries to read IKJPRM00. If it can't locate IKJPRM00, or encounters an I/O error in reading the explicit or default member, it uses the default parameters.\* If TIOC Initialization encounters an invalid parameter, which is not correctly specified in a later entry, it uses the default value.\* Unsupported parameters, if retained from a previous version of IKJPRM00, are ignored.

---

\*See "IBM Supplied Defaults" later in this section.

## IKJPRM00 (continued)

Parameter in IEASYSxx: None  
(or specified by the operator)

### Syntax Rules

The following rules apply to the creation of IKJPRM00 by means of the IEBUPDTE utility:

- Each record must start with the word TIOC, followed by a blank.
- For each record, columns one through 71 are valid for data. Columns 72 through 80 are ignored.
- A parameter must be complete in a record. It may not cross record boundaries. The parameter, however, may be repeated.
- When a parameter is specified more than once in the member, the last occurrence is accepted.
- You may use either a blank or a comma as a separator between adjacent keywords.
- Invalid or misspelled parameters are ignored. Defaults are substituted, and an informational message is issued to the operator.

### IBM-Supplied Defaults

The default values are internal constants in the TIOC program. Those that were in effect for MVT and VS2 Release 1 have been changed, and are now the following:

BUFSIZE=64, BUFFERS=6\*USERMAX, USERMAX=number of time-sharing terminals<sup>1</sup> + 10%, OWAITHI=20, OWAITLO=4, INLOCKHI=4, INLOCKLO=1, RESVBUF=BUFFERS/10, RECONLIM=0

### Internal Parameters

Two parameters valid in MVT and VS2 Release 1 have been removed, and two new parameters have been added to IKJPRM00. The two removed parameters are USERCHG and SLACK. If either parameter is specified, the TIOC will ignore the parameter. The two new parameters are USERMAX and RECONLIM. They are described in the following tabulation.

---

<sup>1</sup>The number of time sharing terminals is the number of TCAM terminals defined as usable for time-sharing. (See *OS/VS2 TCAM Programmer's Guide* for information on defining terminals for time-sharing.)



**IKJPRM00 (continued)**

Parameter	Meaning and Use	Value Range	Default Value
BUFSIZE	Specifies the storage size of a TIOC buffer.	20 - 252	64
BUFFERS	Specifies the number of buffers in the TIOC buffer pool. (See note below.)	4 - 32,767	six times the USERMAX value
INLOCKHI	<p>Specifies the number of TIOC buffers to be allocated to a terminal user for input before his keyboard is locked. This is not an exact lock but works on an input line basis. If the number of buffers used to input one or more lines exceeds the INLOCKHI value, the keyboard remains locked until all of these conditions are satisfied: the user is swapped in, part or all of the input is removed (the TGET is satisfied), and the number of allocated buffers is reduced to or below the INLOCKLO value.</p> <p>INLOCKHI must be large enough to permit the largest possible legitimate input message sent from any terminal in your system to be received by TIOC. Any input message larger than BUFSIZE times INLOCKHI will be canceled and will cause an error message at the terminal.</p> <p><i>Note:</i> If the terminal is defined as "NO BREAK" (in the TERMINAL command or TCAM message control program), the INLOCKHI value is inconsequential, since the keyboard in this case is unlocked only when a demand for input occurs (i.e., when a TGET is issued).</p>	1 - 32,767	4
INLOCKLO	<p>Specifies a low threshold of allocated input buffers. When the number of allocated input buffers is reduced to or below this number, the user's keyboard is unlocked.</p> <p><i>Note:</i> If the terminal is defined as "NO BREAK", the INLOCKLO value is inconsequential. (See <i>Note</i> for INLOCKHI.)</p>	less than INLOCKHI and BUFFERS	1
OWAITHI	Specifies the maximum number of output buffers that can be allocated to a terminal. When that number is reached, the user's address space is placed in output wait and is swapped out of real storage.	1 - 32,767	20

*Note:* If your installation uses the 3270 terminal, specify enough buffers to completely fill the screen. You may compute this number of buffers from the formula:

$$\text{Buffers} = (\text{message length} + 6) \div (\text{BUFFERSIZE} - 6)$$

If there are not enough buffers for a "full screen write," the address space will be put into output wait and swapped out until buffers become available.

**IKJPRM00 (continued)**

<b>Parameter</b>	<b>Meaning and Use</b>	<b>Value Range</b>	<b>Default Value</b>
<b>OWAITLO</b>	Specifies a low threshold value for the number of allocated output buffers. When the number of output buffers reaches this value, the System Resource Manager is notified that the terminal user's job can be swapped into storage and allowed to execute.	less than OWAITHI and BUFFERS	4
<b>RECONLIM</b>	Specifies the time limit in minutes within which a user may reconnect after his TP line has been disconnected.	0 - 32,767	0
<b>RESVBUF</b>	Specifies the minimum number of free buffers that are available. Its purpose is to maintain a reserve of free buffers that can handle output without "bottlenecking" the system. If the number of free buffers falls below this value, all terminals are locked for input, regardless of INLOCKHI value. The terminals will be unlocked when the number of free buffers becomes equal to RESVBUF.	1 - value of BUFFERS	10% of the number of buffers specified in BUFFERS parameter.
<b>USERMAX</b>	Specifies the maximum number of time-sharing users that may be logged on.	1 - 32,767	Total number of terminals that support time sharing +10%. ( <i>Note:</i> The number of terminals that support time sharing is specified in the TCAM Message Control Program.) (See <i>OS/VS2 TCAM Programmer's Guide.</i> )

**Member Name: IRBMF1xx**

### **Status**

New for MVS. This member did not exist in MVT or VS2 Release 1.

### **Use of the Member**

IRBMF1xx is used to control the functioning of the System Activity Measurement Facility (MF/1). See the chapter entitled "How to Use the System Activity Measurement Facility (MF/1)".

MF/1 control options can be specified by the operator in the START command 'parmvalue' field, or be specified in the PARM field of the EXEC statement in the MF/1 cataloged procedure, or in the IRBMF1xx library member. Combinations of these sources are possible. Parameters specified in the START command override any conflicting specifications in either the EXEC statement or in the library member. Parameters which have not been specified in the START command can take values from parameters in the EXEC statement or library member. Parameters which have been specified in neither the START command nor the EXEC statement can take values from specifications in the library member. Finally, parameters not specified in any of these three sources use program default values.

If mutually exclusive parameters (e.g., CPU/NOCPU) are specified within one of the three option sources, the last specification will override and the operator will receive the message 'IRB301I INVALID OR MUTUALLY EXCLUSIVE OPTIONS APPEAR IN MF/1 [OPERATOR] [PARM] [LIBRARY] INPUT'. If, however, conflicting specifications occur within *different* option sources, the higher priority source will override, and no message will be issued.

The foregoing message will also be issued if a parameter contains an invalid value (for example, if 'CYCLE(10)' is specified). If the source in error is the EXEC statement or the library member, the values of these parameters will be listed. The listing includes the last specification in the case of a conflict, or a default in the case of an invalid value. If the option source in error is the START command, the operator is given the opportunity to either re-enter the START command options, or to ignore the START command as an option source.

If an option source contains a syntax error, the operator receives the message 'IRB300I MF/1 SYNTAX ERROR IN OR FOLLOWING TEXT BEGINNING 'XXXXXXXXXX' IN [OPERATOR] [PARM] [LIBRARY] INPUT'. As with invalid values, the options are listed. In the case of options from the START command, the operator is able to make immediate changes or ignore the option source.

If after all controlling options have been gathered, a logical inconsistency is detected in differing options (for example, the specified STOP value is less than the INTERVAL value), the following message is issued: 'IRB301I INVALID VALUES OR MUTUALLY EXCLUSIVE OPTIONS APPEAR IN MF/1 INPUT'.

## IRBMF1xx (continued)

If any of the above messages are issued during MF/1 option determination, and the operator has not yet been given the opportunity to change the controlling options, (or if the 'OPTIONS' parameter was specified), all options are listed. After the options are listed, the message 'IRB306D REPLY WITH MF/1 OPTIONS OR GO' is printed, and the operator has the opportunity to change any options. Exception: He cannot now give another value for the MEMBER parameter, since no further library member scan will be made. If further errors occur in the operator's input, a sequence similar to the above will follow, except that 'REPLY' rather than 'OPERATOR', 'PARM', or 'LIBRARY' will be listed as the source of the error.

If the selected options consist of 'NOCPU', 'NOPAGING', 'NOCHAN', 'NOWKLD' and 'NODEVICE' (indicating that no MF/1 measurements are to be collected), the message 'IRB202I NO MF/1 MEASUREMENTS SELECTED' will be issued, and MF/1 execution will be terminated.

**Parameter in IEASYSxx:** None  
(or issued by the operator)

## Syntax Rules

- MF/1 control options are taken from the library data set as 80-byte card images. Valid data may be placed in columns 1 through 72. Columns 73 through 80 will be ignored.
- MF/1 options may appear in any order, separated by commas, blanks, or comments (in the form /\* text \*/). Additional blanks are ignored, and they may appear anywhere except within option keywords. Comments may appear anywhere that blanks appear.

## IBM-Supplied Defaults

The following values are in the IBM-supplied PARMLIB member IRBMF100. They are also the internal program default values. (See "Internal Parameters" for the meaning of these parameters):

CPU  
CHAN  
DEVICE (CHRDR, COMM, DASD, GRAPH, TAPE, UNITR)  
PAGING  
WKLD (PERIOD)  
CYCLE (250)  
INTERVAL (15M)  
MEMBER (00)

(This is a program default only. This parameter does not occur in IRBMF100, since a member name cannot be specified within a library member.)

OPTIONS  
NORECORD  
REPORT (DEFER)  
STOP (15M)  
SYSOUT (A)

IRBMF1xx (continued)

Internal Parameters

Parameter	Meaning and Use
$\left\{ \begin{array}{l} \underline{\text{CPU}} \\ \underline{\text{NOCPU}} \end{array} \right\}$	Specifies whether system CPU activity is to be monitored by MF/1.
$\left\{ \begin{array}{l} \underline{\text{CHAN}} \\ \underline{\text{NOCHAN}} \end{array} \right\}$	Specifies whether system channel activity is to be monitored by MF/1.
$\left\{ \begin{array}{l} \underline{\text{DEVICE}} \text{ (device list)} \\ \underline{\text{NODEVICE}} \end{array} \right\}$	Specifies whether system device activity is to be monitored by MF/1. If device activity is to be monitored (DEVICE specified), a device list indicates the classes of devices that will be monitored. When the DEVICE parameter is specified, a device list must be included.

Device List Members:

$\left\{ \begin{array}{l} \underline{\text{CHRDR}} \\ \underline{\text{NOCHRDR}} \end{array} \right\}$	Character reader devices.
$\left\{ \begin{array}{l} \underline{\text{COMM}} \\ \underline{\text{NOCOMM}} \end{array} \right\}$	Communications equipment.
$\left\{ \begin{array}{l} \underline{\text{DASD}} \\ \underline{\text{NODASD}} \end{array} \right\}$	Direct access storage devices.
$\left\{ \begin{array}{l} \underline{\text{GRAPH}} \\ \underline{\text{NOGRAPH}} \end{array} \right\}$	Graphics devices.
$\left\{ \begin{array}{l} \underline{\text{TAPE}} \\ \underline{\text{NOTAPE}} \end{array} \right\}$	Magnetic tape devices.
$\left\{ \begin{array}{l} \underline{\text{UNITR}} \\ \underline{\text{NOUNITR}} \end{array} \right\}$	Unit record devices.

$\left\{ \begin{array}{l} \underline{\text{PAGING}} \\ \underline{\text{NOPAGING}} \end{array} \right\}$	Specifies whether system paging activity is to be monitored by MF/1.
$\left\{ \begin{array}{l} \underline{\text{WKLD}} \left\{ \begin{array}{l} \underline{\text{(PERIOD)}} \\ \underline{\text{(GROUP)}} \\ \underline{\text{(SYSTEM)}} \end{array} \right\} \\ \underline{\text{NOWKLD}} \end{array} \right\}$	Specifies whether system workload activity is to be monitored by MF/1. If workload activity is to be monitored (WKLD specified), the level of detail of the report to be produced must be specified. PERIOD requests the most detailed level by performance group period; GROUP requests reporting by performance group; SYSTEM requests a system summary report.
CYCLE (value)	Specifies the frequency at which sampling observations are to be made of channel and device data. The range is from 50 to 999 milliseconds. The default is 250 milliseconds.
INTERVAL $\left\{ \begin{array}{l} \text{(Value)} \\ \text{(Value M)} \end{array} \right\}$	Specifies the interval at which all data will be gathered for report formatting and/or SMF record writing. The range is from 1 to 60 minutes. The default is 15M.

## IRBMF1xx (continued)

Parameter	Meaning and Use
MEMBER (nn)	The value specified by this parameter is appended to IRBMF1 to form the name of the partitioned data set member that contains the MF/1 options. This parameter may be specified in the PARM field of the START command or EXEC statement, but should <i>not</i> be specified within a partitioned data set member. The default is 00, indicating member IRBMF100 in SYS1.PARMLIB.
$\left\{ \begin{array}{l} \text{OPTIONS or OPTN} \\ \text{NOOPTIONS or} \\ \text{NOOPTN} \end{array} \right\}$	Specifies whether or not a list of the options to be used will be printed at the operator's console at MF/1 initialization. If the list is printed (OPTIONS or OPTN is specified), the operator will be able to respond with any desired changes to the option list.  <i>Note:</i> To avoid unnecessary console output and a delay in activating MF/1, specify NOOPTIONS. If any syntax errors are detected by MF/1, OPTIONS will be forced.
$\left\{ \begin{array}{l} \text{RECORD} \\ \text{NORECORD} \end{array} \right\}$	Specifies whether monitored data is to be written to the SMF data set. If SMF records are to be written, SMF data recording must have been specified at IPL as MAN=ALL.
$\left\{ \begin{array}{l} \text{REPORT} \left\{ \begin{array}{l} \text{(REALTIME)} \\ \text{(DEFER)} \end{array} \right\} \\ \text{NOREPORT} \end{array} \right\}$	Specifies whether or not printed reports of the monitored data are to be produced. If the reports are to be produced (REPORT is specified), the REALTIME/DEFER options specify whether the reports are to be printed when formatted at the conclusion of a data gathering interval (REALTIME is specified), or printed after MF/1 processing ends (DEFER is specified).
$\left\{ \begin{array}{l} \text{STOP} \left\{ \begin{array}{l} \text{(value)} \\ \text{(value H)} \\ \text{(value M)} \end{array} \right\} \\ \text{NOSTOP} \end{array} \right\}$	Specifies the desired time duration for MF/1 activity in minutes or hours. The range is from one minute to one week (168 hours or 10,080 minutes). The units default is M (minutes). The default value is 15M. The operator STOP command can terminate execution at any time, regardless of the value for this parameter.
SYSOUT (class)	Specifies the SYSOUT class to which formatted reports are directed. Class A is the default.

**Member Name:** LNKLSTxx (Link Library List)

### Status

A single member, LNKLST00, existed in MVT and VS2 Release 1. The current multi-member option was new in VS2 Release 2.

### Use of the Member

LNKLSTxx contains the names of program libraries on multiple volumes that are to be concatenated to SYS1.LINKLIB. The default member LNKLST00, built at sysgen, contains only the name SYS1.LINKLIB. The installation may add other library names to LNKLST00 through use of the IEBUPDTE utility.

You can set up any number of LNKLSTxx members, although no more than 15 libraries may be concatenated to LINKLIB in the life of a single IPL. Each library can have up to 16 extents. NIP opens and concatenates each library in the order in which library names are listed, starting with the first-specified LNKLSTxx member. If the total number of libraries (excluding LINKLIB) exceeds 15, only the first 15 are used and a warning message is issued.

*Notes:* Any library listed in LNKLST00 or LNKLSTxx is automatically authorized, since LINKLIB is authorized. LNKLST-named libraries must be cataloged in the system master catalog. (OS CVOLs are not searched for LNKLST-named libraries at IPL.)

**Parameter in IEASYSxx:** LNK= { aa  
(or specified by the operator) (aa,bb,01,. . .) }

The two alphameric characters, represented by aa, are appended to LNKLST to identify one or more LNKLSTxx members of parmlib. If the parameter is not specified either in IEASYSxx or by the operator, the default member LNKLST00 is used.

### Syntax Rules

The following rules apply to the creation of LNKLSTxx by means of the IEBUPDTE utility:

- Place on each record a string of data set names separated by commas.
- Indicate continuation by placing a comma followed by at least one blank after the last name on a record.
- Make sure that the total number of data sets, excluding LINKLIB, contained in all the specified LNKLST members doesn't exceed 15. Otherwise, a message is issued, and only the first 15 named data sets are concatenated.
- If you place the name SYS1.LINKLIB on any record in any LNKLST member, the name will be ignored.

## LNKLSTxx (continued)

- Be careful not to specify the same library name more than once in a succession of LNKLSTxx members. The same library will be concatenated as many times as it appears in all specified LNKLST members. The result can be slowed performance, since the same library can be searched two or more times for a module that resides there.

### *Syntax Example*

```
IEASYSxx:      . . . ,LNK=(00,01,02,03)
LNKLST00:      DOG,FOX,EASY,KING,JOG,BOBBY
LNKLST01:      PETER,QUEEN,MIKE
LNKLST02:      GEORGE,BAKER,ABLE
LNKLST03:      DATA1,DATA2,DATA3
```

The result of the foregoing specification is that the following data sets are concatenated to SYS1.LINKLIB:

```
DOG,FOX,EASY,KING,JOG,BOBBY,PETER,QUEEN,MIKE,GEORGE,
BAKER,ABLE,DATA1,DATA2, and DATA3.
```

### **IBM-Supplied Defaults**

Default member LNKLST00 contains only the name SYS1.LINKLIB.

**Internal Parameters:** Not applicable



**Member Name: MVIKEY00**

### **Status**

A new member for VS2 Release 3 that is provided for the Mass Storage System (MSS). MVIKEY00 is automatically built at sysgen, but it can be altered at any time by means of the IEBUPDTE utility.

### **Use of the Member**

MVIKEY00 supplies three parameters to the Mass Storage System Communicator (MSSC) that define the names of the mass storage volume Inventory and volume control Journal data sets, where the space manager messages are to be written, and how often to check the Staging Drive Table for balancing the staging drive groups.

**Parameter in IEASYSxx:** None  
(or specified by the operator)

### **Syntax Rules**

The following rules apply when MVIKEY00 is updated with the IEBUPDTE utility:

- Use columns one through 71. Do not use columns 72-80, since these columns are ignored.
- Avoid embedded blanks.
- Separate consecutive parameters by a comma.
- Do not divide a parameter between consecutive records.
- Indicate continuation by a comma followed by one or more blanks after the last entry on a record.

### *Syntax Examples*

```
MSVCCAT=MSVICAT,MSSCSAMP=02,MSFMSG=02  
MSVCCAT=USERCAT1,MSFMSG=(05,JOURNAL)
```

### **IBM-Supplied Defaults**

The following defaults are automatically placed in MVIKEY00 at sysgen:

```
MSVCCAT=MSVICAT,MSSCSAMP=03,MSFMSG=JOURNAL
```

## MVIKEY00 (continued)

### Internal Parameters

The following parameters are all optional and can be coded in any order.

Parameter	Meaning and Use	Value Range	Default Value
MSFMSG= $\left. \begin{array}{l} \text{xx} \\ \text{JOURNAL} \\ \text{xx,JOURNAL} \end{array} \right\}$	Specifies where the space manager messages regarding the volume inventory status are to be written. xx identifies the route code of the alternate console where they will be written. JOURNAL specifies that the Journal data set is to be used to record these messages. However, if the Journal data set fills up, subsequent messages are lost unless they are also directed to a console.	01-16	JOURNAL
MSSCSAMP= $\left\{ \begin{array}{l} \text{xx} \\ \underline{03} \end{array} \right\}$	Specifies how often the MSSC will sample the load on each of the staging drive groups. The MSSC determines the least and most-used groups and passes this information to Device Allocation so Device Allocation can best choose a 3330V unit when it must mount a new MSS volume.	00-99 minutes	03
MSVCCAT= $\left\{ \begin{array}{l} \text{xxxxxxxx} \\ \underline{\text{MSVICAT}} \end{array} \right\}$	Specifies the high qualifier name of the volume Inventory and volume control Journal data sets. The Journal data set must be identified in the (Master or user) VSAM Catalog as xxxxxxxx.MSVCJRNL and the Inventory data set must be identified as xxxxxxxx.MSVI. If a VSAM user catalog is used, then the name of this user catalog must be xxxxxxxx.	1-8 alphameric characters*	MSVICAT

---

\*First character must be *alphabetic* or national. Following characters can be *alphameric* or national (including the hyphen and 12/0 overpunch).

**MVIKEY00 (continued)**

<b>Parameter</b>	<b>Meaning and Use</b>	<b>Value Range</b>	<b>Default Value</b>
<b>MSVCCAT (continued)</b>	<p>The Inventory data set defaults to MSVICAT.MSVI and the Journal data set defaults to MSVICAT.MSVCJRNL both cataloged in the VSAM Master Catalog or the VSAM user catalog MSVICAT. MSVICAT is automatically generated in the MSS logic at sysgen.</p> <p><i>Note:</i> In an MP system, the Journal and Inventory data sets and the user catalog where they are cataloged must reside permanently on shared DASD. (For information on creating the Inventory and Journal data sets, see <i>OS/VS Mass Storage System (MSS) Services for Space Management.</i>)</p>		

## Member Name: PARMTZ

**Status:** Introduced in VS2 Release 1.

### Use of the Member

PARMTZ contains the time zone constant – the value in hours, minutes, and seconds by which the local time differs from Greenwich Mean Time (GMT), and the direction, east or west, from Greenwich. Time-of-Day Clock Management uses the time zone constant to calculate local time.

The time zone constant can be set (and placed in the CVT) at either sysgen or Initialization. At sysgen, the constant is set in the CVTTZ field of the CVT via the TZ keyword of the CTRLPROG macro.

The time zone constant can be overridden by adding a PARMTZ member to PARMLIB via the IEBUPDTE utility. At Initialization, if PARMTZ can be read, the PARMTZ value is placed in CVTTZ.

The time zone constant defaults to that specified at sysgen if:

- The value in PARMTZ is invalid, or
- There is no PARMTZ member, or
- The member cannot be read.

If no time zone constant is specified at sysgen and no PARMTZ member exists, or it can't be read, a time zone constant of zero is placed in the CVTTZ field of the CVT.

The operator can change the time-of-day clock by responding to a system message at Initialization (if such messages are not suppressed by the TOD keyword in COMMNDxx). Only local time can be changed after an IPL, by means of the SET command. The TOD clock remains unchanged. (See *Operator's Library: OS/VS2 Reference (JES2)* for information on setting local time.)

Operator responses at Initialization can be minimized if you specify TOD=NOPROMPT in the COMMNDxx member of parmlib. This keyword will cause the TOD clock verification messages to be suppressed. In spite of this suppression, however, the operator will be prompted in either of two cases:

- The TOD clock has not been set, or
- Multiple TOD clocks (in a multiprocessing configuration) are not synchronized.

**Parameter in IEASYSxx:** None  
(or entered by the operator)

## PARMTZ (continued)

### Syntax Rules

The following rules apply to the creation of PARMTZ by means of the IEBUPDTE utility:

- The member consists of one record (see examples that follow).
- The member uses the following syntax, whose parameters are explained in “Internal Parameters”.

$$\left\{ \begin{array}{c} E \\ W \end{array} \right\} ,HH[.MM[.SS]]$$

### Syntax Examples

```
E,01.48.32
W,11
W,10.00.59
```

### IBM-Supplied Defaults

No default parmlib member. However, the time zone constant defaults to the sysgen-specified value or to zero if no value was specified at sysgen.

### Internal Parameters

---

Parameter	Meaning and Use	Value Range	Default Value
E	Specifies a time zone east of Greenwich Mean Time (GMT).	Not applicable	None
W	Specifies a time zone west of Greenwich Mean Time (GMT).	Not applicable	None
HH	Specifies the number of hours deviation from GMT.	00–12	None
MM	Specifies the number of minutes. This is an optional parameter.	00–59	None
SS	Specifies the number of seconds. This is an optional parameter which is valid only if minutes (MM) is specified.	00–59	None

## Member Name: SMFPRMxx

### Status

A new member name replaces SMFDEFLT that was used in MVT and VS2 Release 1. Separate foreground options will no longer be specified, since foreground and background options are now the same.

### Use of the Member

SMFPRMxx contains parameters that define how the SMF facility will be used. The parameters are of two types, required and optional. The required parameters specify the job wait time limit and the system on which SMF is active. Optional parameters allow you to select record types, specify physical information about the data sets, permit operator modification, and specify whether exits are to be taken. NIP itself does not use the parameters. It passes the member name to Master Scheduler Initialization for use in SMF initialization.

*Note:* The SYS1.MANX and SYS1.MANY data sets must be cataloged on DASD. This is a new requirement. If they are not, the Master Scheduler will fail during Initialization. (For a full discussion of SMF requirements and usage, refer to *OS/VS System Management Facilities (SMF)*.)

**Parameter in IEASYSxx: SMF=xx**  
(or specified by the operator)

The two alphanumeric characters, represented by xx, are appended to SMFPRM to identify the SMFPRMxx member of parmlib. If the parameter is not specified either in IEASYSxx or by the operator, the default member SMFPRM00 is used.

### Syntax Rules

The following rules apply to the creation of SMFPRMxx by means of the IEBUPDTE utility:

- Use columns one through 71. Do not use columns 72–80, since these columns are ignored.
- Avoid embedded blanks.
- Separate consecutive parameters by comma.
- Enter each parameter in the format: keyword=value.
- Place each parameter completely on a record. That is, do not divide a parameter between consecutive records.
- Indicate continuation by placing a comma after the last entry on a record, followed by a blank before column 72.

#### *Syntax Example*

```
OPT=2,EXT=YES,BUF=4096,SID=A158,  
JWT=15,OPI=YES,MAN=ALL
```

## SMFPRMxx (continued)

### IBM-Supplied Defaults

Sysgen places the following parameters into the default member, SMFPRM00:

OPT=2,EXT=YES,JWT=10,BUF=2000,SID=H155,OPI=YES,MAN=ALL

You should modify this list according to your system requirements. You may place alternate values, plus additional values, in one or more alternate SMFPRMxx lists.

*Note:* You should add DSV=2 or 3 to the default list (or to an alternate list) if you intend to use the IEHUCAT utility to update your OS catalogs for use with MVT or VS2 Release 1.

### Internal Parameters

SMF parameters valid for MVS are described in the following table. Three previously valid parameters not shown in the table have been deleted: MDL, PRM, and ALT. The MDL parameter has been combined with the SID parameter. The PRM and ALT parameters, which formerly specified the volume or device for SYS1.MANX and SYS1.MANY, are no longer needed. (The SMF data sets must now be cataloged on direct access.)

---

Parameter	Meaning and Use	Value Range	Default Value
JWT=nnn	This is a required parameter that initially specifies the number of minutes a job is allowed to wait continuously. When the specified time limit has expired, the time limit exit routine (IEFUTL) is entered, if exits are to be taken. The limit value can be changed by IEFUTL.	1–999	10
SID=xxxx	This required parameter does what SID and MDL did in VS2 Release 1 and MVT. It specifies the system and model on which SMF is active, provided the installation modifies the default value.	four alphameric and/or special characters	H155 ( <i>Note:</i> This is not a typographical error.)
MAN= $\left\{ \begin{array}{l} \text{NONE} \\ \text{USER} \\ \underline{\text{ALL}} \end{array} \right\}$	<p>This “optional” parameter specifies the type of records to be written to the SMF data set(s).</p> <p><i>Note:</i> If MAN=ALL or USER, the BUF parameter is also required, and must not be zero or defaulted.</p> <p>NONE specifies that no records are to be written to the SMF data set(s), regardless of values specified for OPT, DSV, and REC parameters.</p> <p>USER specifies that only user records (types 128 through 255) are to be written to the SMF data set(s).</p>	not applicable	ALL

SMFPRMxx (continued)

Parameter	Meaning and Use	Value Range	Default Value
MAN= (continued)	<p>ALL specifies that both SMF-produced and user-produced records are to be written to the SMF data set(s). All SMF records are created unless suppressed by the OPT, DSV, or REC parameters.</p> <p><i>Note:</i> ALL should be specified if you intend to run the IEHUCAT utility to update OS catalogs.</p>		
OPT= $\left\{ \begin{array}{l} 1 \\ 2 \end{array} \right\}$	<p>This "optional" parameter specifies whether system and job information, as opposed to system, job, and job step information, is to be recorded.</p> <p>1 specifies that only system and job-related information is to be collected by SMF. The step-initiation exit, IEFUSI, and the job step termination exit IEFACTRT are not taken.</p> <p>2 specifies that system, job, and job step information is to be collected by SMF.</p> <p><i>Notes:</i></p> <ol style="list-style-type: none"> <li>If you wish the System Resource Manager to do I/O load balancing, you must specify OPT=2 so that the System Resource Manager can access EXCP counts by job step. OPT=2 is also necessary if you wish to run the IEHUCAT utility to update OS catalogs.</li> <li>If OPT=1 is specified, and the DSV=2 or DSV=3 is also specified, SMF converts OPT=1 into OPT=2 and issues a warning message.</li> </ol>	1-2	2
DSV= $\left\{ \begin{array}{l} 0 \\ 1 \\ 2 \\ 3 \end{array} \right\}$	<p>This "optional" parameter specifies whether data set information and/or direct access volume information is to be collected by SMF.</p> <p>0 specifies that neither data set information nor direct access volume information is to be collected.</p> <p>1 specifies that direct access volume information is to be collected and data set information is to be suppressed.</p> <p>2 specifies that data set information is to be collected and direct access volume information to be suppressed.</p>	0-3	0



### SMFPRMxx (continued)

Parameter	Meaning and Use	Value Range	Default Value
DSV = (continued)	<p>3 specifies that both data set information and direct access volume information are to be collected.</p> <p><i>Notes:</i></p> <ol style="list-style-type: none"> <li>1. You should specify DSV=2 or 3 if you plan to use the IEHUCAT utility to update your catalog for use with MVT or VS2 Release 1.</li> <li>2. If either DSV=2 or DSV=3 is specified and OPT=1 is also specified, SMF converts OPT=1 into OPT=2 and issues a warning message.</li> </ol>		
REC= { 0 } { 2 }	<p>This optional parameter specifies whether record type 17 will be written for temporary data sets. The parameter is not effective unless DSV is also specified as 2 or 3.</p> <p>0 specifies that record type 17 is to be written for non-temporary data sets only, and information is to be suppressed for temporary data sets.</p> <p>2 specifies that record type 17 is to be written for both temporary and non-temporary data sets.</p>	0 or 2	0
BUF= { nnn } { nnnn }	<p>This parameter specifies the size of the SMF buffer. It must be specified in the MAN parameter equals ALL or USER.</p> <p><i>Notes:</i></p> <ol style="list-style-type: none"> <li>1. You should specify a value that is a multiple of 8 bytes (double word). Otherwise, SMF rounds the value to the next <i>lower</i> multiple of 8 bytes.</li> <li>2. Before you reduce the buffer size specified in the previous IPL, you must dump the SMF data set(s). Otherwise, the data set(s) cannot be dumped.</li> </ol>	400 to 8,192	None (No default value exists. Operator is prompted if nonzero BUF value is needed and was not specified.)

### SMFPRMxx (continued)

Parameter	Meaning and Use	Value Range	Default Value
OPI = { YES } { NO }	This optional parameter specifies whether the SMFPRMxx parameters are to be presented on the console during IPL for the operator's inspection and/or modification.  <b>YES</b> specifies that the operator is allowed to modify parameters.  <b>NO</b> specifies that the operator is not allowed to modify parameters.	Not applicable	NO
EXT= { YES } { NO }	This optional parameter specifies whether SMF exits are to be taken. The parameter is independent of the value specified for MAN.  <b>YES</b> specifies that exits are to be taken.  <b>NO</b> specifies that exits are not to be taken.  <i>Note:</i> If EXT is specified as YES, the exits taken will depend on the data collection parameter OPT. If OPT=2, all exits defined for the system will be taken. If OPT=1, however, the job step initiation exit and the job step termination exit will not be taken.	Not applicable	YES

## Member Name: VATLSTxx (Volume Attribute List)

### Status

This member in MVT and VS2 Release 1 was called PRESRES. It was read during Master Scheduler Initialization. In MVS, it is read during general NIP processing. Principal changes from previous usage are as follows:

- Multiple members of the name VATLSTxx are supported.
- The format of a VATLSTxx entry differs from that of a PRESRES entry.
- Informational messages are issued only when errors are detected.

### Use of the Member

VATLSTxx member(s) contain the volume attribute list(s) that predefine the “mount” and “use” attributes of direct access volumes. The Volume Attribute Processor reads the volume attribute list(s) in order to set “mount” and “use” status bits in direct access UCBs\*.

The system programmer can predefine volume “mount” attributes as permanently resident or reserved, and can predefine volume “use” attributes as storage, public, or private. Critical direct access volumes can thus be controlled, since the “mount” and “use” attributes determine the type of data sets that can be placed on a volume. Data sets on volumes marked permanently resident or reserved receive preferential treatment during allocation. (See “Device Allocation” in “System Performance Factors” chapter.)

You can ensure a faster initialization by efficiently specifying the volume attribute list(s). Do not specify a list at a given IPL that contains entries for volumes that will not be mounted. Unmounted volumes require operator intervention with resultant delay.

### Use of 3344 and 3350 Emulated 3330-1 and 3330-11 Devices

The category of devices consisting of the 3344 emulated 3340 and the 3350 emulated 3330-1 and 3330-11 must be made permanently resident via the VATLST facility because of a conflict in mount states. Because these emulated devices cannot be demounted as can their real counterparts, the mount and demount messages that the operating system ordinarily issues for the real devices are erroneous. To prevent these erroneous messages, you must mark the emulated devices as permanently resident by so noting them in their respective VATLST entries.

Also, take care to ensure that all emulated devices necessary for a given IPL are ready and available (not held in reserve by another CPU) at IPL time. Specifically, if another CPU has an emulated device reserved at IPL time, the operator must reply “WAIT” to the message “IEAI20A DEVICE ddd SHARED. REPLY ‘CONT’ OR ‘WAIT’ ”.

---

\*“Mount” attributes are set in the UCBSTAT field, status byte A (offset 03) in the UCB. “Use” attributes are set in the UCBSTAB field, status byte B (offset 34 decimal, 22 hex) in the UCB.

## VATLSTxx (continued)

## Definitions of the Mount and Use Attributes

The *mount* attribute determines the conditions under which a volume can be demounted. There are three “mount” attributes: permanently resident, reserved, and removable. The permanently resident or reserved attributes may be specified in a volume attribute list. The removable attribute automatically applies to any volume that VATLSTxx does not designate or default as permanently resident or reserved.

A *permanently resident* volume is one that either can't be physically demounted (e.g., a drum, 3344, 3350), or can't be demounted until its device is varied offline. Only direct access volumes can be made permanently resident. The following volumes are *always* marked permanently resident by NIP. You should therefore specify only the *use* attribute of these volumes in a volume attribute list:

- Volumes that can't be physically demounted (e.g., a 2305 or 3350 volume).
- The system residence volume. This includes the SYS1.LOGREC, SYS1.SVCLIB, SYS1.NUCLEUS data sets.
- Volumes that contain these system data sets: SYS1.LINKLIB and data sets concatenated to it, SYS1.DSSVM, SYS1.DUMPxx, SYS1.STGINDEX, page data sets, and swap data sets.

*Note:* Although it is impossible to demount the devices physically, 3344 emulated 3340 devices and 3350 emulated 3330-1 and 3330-11 devices are *not* automatically marked permanently resident. You must instead, mark their respective VATLST entries yourself.

A *reserved* volume remains mounted until the operator issues an UNLOAD or a VARY OFFLINE command. A volume is marked reserved when it is so designated in a volume attribute list, or when the operator issues a MOUNT command for the volume.

A *removable* volume can be demounted after its last use in a job, or when the device on which it is mounted is needed for another volume. Any volume not designated as either permanently resident or reserved is considered removable. The operator can change a removable volume to a reserved volume by issuing the MOUNT command for the volume.

The *use* attribute controls the type of request for which a volume can be assigned: a specific volume request, a temporary, non-private non-specific volume request, or a non-temporary, non-private, non-specific volume request. Three *use* attributes are used for allocating these types of volume requests, as follows:

- A *private* volume is allocated only to a specific volume request. For further information on this attribute see *OS/VS2 JCL*.
- A *public* volume is allocated to a temporary, non-specific volume request (or possibly to a specific volume request). Thus, a scratch data set would be placed on a public volume.

### **VATLSTxx (continued)**

- A *storage* volume is allocated primarily to a non-temporary non-specific volume request. (A temporary non-specific volume request may also be allocated to a storage volume. A storage volume can also be allocated to a specific volume request.)

*Note:* A storage volume is required by the *SAVE* subcommand of *EDIT* for a newly created data set. If a *STORAGE* volume is not available, the *SAVE* subcommand cannot save the data set.

For additional information on the public and storage attributes, see *OS/VS2 System Programming Library: Job Management – Allocation Services* section.

### **Processing the VATLSTxx Member(s)**

Volume Attribute Processing reads the VATLSTxx member(s) that were specified in the *VAL* parameter. If an invalid VATLST entry is detected, an informational message (IEA855I) is issued, and processing continues with the remaining entries.

## VATLSTxx (continued)

If an I/O error occurs during the reading, the operator is given the option to receive an informational message (IEA850I) that lists all the volumes, device types, and attributes that will be processed. A second message (IEA853A) allows the operator to choose one of several recovery options:

- Continue processing any remaining lists.
- Stop the processing of remaining lists.
- Specify a new VATLSTxx member by replying r0,xx. (The xx is the two-character identifier for VATLSTxx.)
- If necessary, the operator can re-IPL the system.

Volume Attribute Processing compiles a list of all VATLSTxx entries. If a particular volume serial number appears on more than one entry, the volume attributes specified in the last entry for that volume serial will be accepted. If the volume serial does not duplicate another entry, but is for an MSS volume (see the 'device type' parameter description), then the last entry with that particular MSS unit address will be used to set the volume attributes for the volume.

When Volume Attribute Processing has set *mount* and *use* attributes for all mounted volumes specified in the VATLSTxx entries, it issues a macro to mount all unmounted MSS volumes. When all MSS volumes have been mounted and their attributes set, it issues mount messages (IEA851I and IEA851A) for entries that specify unmounted volumes (unless mount message suppression was requested in the entries). Mount messages can be issued for unmounted volumes, up to the maximum number of processed entries. As many as 300 unique VATLSTxx entries can be processed at one IPL.

The operator may respond to the mount messages by replying with a valid unit address of the requested device type (e.g., 3330-1 or 2305-1). If the operator chooses to mount no volumes, he replies 'U', END, or ENTER.

If the device addresses that the operator enters are invalid (e.g., invalid unit address, or a path to a device is not available), he can reenter new unit addresses. He may enter 'U', END, or ENTER to indicate that no more volumes will be mounted.

When message IEA860A lists the devices that need volumes, the operator should mount the required volumes on the replied devices. When all devices have become ready (green lights on – this is different from volume just being mounted), the operator replies 'U', END or ENTER to message IEA860A. Volume Attribute Processing then scans for mounted volumes.

If a volume that did not appear in the mount message is mounted on a unit specified by the operator, it is unloaded. The volume is also unloaded if the operator mounts the requested volume on a device type other than the one specified in the volume attribute list, or on an unrequested unit.

## VATLSTxx (continued)

If all the required devices do not become ready, Volume Attribute Processing issues message IEA893A, that lists the devices that are not ready. If the operator intends to ready these devices, he may do so before replying 'U', END, or ENTER to the message. If he cannot mount a volume on a device for some reason (e.g., hardware problem), he should reply NO to the message, after all other required devices have been processed. This response indicates to Volume Attribute Processing that no more volumes will be mounted.

(See *OS/VS Message Library: VS2 System Messages* for detailed information on Volume Attribute messages IEA850I-855, IEA859-861, IEA866, 867, IEA893-895, IEA947-949, and IEA985-987.

**Parameter in IEASYSxx: VAL=**  $\left\{ \begin{array}{l} \text{aa} \\ (\text{aa,bb}, \dots) \end{array} \right\}$   
(or entered by the operator)

Two alphameric characters (e.g., A1 or 30) are appended to VATLST to specify the VATLSTxx member(s) of parmlib. If the parameter is not specified either in IEASYSxx or by the operator, the default member VATLST00 is used, if it exists. (VATLST00 is built by the installation, not through sysgen.) If the VAL parameter specifies multiple members, the members are processed in the order specified. If a particular volume serial number appears on more than one entry, the volume attributes specified in the last entry for that volume serial will be accepted. If the volume serial does not duplicate another entry, but is for an MSS volume (see the 'device type' parameter description), then the last entry with that particular MSS unit address will be used to set the volume attributes for the volume. If the VAL parameter has invalid format, or if it specifies a member that doesn't exist in parmlib, the operator is prompted to respecify the member or to reply 'U' to cause the member to be ignored.

## Syntax Rules

The following rules apply to the creation of a VATLSTxx member by means of the IEBUPDTE utility:

- Each record consists of 80 columns, although columns 22 through 80 are ignored.
- The fields are column-dependent, as shown in "Internal Parameters" (below).
- There are only two required fields: the volume serial number and the device types. All other fields have defaults.
- Separate the adjacent fields by comma, except before the optional information field.
- Specify all characters in EBCDIC.

## VATLSTxx (continued)

### Syntax Example

Columns: 1            8 10 12            21 23  
           ↓           ↓ ↓ ↓           ↓ ↓  
           30565A,0,1,2305-2, PAGING VOLUME

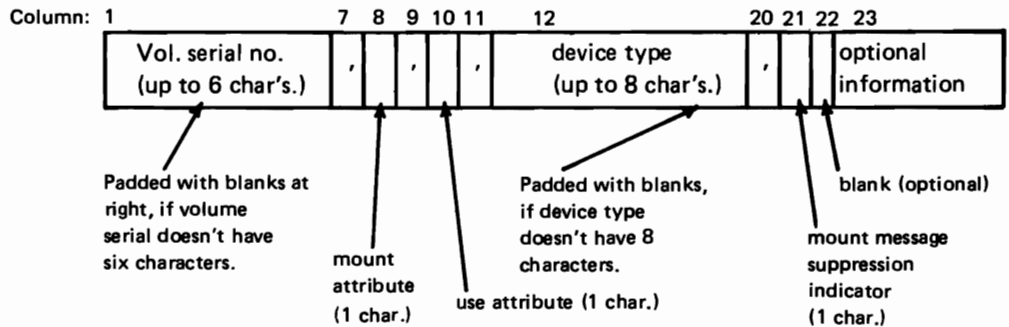
In this example, a volume whose serial number is 30565A is to be mounted on a 2305-2 and marked permanently resident. The volume's *use* attribute is to be *public*. A mount message is to be issued if the volume is unmounted. The optional information, "paging volume", is ignored but will appear in installation printouts.

### IBM-Supplied Defaults

No default member is supplied by IBM. The installation can, however, create its own default member, named VATLST00.

### Internal Parameters

The column dependency of fields and the associated separators are depicted in the following figure. Tabular description of fields follows the figure.





VATLSTxx (continued)

Parameter	Column	Meaning and Use	Value Range	Default Value
volume serial	1–6	Specifies the direct access volume whose <i>mount</i> and <i>use</i> attributes are to be set. The volume serial number must begin at the first character position in the record.	1 to 6 alphameric characters, left justified in the field, and padded with blanks at right to occupy six columns (see figure).	None
<i>mount</i> attribute	8	Specifies whether the volume is to be permanently resident or reserved. Default is assigned if any character other than 1 is specified. 0 specifies permanently resident. 1 specifies reserved.	0 or 1	0
<i>use</i> attribute	10	Specifies whether the volume is to be defined as storage, public, or private. The default is assigned if any character other than 0 or 2 is specified. 0 specifies storage. 1 specifies public. 2 specifies private.	0–2	1
device type	12–19	Specifies the device name, such as 2305-1. Up to eight characters may be specified, but the first character must start at column 12. Only supported device types are acceptable.* This parameter indicates the basic device but does not denote special features, such as track overflow. The operator must be informed if the device requires a special feature to process the data on the designated volume.  <i>Caution:</i> For 3330V volumes, do not specify “3330V” as you would specify another device. Instead, specify “Vxxx” where xxx is the unit address of the 3330V device that the volume is to be mounted on.	Up to 8 characters, left justified within the field, and padded with blanks at the right to occupy eight columns (see figure).	None

\*Supported device types are:

2314	3330 Mod 11	2305-1
3330	3340/3344	2305-2
3330-1	3350	
3330V, specified as Vxxx		

VATLSTxx (continued)

Parameter	Column	Meaning and Use	Value Range	Default Value
mount message suppression	21	<p>Specifies whether mount messages should be issued for the volume if it is not already mounted. This parameter is ignored for MSS volumes.</p> <p>N specifies that mount messages should be suppressed.</p> <p>b (or any character except N) specifies that mount messages should be issued.</p>	N, blank, or any character.	Blank (or any character except N), indicating that mount messages should be issued.
optional information	23–80	<p>Contains any desired descriptive information. The information is not used by the system. It appears in installation printouts.</p>	Not applicable.	None

## Part 2.1: Auxiliary Storage Management Initialization

This chapter gives guidelines for the effective use of page and swap data sets. Additional information on this subject appears under PAGE, SWAP, DUPLEX, and PAGNUM parameters of parmlib member IEASYSxx, and in the performance topic “The Pageable Link Pack Area: Its Advantages and Uses”.

### Page/Swap Operations

The paging and swapping controllers of the auxiliary storage manager (ASM) attempt to maximize I/O efficiency by incorporating a set of algorithms to distribute the I/O load as evenly as practical. In addition, every effort is made to keep the system operable in situations where a shortage of a specific type of page (swap) space exists. As a general rule, the best system performance is obtained when ASM is able to use the specific resource (data set) intended to be used for a specific operation. The following topics discuss some of the algorithms ASM uses to achieve these ends.

ASM

### Paging Operations and Algorithms

To page efficiently and expediently, ASM divides the pages of the system into classes, namely PLPA, common, and local. Contention is reduced when these classes of pages are placed on different physical devices. Additionally, some performance improvement can be obtained when the local page data set is divided into multiple local data sets, even though some devices may be large enough to hold the entire amount of necessary page space. The PLPA and common page data sets are both required data sets, and there can be only one of each. Because the system restricts the common area pages (PLPA, MLPA, CSA) to a maximum of eight megabytes, the combined space requirements of these data sets is a maximum of eight megabytes (excluding space for error handling). Spillage back and forth between these two data sets is permissible, but in the interest of performance, only spilling from PLPA to common should be tolerated.

Local page data sets are placed on one of two queues: the fixed head device queue or the movable head device queue. The general intent of the ASM algorithms for device selection and channel program construction is to:

- *Favor fixed head devices:* ASM attempts to write exclusively to fixed head devices, provided (1) they are defined, not in use, and contain free slots, and (2) the “service burst” (defined below) can contain all pending write requests.
- *Scan device queues in a circular fashion:* When attempting to write, ASM tries to use all devices of that type (fixed or movable head) equally. To accomplish this task, control blocks representing these devices are chained in a circular queue. Each time ASM writes a group of data, it selects the next available device containing free space in the chain.

- *Use write-between-read logic:* Once ASM selects a device, it intersperses read and write requests to accomplish the required I/O in the fewest possible revolutions of the device.
- *Use service bursting.* When ASM starts an I/O request for a paging data set, it starts only the number of requests that can be completed in a given “service burst” (currently set to 50 milliseconds). The number of pages read or written during a service burst varies as channel and device contention varies, thereby producing a “load balance”.
- *Use arm position information.* By remembering the previous arm position, ASM can often accomplish I/O with little or no arm motion on movable head devices.

*Note:* Because ASM has no control over other users, the remembered position may be erroneous when devices are shared with other users.

If a duplex page data set is specified on a CLPA IPL, ASM “duplexes” the pages written to the PLPA and common page data sets on the duplex data set to provide greater system reliability. If a read request for either the PLPA or common page data sets fails, ASM tries to read from the duplex page data set; if successful, system operation remains uninterrupted. Whenever the duplex page data set becomes full, ASM suspends duplexing but can read from that data set when the need arises.

If the user supplies no swap data sets, or if the swap data sets are temporarily filled, ASM writes the swap LSQA pages to a local page data set. This situation is inadvisable because it may reduce page or swap (and therefore system) performance.

## Swap Operations and Algorithms

ASM sends LSQA pages to special swap data sets as long as they are defined and contain free space. A swap data set consists of groups of 4096-byte slots called “swap sets”. Each swap set consists of twelve contiguous slots. Swap data sets use only one seek per swap set. To ensure this seek efficiency, ASM prevents the swap set from crossing cylinder boundaries and uses the direct access device multi-track feature.

There are two classes of data sets for swap operations: 1) fixed-head devices and 2) movable-head devices. ASM places a control block representing each data set on one of two circular queues (which represent either the fixed or movable head devices). ASM uses fixed-head devices for the swap operation as long as they are available and not full. ASM uses movable-head devices only when:

- no fixed head device is defined
- none are free
- none contain free space.

ASM frees swap sets immediately upon swap-in; that is, swap pages are valid on the swap data set only for that period between swap-out and swap-in.

## Page and Swap Data Set Sizes

Page and swap data set sizes can affect system performance. Note the following recommendations:

- *PLPA data set.* Where possible, the PLPA page data set should reside on a fixed-head device, or exclusively on a movable-head device because it is a high usage data set. If a fixed-head device is used for the PLPA data set, one can place other data sets on the device if enough space is available. However, if the common page data set has a higher use rate, it should be on the fixed head device in place of PLPA unless there is enough space for both data sets to be on a fixed head device.

The total combined size of the PLPA page data set and common page data set cannot exceed eight megabytes. In defining the size of these data sets, a reasonable starting value might be four megabytes each, as spilling will occur if the PLPA data set becomes full. After the system is running, RMF reports can be used to determine the exact size requirements of these data sets.

- *Common page data set.* The common page data set should be large enough to contain all of the common area pages, plus room for any expected PLPA spill. Although it is possible for the common page data set to spill to the PLPA page data set, this situation should not be allowed to occur because it may heavily impact performance. As noted for the PLPA data set, a reasonable starting size for the common page data set might be 4 megabytes. After the system is running, RMF reports can be used to determine the exact size requirements of this data set.
- *Duplex page data set.* The duplex page data set should be large enough to hold all the pages residing on both the PLPA and the common page data sets. A data set size of eight megabytes should therefore always be sufficient because the total combined size of the PLPA and common page data sets can approach eight megabytes. After the system is running, RMF can be used to determine the exact size requirement of this data set.
- *Local page data sets.* The local page data sets must be large enough to hold all of the private area and VIO pages of the system. To estimate the amount of space required for local page space, one should understand the ILRSLOT C and ILRSLOT V values.

### ILRSLOT C and ILRSLOT V Values

ASM reserves slots as backup for each address space created (defined here as a TSO terminal, batch address space, or VIO data set). The total slots available for user address spaces is equal to the total data set space defined as local. ASM calculates the number of slots to reserve as follows:

- 1) For a TSO terminal or batch address space, ASM computes the number of slots needed for mapping the user private area. ASM then divides this value by the constant ILRSLOT C in the nucleus CSECT ILRSLOT C (currently initialized to eight).

- 2) For a VIO data set, ASM uses the maximum relative page number value specified as part of the ASSIGN request for the number of slots required. ASM then divides this value by the constant ILRSLOTV in the nucleus CSECT ILRSLOTV (currently initialized to four).

As ASM creates each address space, it subtracts the anticipated number of slots for that address space (the reserve) from the available slot count. When there are no longer enough slots available, ASM rejects the creation of an address space.

**Example:** Expressed in terms of cylinders per paging data set, the reserve for each address space might be calculated as follows:

$$\frac{8 \text{ megabytes} - \text{average private area size}}{8 - \text{default constant}} = 1 \text{ megabyte of address space}$$

This one-megabyte reserve space is represented by 245 slots, which are equivalent to 4.2 cylinders on a 3330 paging device, or 8.2 cylinders on a 2314 paging device (see Figure 2.1-1).

Therefore, for each address space that ASM creates, the number of cylinders available for paging is reduced by 4.2 for a 3330 or 8.2 for a 2314.

#### Changing ILRSLOTC and ILRSLOTV Values

You can change the ILRSLOTC and ILRSLOTV values by using the AMASPZAP service aid so that ASM subtracts either more or fewer slots from the number of available slots when creating each address space. (The ILRSLOTC value is for batch address spaces, while ILRSLOTV is for VIO data sets.) For example, if you increase ILRSLOTC from eight to 16, then approximately 500,000 bytes of address space is subtracted from the number of available slots with the address space assignment. This is approximately 123 slots (2.2 cylinders on a 3330, or 4.1 cylinders on a 2314). Conversely, you can also decrease the slot values to reserve more slots. You can also modify the ILRSLOTC value to maintain the suggested one-megabyte reserve per address space when the private area size is not eight megabytes.

**Note:** Error message IEA890I, a 03C wait state, or a system completion code of 0E1 may indicate that the reserve count is exhausted.

**Example 1:** You can change the value of the ILRSLOTC and ILRSLOTV constants by using the AMSPZAP service aid. The following control cards change the value from eight to the value of  $x$ , or the value of four to the value of  $y$  (where  $x$  and  $y$  are installation-defined value):

NAME	IEANUC01	ILRSLOTC
VER	00	00000008
REP	00	0000000x
NAME	IEANUC01	ILRSLOTV
VER	00	00000004
REP	00	0000000y

**Example 2:** You can also use the HMASMP service aid to perform the same operation by preparing the following control cards:

```

++PTF(number)
++ZAP(ILRSLOT)  DISTLIB(AOSC5)
  NAME          ILRSLOT
  VER           00          0000,0004
  REP           00          0000,000x
  NAME          ILRSLOTV
  VER           00          0000,0004
  REP           00          0000,000y
    
```

- *Swap data set.* When defining a swap data set, one should allow enough space so that spillage to the local data set never occurs. Generally, one swap set (group of slots) per address space plus a buffer of 20-50% is sufficient. If, however, the LSQA for specific address spaces exceeds the swap set size (currently 12 pages), this value may need increasing. The number of swap sets per cylinder varies on different devices. Figure 2.1-2 provides device-related data for calculating the number of necessary cylinders. RMF reports can ultimately be used to verify correct values.

### Space Calculation Examples

Figures 2.1-1 and 2.1-2 show the values respectively for page and swap data sets. The examples following these figures show how to apply their tabular information to typical initialization considerations.

Device Category	Device Type	Slots/Cyl	Cyl/Meg	Slots/Device
fixed head	2305-1	25	10	1,200
	2305-2	26	10	2,496
movable head	2314	30	9	6,000
	3330	58	4.5	23,432
	3330-1	58	4.5	46,864
	3340	24	11	8,352
	3340-2	24	11	16,704
	3350	130	2	65,520 (max usable)

Figure 2.1-1. Page Data Set Values

Device Category	Device Type	Slots/Cyl	Swap Sets/Cyl	Swap Sets/Device
fixed head	2305-1	24	2	96
	2305-2	24	2	192
movable head	3330	57 48 used	4	1616
	3330-1	57	4	3232
	3340	24	2	696
	3340-2	24	2	1392
	3350	120	10	5500

*Note:* ASM uses space on a swap data set twelve slots at a time; all twelve slots must be in one cylinder. Therefore, the 3330 devices with 57 slots per cylinder available only use 48 (4 x 12) slots per cylinder for swap data sets.

Figure 2.1-2. Swap Data Set Values

**Example 1: Sizing the PLPA Page Data Set, Size of the PLPA Unknown**

Define the PLPA page data set to hold four megabytes; if that amount of space is exceeded, the remainder can be placed on the common page data set until the PLPA value is determined exactly.

*Therefore:* From the tables, 40 cylinders on a 2305 are defined for the PLPA page data set.

**Example 2: Sizing the PLPA Page Data Set, Size of the PLPA Known**

Define the PLPA page data set to hold 3.5 megabytes plus 5%, or 3.67 megabytes. (The extra 5% allows for loss of space as a result of permanent I/O errors.)

*Therefore:* From the tables, 37 cylinders on a 2305 is defined for the PLPA page data set.

**Example 3: Sizing the Common Page Data Set**

Assume that the exact size of the PLPA is unknown, but the PLPA page data set is defined to hold four megabytes. The combined sizes of PLPA and common page data sets need not exceed eight megabytes. For practical purposes, it is appropriate to use a total value of eight megabytes and consider the excess as a "buffer" value in case of errors. The total size (eight megabytes) minus the PLPA size (four megabytes) results in a definition of a four megabyte common page data set.

*Therefore:* From the tables, 88 cylinders on a 3340 is defined for the duplex page data set.



**Example 4: Sizing the Duplex Page Data Set**

Assume that the combined sizes of the PLPA and common page data sets cannot exceed the value of eight megabytes minus the size of the nucleus and SQA. Use the total size of eight megabytes and consider the extra space as a buffer in case of permanent I/O errors.

*Therefore:* From the tables, 88 cylinders on a 3340 is defined for the duplex page data set.

**Example 5: Sizing Local Page Data Sets**

Assume that the master address space and JES address space can each use about eight megabytes of private area storage. Next, determine the number of address spaces that will be used for subsystem programs such as VTAM, and allow eight megabytes of private area storage for each. To determine the amount of space necessary for batch address spaces, multiply the maximum number of batch address spaces that will be allowed to be active at once by the maximum size of that private area (IEALIMIT value if the exit is supplied by the user – see IEALIMIT in *OS/VS2 System Programming Library: Supervisor* – or estimated size of REGION parameter).

To determine the amount of space necessary for TSO, multiply the maximum number of TSO address spaces allowed on the system at once by the maximum size of that private area (IEALIMIT value if the exit is supplied by the user, or the estimated size of the REGION parameter).

Finally, estimate the space requirements for VIO data sets. Approximate this requirement by multiplying the expected number of VIO data sets used by the entire system by the average size of a VIO data set for the installation. After the system is fully loaded, you can use MF/1 or RMF reports to evaluate the estimates.

For example purposes, assume that the total space necessary for local data sets is:

8 megabytes for the master address space
8 megabytes for the VTAM address space
8 megabytes for the JES address space
6 megabytes for the batch address spaces (6 batches x 1 meg <sup>1</sup> each)
40 megabytes for TSO address spaces (40 TSO users x 1 meg <sup>1</sup> each)
+ 20 megabytes for VIO data sets (100 data sets x 0.2 meg each)
<hr/>
90 megabytes total + 4 meg (approx. 5%) buffer = 94 megabytes

*Therefore:* From the tables, 423 cylinders on 3330 type devices are necessary.

<sup>1</sup>These values are one megabyte because the eight megabyte private area is divided by the ILRSLOT default value (which is eight).

**Example 6: Sizing Swap Data Sets**

Assume that six batch address spaces and 50 TSO address spaces constitute the maximum number of address spaces to be active at any given time. At least one swap set is necessary for each, and some may require two. Additionally, define a few extra swap sets in case of disk pack errors. A good starting value is therefore 1.5 swap sets times the maximum number of address spaces, or in this example:

$$56 \times 1.5 = 84 \text{ swap sets}$$

*Therefore:* From the table, 42 cylinders on a 2305 is defined for the swap data set.

**Performance Recommendations**

The following recommendations may improve system performance through the careful use of paging data sets and devices:

- If you place multiple paging data sets on movable-head devices (3330s, for example) try to avoid placing more than one paging data set on any single movable-head device. Doing this reduces contention among multiple data sets for the use of the device.

**Reason:** When ASM starts I/O requests for a paging data set, it starts only the number of requests that can complete within a service burst (currently set to 50 milliseconds). If multiple paging data sets are on the same device, contention for use of the device is more likely compared with one paging data set per device, and ASM therefore adjusts the length of time (the service burst) allowed for request completion. Consequently, the request takes longer than expected, and fewer requests can start the next time.

**Comments:** You can experiment with multiple page data sets per device and check for device contention by executing MF/1 or RMF with typical job streams to obtain Direct Access Device Activity reports during various time intervals. The MF/1 or RMF data on device activity count, percent busy, and average queue length should suggest whether device contention is a problem. MF/1 or RMF data for devices that contain only one data set can be used as a comparison base.

You may, however, place low-activity non-paging data sets on a device that holds a paging data set. You can then use the MF/1 or RMF Direct Access Device Activity report to detect whether device contention is occurring.

- Specify a data set to contain the primary copy of PLPA exclusively. You can do this by specifying the data set as the first dsname in the PAGE parameter in IEASYSxx or as the first dsname in the first DATASET macro containing the PAGEDSN parameter during sysgen. (The sysgen process places the dsname in PAGEDSN into IEASYS00.) You should put the PLPA modules on the fastest available device because they are subject to high frequency usage. If real storage management steals a PLPA module page that ASM needs later, ASM will subsequently have to read the page from the PLPA page

**Example 2:** You can also use the HMASMP service aid to perform the same operation by preparing the following control cards:

```

++PTF(number)
++ZAP(ILRSLOTC)  DISTLIB(AOSC5)
  NAME           ILRSLOTC
  VER            00           0000,0004
  REP            00           0000,000x
  NAME           ILRSLOTV
  VER            00           0000,0004
  REP            00           0000,000y
    
```

- *Swap data set.* When defining a swap data set, one should allow enough space so that spillage to the local data set never occurs. Generally, one swap set (group of slots) per address space plus a buffer of 20-50% is sufficient. If, however, the LSQA for specific address spaces exceeds the swap set size (currently 12 pages), this value may need increasing. The number of swap sets per cylinder varies on different devices. Figure 2.1-2 provides device-related data for calculating the number of necessary cylinders. RMF reports can ultimately be used to verify correct values.

### Space Calculation Examples

Figures 2.1-1 and 2.1-2 show the values respectively for page and swap data sets. The examples following these figures show how to apply their tabular information to typical initialization considerations.

Device Category	Device Type	Slots/Cyl	Cyl/Meg	Slots/Device
fixed head	2305-1	25	10	1,200
	2305-2	26	10	2,496
movable head	2314	30	9	6,000
	3330	58	4.5	23,432
	3330-1	58	4.5	46,864
	3340	24	11	8,352
	3340-2	24	11	16,704
	3350	130	2	65,520 (max usable)

Figure 2.1-1. Page Data Set Values

Device Category	Device Type	Slots/Cyl	Swap Sets/Cyl	Swap Sets/Device
fixed head	2305-1	24	2	96
	2305-2	24	2	192
movable head	3330	57 48 used	4	1616
	3330-1	57	4	3232
	3340	24	2	696
	3340-2	24	2	1392
	3350	120	10	5500

*Note:* ASM uses space on a swap data set twelve slots at a time; all twelve slots must be in one cylinder. Therefore, the 3330 devices with 57 slots per cylinder available only use 48 (4 x 12) slots per cylinder for swap data sets.

Figure 2.1-2. Swap Data Set Values

**Example 1: Sizing the PLPA Page Data Set, Size of the PLPA Unknown**

Define the PLPA page data set to hold four megabytes; if that amount of space is exceeded, the remainder can be placed on the common page data set until the PLPA value is determined exactly.

*Therefore:* From the tables, 40 cylinders on a 2305 are defined for the PLPA page data set.

**Example 2: Sizing the PLPA Page Data Set, Size of the PLPA Known**

Define the PLPA page data set to hold 3.5 megabytes plus 5%, or 3.67 megabytes. (The extra 5% allows for loss of space as a result of permanent I/O errors.)

*Therefore:* From the tables, 37 cylinders on a 2305 is defined for the PLPA page data set.

**Example 3: Sizing the Common Page Data Set**

Assume that the exact size of the PLPA is unknown, but the PLPA page data set is defined to hold four megabytes. The combined sizes of PLPA and common page data sets need not exceed eight megabytes. For practical purposes, it is appropriate to use a total value of eight megabytes and consider the excess as a "buffer" value in case of errors. The total size (eight megabytes) minus the PLPA size (four megabytes) results in a definition of a four megabyte common page data set.

*Therefore:* From the tables, 88 cylinders on a 3340 is defined for the duplex page data set.

**Example 4: Sizing the Duplex Page Data Set**

Assume that the combined sizes of the PLPA and common page data sets cannot exceed the value of eight megabytes minus the size of the nucleus and SQA. Use the total size of eight megabytes and consider the extra space as a buffer in case of permanent I/O errors.

*Therefore:* From the tables, 88 cylinders on a 3340 is defined for the duplex page data set.

**Example 5: Sizing Local Page Data Sets**

Assume that the master address space and JES address space can each use about eight megabytes of private area storage. Next, determine the number of address spaces that will be used for subsystem programs such as VTAM, and allow eight megabytes of private area storage for each. To determine the amount of space necessary for batch address spaces, multiply the maximum number of batch address spaces that will be allowed to be active at once by the maximum size of that private area (IEALIMIT value if the exit is supplied by the user — see IEALIMIT in *OS/VS2 System Programming Library: Supervisor* — or estimated size of REGION parameter).

To determine the amount of space necessary for TSO, multiply the maximum number of TSO address spaces allowed on the system at once by the maximum size of that private area (IEALIMIT value if the exit is supplied by the user, or the estimated size of the REGION parameter).

Finally, estimate the space requirements for VIO data sets. Approximate this requirement by multiplying the expected number of VIO data sets used by the entire system by the average size of a VIO data set for the installation. After the system is fully loaded, you can use MF/1 or RMF reports to evaluate the estimates.

For example purposes, assume that the total space necessary for local data sets is:

8 megabytes for the master address space
8 megabytes for the VTAM address space
8 megabytes for the JES address space
6 megabytes for the batch address spaces (6 batches x 1 meg <sup>1</sup> each)
40 megabytes for TSO address spaces (40 TSO users x 1 meg <sup>1</sup> each)
+ 20 megabytes for VIO data sets (100 data sets x 0.2 meg each)
<hr/>
90 megabytes total + 4 meg (approx. 5%) buffer = 94 megabytes

*Therefore:* From the tables, 423 cylinders on 3330 type devices are necessary.

---

<sup>1</sup>These values are one megabyte because the eight megabyte private area is divided by the ILRSLOT default value (which is eight).

**Example 6: Sizing Swap Data Sets**

Assume that six batch address spaces and 50 TSO address spaces constitute the maximum number of address spaces to be active at any given time. At least one swap set is necessary for each, and some may require two. Additionally, define a few extra swap sets in case of disk pack errors. A good starting value is therefore 1.5 swap sets times the maximum number of address spaces, or in this example:

$$56 \times 1.5 = 84 \text{ swap sets}$$

*Therefore:* From the table, 42 cylinders on a 2305 is defined for the swap data set.

**Performance Recommendations**

The following recommendations may improve system performance through the careful use of paging data sets and devices:

- If you place multiple paging data sets on movable-head devices (3330s, for example) try to avoid placing more than one paging data set on any single movable-head device. Doing this reduces contention among multiple data sets for the use of the device.

**Reason:** When ASM starts I/O requests for a paging data set, it starts only the number of requests that can complete within a service burst (currently set to 50 milliseconds). If multiple paging data sets are on the same device, contention for use of the device is more likely compared with one paging data set per device, and ASM therefore adjusts the length of time (the service burst) allowed for request completion. Consequently, the request takes longer than expected, and fewer requests can start the next time.

**Comments:** You can experiment with multiple page data sets per device and check for device contention by executing MF/1 or RMF with typical job streams to obtain Direct Access Device Activity reports during various time intervals. The MF/1 or RMF data on device activity count, percent busy, and average queue length should suggest whether device contention is a problem. MF/1 or RMF data for devices that contain only one data set can be used as a comparison base.

You may, however, place low-activity non-paging data sets on a device that holds a paging data set. You can then use the MF/1 or RMF Direct Access Device Activity report to detect whether device contention is occurring.

- Specify a data set to contain the primary copy of PLPA exclusively. You can do this by specifying the data set as the first dsname in the PAGE parameter in IEASYSxx or as the first dsname in the first DATASET macro containing the PAGEDSN parameter during sysgen. (The sysgen process places the dsname in PAGEDSN into IEASYS00.) You should put the PLPA modules on the fastest available device because they are subject to high frequency usage. If real storage management steals a PLPA module page that ASM needs later, ASM will subsequently have to read the page from the PLPA page

data set. This read operation is faster if the PLPA page data set is on a fast device, for example a fixed-head device. Avoid placing other paging data sets on the same device unless you are using a fixed-head device (such as a 2305-2). For additional suggestions about the PLPA, see the topic “The Pageable Link Pack Area: Its Advantages and Use”.

**Note:** The common page data set or local page data sets may be placed on the fastest device if either or both show a higher use rate than the PLPA page data set’s use rate.

**Reason:** The PLPA is a high usage data set. It is desirable to speed the access to this data set and to avoid device contention. It may be possible to share the device with other performance-sensitive data sets, such as the primary spool volume (checkpoint records and spool messages queue only), provided contention does not develop. You may use the MF/1 or RMF Direct Access Activity report to check for possible device contention.

- Place most paging data sets on moderate speed devices, such as 33xx’s, to avoid overloading the faster devices. You should probably place only one paging data set (the PLPA) plus the primary spooling data set on the faster device (such as a 2305-2). An exception to this general rule involves the swap data set or paging data sets for a large time sharing or teleprocessing installation for which response time is critical. Such uses may require the fastest device obtainable.

**Reason:** The fastest devices should be used for the most critical applications. In general, this does not include local page data sets. However, to take advantage of the 33xx type of devices (reasonable speed plus size), ASM splits the paging load as evenly as possible across all defined local page data sets. To do this, ASM divides the local page data sets into two classes – fixed-head devices and movable-head devices. The control blocks for each of these classes are then arranged in a circular queue. ASM uses defined fixed-head devices as long as they contain free slots and are not in use. ASM scans the circular queue while work is pending. This causes all the data sets on the queue to be used in equal proportions. ASM also uses the movable-head devices in exactly the same way, but only if no fixed-head device can be selected.

**Note:** Using 2314s as paging devices is inadvisable because circular queueing causes them to be used as often as every other data set in the queue.

- Overspecify space for all page data sets to allow for the creation of additional address spaces before the deletion of current ones, and to permit some reasonable increase in the number of concurrent VIO data sets. VIO data-set growth may become a problem because there is no simple way to limit the total number of VIO data sets used by multiple jobs and TSO sessions. One method for eliminating VIO data sets is to re-IPL, specifying CVIO, but this action prevents warm start for the jobs that are using VIO data sets. (For additional space considerations, see the guideline for estimating paging space later in this topic.)

- Use multiple local page data sets even if the total required space is containable on one device.

**Reason:** When ASM uses multiple data sets, it can page concurrently on multiple devices. This is especially important during peak loads.

- Distribute ASM data sets among channels and control units.

**Reason:** Although ASM attempts to use multiple data sets concurrently, the request remains in the I/O supervisor queues if the channel or control unit is busy.

### Estimating Total Size of Paging Data Sets

You may obtain a general estimate of the total size of all paging data sets by considering the following space factors (a general formula for this calculation is in *OS/VS2 System Programming Library: Storage Estimates*).

1. The space needed for the common areas of virtual storage (PLPA, MLPA, and CSA). Double this space estimate if duplexing is desired.
2. The space needed for areas of virtual storage that are not duplexed: private areas of concurrent address spaces, and concurrently existing VIO data sets. (The system portion of concurrent address spaces needs to be calculated only once, since it represents the same system modules.)

### Using Measurement Facilities

You can possibly simplify the space estimation for the private areas mentioned above by picking an arbitrary value. Set up this amount of paging space, and then run the system with some typical job loads. To determine the accuracy of your estimate, start MF/1 or RMF while the jobs are executing. The Paging Activity Report of either measurement program gives data on the number of unused 4K slots, the number of VIO data set pages and address space pages, and the number of unavailable (defective) slots. (See the topic "Using the Paging Activity Report" in "Part 4: How to Use the System Activity Measurement Facility (MF/1)" for additional information.) The Auxiliary Storage User Pool portion of the MF/1 Paging Activity Report indicates a snapshot of slot usage at the end of the report interval.

The RMF version of the report replaces the snapshot values with average values based on a number of samples taken during the report interval. These average values are in a portion of the report entitled "Local Page Data Set Slot Counts". They are somewhat more representative of actual slot usage because slot usage is likely to vary during the report interval. The values from the Paging Activity Report of either measurement program should enable you to adjust your original space estimate as necessary.



## Adding More Paging Space

To add more paging space, you 1) must use the DEFINE PAGESPACE command of access method services to preformat and catalog each new page data set, and 2) may use the PAGEADD operator command or specify the data set at the next IPL by means of the PAGE parameter. (See *OS/VS2 Access Method Services* for information about the DEFINE PAGESPACE command, and on the related commands – ALTER and DELETE – used for the handling of VSAM data sets. Also see the description of the PAGE parameter in parmlib member IEASYSxx in “Part 2: System Initialization” in this manual.

## Questions and Answers

Customers have often asked the following questions about ASM:

*Q: What happens if there is an unrecoverable I/O error on a paging device ?*

*A:* If the error occurs during the reading of a common duplexed area and duplexing is active, the secondary copy is read. If an I/O error occurs during the reading of a non-duplexed area or if duplexing is not active, the job terminates, and all non-sharable devices are freed. If an error is detected on output, the pageout is retried to a different slot. However, since there is no readback check, ASM may not know if a data error occurred.

*Q: Does ASM use I/O load balancing ?*

*A:* Yes, it does its own, by keeping a history of the average length of time to read or write one page on each paging data set, then starting more or fewer page requests depending on how fast page requests are being satisfied by the device. By starting only 50 milliseconds' worth of requests to each page space, ASM keeps CCW strings relatively short, and therefore spreads resources among all paging data sets. ASM can react quickly to busy spurts on particular devices or channels and favor those devices or channels that are providing the best service.

*Q: How does the auxiliary storage shortage prevention algorithm in the SRM prevent shortages ?*

*A:* It does so by swapping out address spaces that are accumulating paging space at a rapid rate. Page space is not immediately freed, but another job or TSO session (still executing) will eventually complete and free page space. The SRM also prevents the creation of new address spaces and informs the operator of the shortage so that he can optionally cancel a job.

*Q: Is running out of auxiliary storage (paging space) catastrophic ?*

*A:* No, not necessarily; it may be possible to add more page data sets via the PAGEADD operator command. Additionally, after determining what kind of workload placed a heavy demand on paging space (TSO, batch storage, or VIO), it may be possible to modify key ASM constants (ILRSLOTC and ILRSLOTV) prior to re-IPLing with the same page data sets. Otherwise, it is necessary to re-IPL to specify an additional preformatted and cataloged page data set. (See the description of the PAGE parameter of the IEASYSxx member in the “System Initialization” chapter.)

*Q: Can we dynamically allocate more paging space ?*

*A:* Yes. Additional paging space may be added via the PAGEADD operator command if the PAGNUM parameter allowed for expansion (See the description of the PAGNUM parameter of the IEASYSxx member in the "System Initialization" chapter, and PAGEADD operator command in *Operators Library: OS/VS2 MVS JES2 Commands*).

*Q: Should 2314s be used as ASM data sets ?*

*A:* Generally speaking, the answer is no because 2314s tend to be too slow for most paging applications. The duplex page data set represents the possible exception to this general rule because this data set experiences few (if any) read operations. The activity to this data set after IPL is limited mostly to common area write operations (excluding PLPA).

*Q: How does ASM select slots ?*

*A:* ASM first selects a cylinder on a page space, maintaining a circular action of the disk arm from the beginning of the data set to the end, then starting over at the beginning again. ASM remembers the last arm position and selects the nearest cylinder beyond the one last processed. When it selects a cylinder, it chooses slots with the cylinder according to the best rotational position of eligible slots. ASM follows the same procedure for a 2305 device (reads from low-numbered slots are not favored over reads from high-numbered slots).

*Q: How does ASM select a local page data set for a page-out ?*

*A:* ASM selects a local page data set for page-out from either fixed-head or movable-head lists of available paging data sets. As long as fixed-head devices are available, are not in use, and contain free space, ASM uses these devices. If no fixed-head device is available, ASM uses the movable-head devices. ASM then selects the first available page data set and starts a 50-millisecond burst of work. ASM continues around the list until either there is no more work to do or no more page data sets are available.

## Introduction

To a large degree, an installation's control over the functioning of the OS/VS2 system is exercised through the system resources manager (SRM). The following sections describe the types of control available through the SRM, the functions used to implement these controls, the concepts inherent in the use of SRM parameters, and the parameters themselves.

Guidelines are presented for defining these parameters and several complete sample specifications are described. Finally, a list of potential problems along with guidelines for evaluation and adjustment of parameters are provided.

## System Tuning and the SRM

The task of tuning a system is an iterative and continuous process. The controls offered by the SRM are only one aspect of this process. Initial tuning consists of selecting appropriate sysgen parameters and parameters for various components and subsystems of MVS, such as TCAM and JES2. Once the system is operational and criteria have been established for the selection of jobs for execution via job classes and priorities, the SRM will control the distribution of available resources according to the parameters specified by the installation.

The SRM, however, can only deal with available resources. If these are inadequate to meet the needs of the installation, even optimal distribution may not be the answer – other areas of the system should be examined to determine the possibility of increasing available resources.

When requirements for the system increase and it becomes necessary to shift priorities or acquire additional resources, such as a larger processor, more storage, more terminals, etc., the SRM parameters may have to be adjusted to reflect changed conditions.

## Section 1: Description of the System Resources Manager (SRM)

The SRM is a component of the MVS control program. It determines which address spaces, of all active address spaces, should be given access to system resources and the rate at which each address space is allowed to consume these resources.

Before an installation turns to the SRM, it should be aware of the response time and throughput requirements for the various types of work that will be performed on its system. Questions similar to the following should be considered:

- How important is turnaround time for batch work, and are there distinct types of batch work with differing turnaround requirements ?

- Should subsystems such as IMS and CICS be controlled at all, or should they receive as much service as they need ? That is, should they be allowed unlimited access to resources without regard to the impact this would have on other types of work ?
- What is acceptable TSO response time for various types of commands ?
- What is acceptable response time for compiles, sorts, or other batch-like work executed from a terminal ?

(Guidelines for defining installation requirements are discussed in Section 3).

Once these questions have been answered and, whenever possible, quantified, and the installation is reasonably confident that its requirements do not exceed the physical capacity of its hardware, it should then turn to the SRM to specify the desired degree of control.

### Objectives

The SRM bases its decisions on two fundamental objectives:

1. To distribute system resources among individual address spaces in accordance with the installation's response, turnaround, and work priority requirements.
2. To achieve optimal use of system resources as seen from the viewpoint of system throughput.

An installation specifies its requirements for the first objective in a member of PARMLIB called the Installation Performance Specification (IPS). Through the IPS, the installation divides its types of work into distinct groups, called *domains*, assigns relative importance to each domain, and specifies the desired performance characteristics for each address space within these domains. The meaning and use of domains is discussed in detail in succeeding sections.

A secondary input to the SRM is another member of PARMLIB, the OPT member. Through a combination of IPS and OPT parameters, an installation can exercise a degree of control over system throughput characteristics (objective number 2). That is, the installation can specify whether, and under what circumstances, throughput considerations are more important than response and turnaround requirements when the need arises to make trade-offs between objective number 1 and objective number 2.

The SRM attempts to ensure optimal use of system resources by periodically monitoring and balancing resource utilization. If resources are underutilized, the SRM will attempt to increase the system load. If resources are overutilized, the SRM will attempt to alleviate this by reducing the system load or by shifting commitments to underutilized resources. Examples of such resources are the CPU, logical channels, auxiliary storage and pageable real storage.

## Types of Control

The SRM offers four distinct types of control to an installation:

- Domain Control
- Workload Control
- Dispatching Priority Control
- Throughput Control.

The first three derive their input from the parameters in the IPS; the fourth derives its input from both IPS and OPT parameters.

The remainder of this section describes these types of control and the functions used by the SRM to implement them.

### Domain Control

Domains are a mechanism that allows an installation to divide its types of work into distinct groups and thereby exercise individually tailored control over different types of work, such as batch work, IMS message processors, short TSO commands and long-running TSO commands.

Domains, therefore, are simply the means by which related types of work are grouped together, such that all work (address spaces) assigned to one domain has some common set of characteristics that differentiates it from the work assigned to other domains. What constitutes such a set of characteristics is entirely dependent on an installation's requirements. Work could be differentiated on the basis of execution characteristics such as short versus long-running and batch versus foreground, or according to different use characteristics such as IMS message processors and student or test programs. On the other hand, an installation may choose to use different user requirements as the basis for differentiating and divide its user population into domains, regardless of the execution characteristics of individual jobs. A mixture of the above considerations is, of course, equally possible.

Domain control enables an installation to do the following:

- Guarantee access to system resources to at least a minimum number of address spaces for each type of work.
- Limit the number of address spaces, for each type of work, that are given access to system resources.
- Assign degrees of importance to different types of work.

An installation may use these capabilities for various purposes, such as:

- To exercise either complete control, partial control, or no control at all over a particular type of work (or group of users).
- To prevent one type of work from competing with another type of work for access to system resources.
- To prevent one type of work from dominating the system.

The creation of domains, competition between domains, and the association of address spaces with domains are discussed in Section 2.

### Workload Control

Address spaces compete for access to system resources with all other address spaces associated with the same domain. The installation establishes the rules for this competition via IPS parameters that allow specification of desired performance characteristics for each address space. These specifications are used by the workload management functions of the SRM in apportioning system resources to individual address spaces.

An example of the competition within a domain would be a batch domain with different performance/throughput requirements for various initiator classes, such as payroll jobs versus test programs.

### Dispatching Priority Control

The SRM maintains control over a range of 16 dispatching priorities called the Automatic Priority Group (APG). By placing jobs in the APG range, further control over their performance characteristics is gained since this enables the installation, via the IPS, to alter the dispatching priorities of address spaces as their execution characteristics change.

Dispatching priorities control the rate at which address spaces are allowed to consume resources after they have been given access to these resources. This form of competition takes place outside the sphere of domain control, that is, all address spaces compete with all other address spaces with regard to dispatching priorities.

### Throughput Control

The SRM attempts to maximize system throughput. If contention for a system resource is very high, bottlenecks may result, impeding throughput. For instance, when contention for the CPU becomes excessive, the SRM will seek out an address space that is a major contributor to the problem and recommend that it be denied access to resources. Through an OPT parameter, the installation has the ability to specify the importance of such a load-adjusting recommendation. (The installation has the option of specifying, via the IPS, whether throughput considerations should play a role or not.)

Another example of throughput control is the SRM's enqueue processing. A user may request a serial resource that is held by another user. In this case, the SRM will ensure that the holder of the resource is in real storage and remains there for a fixed execution time interval in the hope that the serial resource is released before the user is swapped out. The installation has the ability to specify this time interval through an OPT parameter.

## Functions

This section discusses the functions used by the SRM to implement the controls described in the previous section. Constants referred to in this discussion are described in the section entitled “SRM Constants”.

There are seven such functions:

1. Swapping
2. Dispatching Priority Control
3. Resource Use Functions
4. Enqueue Delay Minimization
5. Device Allocation
6. Prevention of Storage Shortages
7. Pageable Frame Stealing.

### Swapping

Swapping is the primary function used by the SRM to exercise control over distribution of resources and system throughput. Using information specified by the installation via IPS and OPT parameters, and system status information that is periodically monitored, the SRM determines which address spaces should have access to system resources.

There are six distinct types of swapping. The first three described below are used for control of domains and the competition for resources between individual address spaces within a domain, while the remaining three provide control over system-wide performance aspects and help increase throughput.

**Unilateral swap in.** If the number of a domain’s address spaces that are in real storage is less than the number the installation specified, or less than the SRM considers optimal for the domain, the SRM will swap in additional address spaces from that domain, if possible.

**Unilateral swap out.** If the number of a domain’s address spaces that are in real storage is greater than the number the installation specified, or greater than the number the SRM considers optimal for the domain, the SRM will swap out address spaces from that domain.

**Exchange swap.** All address spaces of a domain compete with one another for system resources according to the installation’s specifications in the IPS and OPT. When an address space in real storage has exceeded its allotted portion of resources, relative to an address space of the same domain waiting to be swapped in, the SRM performs an exchange swap. That is, the address space in real storage is swapped out and the other address space is swapped in. This competition between address spaces is described in detail in Section 2.

**Swaps due to storage shortages.** Two types of shortages cause swaps: auxiliary storage shortages and pageable frame shortages. If the number of available auxiliary storage slots is low, the SRM will swap out the address space that is acquiring auxiliary storage at the fastest rate. Likewise, if the number of fixed frames is very high, the SRM will swap out the address space that acquired the greatest number of fixed frames. This process continues until the number of available slots rises above a fixed target, or until the number of fixed frames falls below a fixed target.

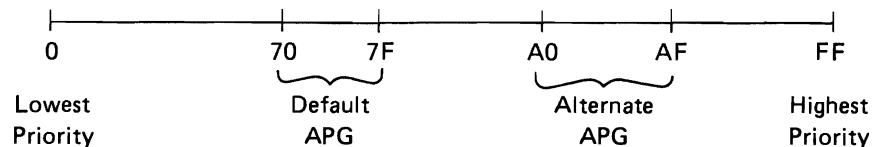
**Swaps due to wait states.** In certain cases, such as a batch job going into a long wait state (LONG option specified on the WAIT SVC, or a STIMER specification of  $> = 0.5$  seconds), the address space will itself signal the SRM to be swapped out in order to release storage for the use of other address spaces. Another example would be a time sharing user's address space that is waiting for input from the terminal after a transaction has completed processing. The SRM also detects address spaces in a wait state. That is, address spaces in real storage that are not executable for a fixed interval will be swapped out.

**Request Swap.** The system may request that an address space be swapped out. For instance, jobs which request real storage via the ADDRSPC=REAL specification on their job card, or nonswappable programs, will be swapped out and then swapped back in into a preferred area of real storage.

### Dispatching Priority Control

In MVS, dispatching of work is done on an address space priority basis. That is, those ready address spaces with the highest priority are dispatched first. When an address space is added to the dispatching queue, it is positioned according to its priority; if other address spaces with the same priority are already on the queue, it is added to the bottom of the group with that priority.

The total range of priorities is from 0 to 255. The range can be divided into 16 sets of 16 priorities each. The SRM maintains control over one such set, called the Automatic Priority Group (APG). The default APG range, as specified in PARMLIB member IEASYS00, is 70 to 7F (HEX). The installation may choose a different set of 16 values, such as A0-AF (it must start with x0,) as illustrated below.



Within the APG, specific numbers denote distinct types of priority schemes used by the SRM, enabling the installation to exercise precise control over the rules governing the dispatching of work. These rules also apply to nonswappable address spaces.

(Note: the default APG range is assumed in the remaining discussions and examples of "Part 3: The System Resources Manager.")



***Mean-time-to-wait group (70-76).***

These dispatching priorities can be used to increase system throughput by increasing CPU and I/O overlap. The SRM periodically monitors each address space's CPU utilization. Those address spaces that are considered to be CPU-bound are assigned lower dispatching priorities, within this group, than those which are considered to be I/O-bound. This permits the CPU-bound jobs to productively use the time spent waiting for I/O processing to complete.

***Rotate Priority (7B).***

This priority can be used to ensure that one address space does not dominate the CPU in relation to all other address spaces having the same dispatching priority. The SRM periodically repositions the address space that is highest on the dispatching queue at priority 7B to the bottom of the group of address spaces having that priority (7B). This means, for instance, that if two IMS control regions were active at the same time, using the Rotate priority would allow them to be dispatched on an alternate basis.

***Fixed Priorities (77-7A, 7C-7F).***

These priorities have no special significance other than that they belong to the APG and therefore are under the control of the SRM.

The implications and uses of SRM control over the APG will be discussed in later sections.

**Resource Use Functions**

The resource use functions of the SRM attempt to optimize the use of system resources on a system-wide basis, rather than on an individual address space basis. There are three such functions:

- Multiprogramming Level Adjusting
- CPU Load Adjusting
- I/O Load Adjusting.

The installation can influence or eliminate the effect of the multiprogramming level adjusting function via the parameter values of the domain specification in the IPS. The installation can influence or eliminate the effect of the CPU and I/O load adjusting functions with the response/throughput bias (RTB) and the resource factor coefficients (RFC) specifications in the IPS and OPT, respectively.

**Multiprogramming Level Adjusting:** The SRM monitors system-wide utilization of resources, such as the CPU and paging subsystem, and seeks to optimize it by alleviating imbalances, that is, overutilization or underutilization. This is accomplished by periodically adjusting the number of address spaces that are allowed in real storage (multiprogramming level) for appropriate domains.

For instance, when the paging rate is too high, it indicates that the percentage of CPU time used for productive (problem program) work is lower than it should be. That is, the system is being overutilized. To reduce this excessive contention for storage, the SRM will select a domain and reduce the number of address spaces allowed in real storage for that domain.

When system contention factors indicate that the power of the system is not being fully utilized, the SRM will select a domain and increase the number of address spaces allowed in real storage for that domain, thereby increasing utilization of the system.

**CPU Load Adjusting:** This function attempts to maintain a dispatchable job mix that neither underutilizes nor overutilizes the CPU. The SRM periodically monitors the system-wide CPU load to determine whether such imbalances exist.

The CPU is overutilized if, during the period under consideration, it did not enter the wait state, and the lowest priority address space on the dispatching queue was not dispatched (constant CCCUTHIT). The CPU is underutilized when its utilization is less than a fixed percentage (constant CCCUTLOT).

The SRM also monitors the CPU usage of individual address spaces. A threshold mean execution time before entering the wait state (constant CCCSIGUR) defines a heavy CPU user.

When the SRM determines that a CPU imbalance exists, it searches for heavy CPU users and calculates recommendation values for swap out (to correct overutilization) or swap in (to correct underutilization). These values are based on the extent to which the CPU load is out of balance.

**I/O Load Adjusting:** This function attempts to maintain a job mix that neither underutilizes nor overutilizes the system's logical channels. A logical channel is the set of all physical channels (paths) by which a device can be reached. When a request for I/O to a data set is delayed because of a channel busy condition, the delay is caused by the fact that all physical channels of the logical channel are busy. Thus the overuse of a physical channel is, in itself, of concern only when that physical channel is the only path to a device, and so constitutes the whole of the logical channel.

Each logical channel in the system is periodically monitored to determine whether its usage is above or below threshold bounds (constants ICCINHI1 and ICCINLO1). Every user in the system is periodically monitored to determine heavy I/O users (constant ICCSIGUP). When the I/O load adjusting function determines that there is an out-of-balance logical channel (that is, that the use of some logical channel falls outside the threshold limits), it recommends the swapping of a heavy user of that channel to alleviate the imbalance. The recommendation is based on the extent to which the address space makes use of out-of-balance logical channels and on the degree to which the channels are out of balance. Note that the percent of delayed requests is the control mechanism.

I/O load adjusting is dependent upon SMF data set activity recording being active. The parameter OPT in PARMLIB member SMFPRMxx must be specified as OPT=2 if I/O load adjusting is to be operative.

#### **Enqueue Delay Minimization**

This function deals with the treatment of address spaces enqueued upon system resources that are in demand by other address spaces.

If an address space controlling a resource is swapped out and that resource is required by another address space, the SRM will ensure that the holder of the resource is swapped in again as soon as possible. Once in real storage, a swap out of the controlling address space would increase the duration of the enqueue bottleneck. Therefore, the controlling address space is given a period of CPU service during which it will not be swapped due to service considerations (discussed in Section 2). The length of this period is specified by the installation by means of a tuning parameter called the Enqueue Residence Value (ERV), contained in PARMLIB member IEAOPTxx.

### Device Allocation

This function attempts to balance the utilization of all logical channels by selecting the proper device for allocation to an address space from a list of candidate devices. Because VIO satisfies many temporary data sets in MVS, device allocation primarily selects candidates for permanent data sets on mountable devices (JCL specifies non-specific VOLUME information).

Device allocation supports the I/O load adjusting function. The former tries to equalize logical channel utilization while the latter makes swap recommendations to keep logical channel utilization within predetermined bounds.

The installation's decisions at sysgen time (the OPTCHAN parameter of the IODEVICE macro, DEVPREF on the SCHEDULR macro, and the UNITNAME macro) and at IPL time (the number and placement of online mountable devices) greatly affect the effectiveness of the device allocation function.

Since device allocations tend to be made at the beginning of a job's execution, the future impact of that job's I/O activity on logical channel utilization is not known when the allocations are made. Therefore, simply choosing the least utilized logical channel would cause all or most allocations for a job to be made to the same logical channel. This effect is known as 'clumping'. The device allocation function attempts to avoid this effect by assuming that each data set will have a fixed impact on logical channel utilization. Thus, when making an allocation choice for a user, a constant (constant ICCEDSUT) is added to the measured utilization of a logical channel for each data set previously allocated to the job on that channel.

### Prevention of Storage Shortages

The SRM periodically monitors the availability of three types of storage and attempts to prevent shortages from becoming critical. The three types of storage are:

- Auxiliary storage
- Pageable frames
- SQA.

**Auxiliary storage:** When more than a fixed percentage (constant MCCASMT1) of auxiliary storage slots have been allocated, the SRM will reduce demand for this resource by taking the following steps:

- LOGON, MOUNT and START commands are inhibited until the shortage is alleviated.
- Initiators are prevented from executing new jobs.
- The number of address spaces allowed in real storage for each domain is set to its minimum value (see "Domains" in Section 2).
- The address space(s) acquiring this resource at the fastest rate is swapped out.
- The operator is informed of the shortage and the name(s) of the job(s) most responsible for it is listed, permitting cancellation.

If the number of available auxiliary slots continues to decrease (constant MCCASMT2), the SRM informs the operator that a critical shortage exists.

When the shortage has been alleviated, the operator is informed and the SRM halts its efforts to reduce the demand for auxiliary storage.

**Pageable Frames:** When the number of available pageable frames falls below a threshold (constant PVTPERFX) due to too many fixed frames, the SRM will alleviate this shortage by taking the following steps:

- LOGON, MOUNT and START commands are inhibited until the shortage is alleviated.
- Initiators are prevented from executing new jobs.
- The number of address spaces allowed in real storage for each domain is set to its minimum value (see "Domains" in Section 2).
- The address space(s) holding the greatest number of fixed frames is swapped out.
- The operator is informed that a pageable frame shortage exists and the name(s) of the job(s) most responsible for it is listed, permitting cancellation.

If the number of fixed pages continues to increase, the SRM informs the operator that a critical shortage of pageable frames exists.

When the shortage has been alleviated (constant PVTPEROK), the operator is informed and the SRM halts its efforts to reduce the demand for pageable frames.

**SQA:** When the number of available SQA and CSA frames falls below a threshold, the SRM will take the following steps:

- LOGON, MOUNT, and START commands are inhibited until the shortage is alleviated.
- The operator is informed that an SQA shortage exists.

If the number of available SQA and CSA frames continues to decrease, the SRM informs the operator that a critical shortage of SQA space exists.

When the shortage has been alleviated, the operator is informed and the SRM halts its efforts to prevent acquisition of SQA space.

(*Note:* For a discussion of possible operator response to the messages described above, see the SRM section, IRAXxxx messages, in the publication *OS/VS Message Library: VS2 System Messages, GC38-1002.*)

### Pageable Frame Stealing

Pageable frame stealing is the process of taking an assigned frame away from an address space to make it available for other purposes.

When the demand for pageable frames is excessive, the SRM will steal those frames that have gone unreferenced for the longest time and return them to the system. The Unreferenced Interval Count (UIC) of each frame indicates how long it has been since an address space referenced it. This count is updated periodically by the SRM.

Stealing takes place strictly on a demand basis, that is, there is no periodic stealing of long-unreferenced frames.

## Section 2: Introduction to SRM Parameter Concepts

This section discusses the concepts inherent in the IPS and OPT parameters that control the SRM's distribution of system resources to individual address spaces. Parameter descriptions and syntax rules are provided in Section 5 and should be referred to when necessary.

The section entitled "Guidelines and Examples" discusses recommendations for the selection of specific parameter values, but these will not be meaningful unless the ideas presented in this section are understood.

Although care has been taken to choose typical numbers for the examples in this section, these numbers are not intended as guidelines for actual cases.

The parameters are presented in the following order:

- domains
- service definition coefficients
- workload levels
- service rates
- IPS ● performance objectives
- interval service values
- dispatching priorities
- response-throughput bias
- performance periods
- performance groups
- OPT ● resource factor coefficients
- enqueue residence values.

In addition to these parameters, the meaning of terms such as service, service units, time interval, and transaction is discussed in detail where necessary to support the discussion of related parameters.

This entire section should be studied as one continuous presentation because the ideas introduced under each topic are dependent on the concepts, terms, and examples presented under the preceding topic. The examples, for instance, are continuously expanded with each newly introduced parameter until, under the topic "Performance Group", a complete IPS has evolved. At that point, the official syntax is used for the first time, requiring the reader to refer to Section 5.

The discussion of OPT concepts is also dependent on preceding topics, that is, on IPS parameter concepts.

It is assumed that the reader is familiar with the basic mathematical concepts inherent in the plotting of the relationship between dependent and independent variables on a graph.

## IPS Concepts

The parameters discussed in this part are specified in the IEAIPSxx member of SYS1.PARMLIB.

### Domains

Domains, as discussed under "Domain Control" in Section 1, are a means to differentiate one type of work (or user) from another and thereby make it possible to establish different, individually suited, control over different types of work. This control is derived from the ability to specify the number of address spaces that should be allowed in real storage at one time for a domain. This number is called the *multiprogramming level*, or MPL, of a domain. Via the IPS, the installation specifies the desired range for each domain's multiprogramming level, that is, the minimum and the maximum MPL (minMPL and maxMPL, for brevity).

The SRM periodically computes for each domain an optimal MPL called the *target MPL*, adjustable up or down in response to changing activity levels.

The target MPL is the basis for SRM domain control since it represents the actual number of a domain's address spaces that are allowed in real storage at any one time.

The range of the target MPL is subject to the installation-specified constraints (minMPL and maxMPL), that is, it cannot be less than minMPL or greater than maxMPL, even when the computed optimal value exceeds these bounds. The installation, therefore, has the ability to select the degree of control it wishes to exercise over each domain, ranging from complete control to no control at all. More control by the installation, of course, implies less control by the SRM.

Complete control is achieved by setting minMPL equal to maxMPL, resulting in a "fixed" target MPL and therefore eliminating the SRM's ability to make MPL adjustments in response to changing activity levels.

Partial control is achieved by choosing different values for minMPL and maxMPL, thereby allowing room for SRM adjustment of the target MPL.

No installation control at all, or full SRM control, is the result of setting minMPL and maxMPL to their extreme values: minMPL = 0 and maxMPL = 255 (see Section 5 for value ranges).

**Domain Examples:** Some basic examples will now be discussed to illustrate the purpose and use of domains and the effects achieved by the selection of particular constraints.

*Example 1.*

A domain could be created for work that, for reporting purposes, should not be mixed with other types of work. For instance, nonswappable address spaces, or address spaces associated with different subsystems, could be grouped into unique domains. Because the Resource Measurement Facility (RMF) provides workload information on a domain basis, placing such work into unique domains provides distinction for analysis purposes. That is, it enables the installation to determine how much of the total system is being used by a particular nonswappable address space, or by a particular subsystem.

Also, since APG priorities may be assigned to nonswappable address spaces, the effects on performance resulting from changes to the dispatching priority of such an address space can be evaluated by examining the RMF workload report for the respective domain.

*Example 2.*

A domain could be created for work that has particularly fast response time or turnaround requirements, such as short (interactive) TSO commands. Placing such work into a unique domain with minMPL equal to the maximum number of ready users will ensure that each address space is swapped in as soon as it becomes ready. That is, forcing the target MPL to be equal to the maximum number of ready address spaces guarantees all address spaces of that domain immediate access to real storage.

*Example 3.*

A domain could be created for users that might dominate the system. The maxMPL value can be used to limit the number of address spaces of this type of work allowed in real storage at one time.

For example, TSO users executing sorts or compiles from the terminal can slow down the system considerably. Placing such users into a domain with minMPL = maxMPL = 1 would have the effect of serializing such work, since only one address space of this type would be allowed in real storage at one time.

Batch work could be controlled in the same manner. Placing all batch work into a unique domain with maxMPL = 2 will limit the number of batch jobs concurrently in real storage to two, even if, for example, eight initiators were started.

**Example 4.**

The MPL adjustment function can set the target MPL of a domain to a maximum that is equal to the average number of ready address spaces plus one, unless prevented from doing so by the domain constraints. The adjustment function is most effective when the arrival rate of work is somewhat consistent, that is, when the number of ready address spaces over a time interval does not vary sharply.

The arrival rate of batch work, for example, is fairly consistent, since it is approximately equal to the number of started initiators. TSO work, however, can exhibit drastic fluctuations in arrival rate, causing considerable variations in response times if the MPL adjustment function is allowed to determine the target MPL.

Consider, for example, a domain with minMPL = 1 and maxMPL = 10. If TSO work is arriving at a rate of 10 address spaces every 30 seconds (that is, 10 commands are entered), the SRM will keep the target MPL near the minimum of 1, since, on the average, less than 1 address space is ready every second. Assume now that 15 users issue TSO commands at the same time. The MPL adjustment function does not respond fast enough to this sudden burst of activity since it computes the target MPL at fixed intervals, resulting in erratic response times for some of the 15 users.

**Domain Importance:** As discussed in Section 1, the SRM monitors system utilization and periodically adjusts some domain's target MPL in response to changes in the system load.

An installation has the ability to influence the SRM's decision as to which domain's target MPL will be increased or decreased by assigning to each domain a *weight*, or priority, via the IPS domain specification.

Giving a domain a high weighting factor relative to other domains implies that the work assigned to that domain should be given preference when the SRM considers the system underutilized and seeks to raise some domain's target MPL to allow more address spaces in real storage.

The SRM's decision is based on each domain's contention index, which is computed by the formula:

$$\text{contention index} = \frac{\text{average ready users} * \text{weight}}{\text{target MPL}}$$

The contention index is used as follows:

1. If the SRM determines that the total number of swapped-in address spaces should be increased, it selects the domain that a) has the highest contention index, b) has not yet reached its maxMPL, and c) has a target MPL less than or equal to its average number of ready users; and increases that domain's target MPL by one.



2. If the SRM determines that the total number of swapped-in address spaces should be decreased, it selects the domain that a) has the lowest contention index and b) has not yet reached its minMPL; and decreases that domain's target MPL by one.
3. If the SRM determines that the total number of swapped-in address spaces should not be increased or decreased, it will attempt to equalize the domains' contention indexes by periodically increasing the target MPL for the highest contention domain (thereby decreasing that contention index), and decreasing the target MPL for the lowest contention domain.

### Service

One of the basic functions of the SRM is to monitor the dynamic performance characteristics of all address spaces under its control in order to ensure distribution of system resources as intended by the installation.

A fundamental aspect of these performance characteristics is the *rate* at which an address space is receiving *service* relative to other address spaces competing for resources within the same domain.

Before discussing the meaning of service, a brief explanation of terminology is in order to avoid confusion. We have associated performance characteristics with address spaces. It is, of course, not the address space that has performance characteristics but the transaction associated with the address space. A transaction is simply the SRM's way to measure the service consumption within an address space, that is, it is a way to delineate a unit of work that is consuming service. For batch jobs, a transaction corresponds to a job or job step. For TSO work, a transaction normally corresponds to a command. (See Note.)

Since only one transaction can be active in an address space at one time, and it is indeed the address space that is swapped, the use of the term "address space" is more convenient when the emphasis of the discussion, for example, is on swapping. For the remainder of this section, the two terms will be used interchangeably, whichever is more appropriate.

*Note:* A new transaction is defined for a batch job step if it is the first step of the job, or if the performance group number (PGN, discussed later) is different from the previous job step's PGN.

A new transaction is defined for an active TSO user whenever

1. terminal input is entered and the line is not continued;
2. the 3270 field mark key separates commands on the same input line;
3. a command's output is detained while waiting for a TCAM output buffer.

Thus, a CLIST is one transaction, unless case 3 applies; every command typed ahead on an unlocked keyboard results in a new transaction.

The amount of service consumed by an address space is computed by the formula:

$$\text{service} = \text{CPU} * \text{CPU Service Units} + \\ \text{IOC} * \text{I/O Service Units} + \\ \text{MSO} * \text{Storage Service Units}$$

where CPU, IOC and MSO are installation defined *service definition coefficients* and:

CPU Service Units = task (TCB) execution time, divided by an SRM constant which is CPU model dependent. (Refer to "Selecting Service Definition Coefficients" in Section 3).

I/O Service Units = sum of individual SMF data set activity EXCP counts for all data sets associated with the address space.

Storage Service Units = (real page frames) X (CPU service units) x 1/50, where 1/50 is a scaling factor designed to bring the storage service component in line with the CPU component.

### Service Definition Coefficients

The service definition coefficients are used to assign additional weight to one type of service relative to another, allowing the installation to specify which type of resource consumption should be emphasized in the calculation of service rates. For example, an IPS may contain the following service definition coefficients:

$$\text{CPU}=10.0 \quad \text{IOC}=5.0 \quad \text{MSO}=3.0$$

In this case, if an address space has accumulated 100 CPU service units, 200 I/O service units and 300 storage service units, its total accumulated service would be:

$$(10 * 100) + (5 * 200) + (3 * 300) = 2900 \text{ service units}$$

### Service Rate

The rate at which service is consumed is computed by the formula:

$$\text{service rate} = \frac{\text{service}}{\text{time interval}}$$

where "service" is defined by the equation above and "time interval" is a combination of specific time periods in the life of a transaction that are defined as follows:

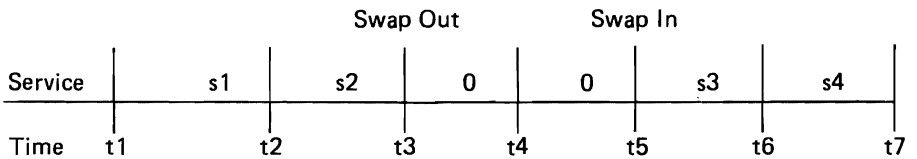
Transaction Resident Time — swapped in time  
 Transaction Out Time — swapped out (ready) time  
 Transaction Long Wait Time — swapped out (not ready) time  
 Transaction Active Time — Total Resident Time + Total Out Time  
 Transaction Elapsed Time — Total Resident Time + Total Out Time + Total Long Wait Time

“Time interval” in the service rate formula has two possible meanings:

- a. For a swapped-in address space –  
time interval = last Out Time + current Resident Time.
- b. For a swapped-out address space –  
time interval = last Resident Time + current Out Time.

If the Out Time is not factored into the time interval, the service rate is referred to as *absorption rate*.

Example of the calculation of service rate:



time t1: The transaction is starting. It has not used any service. Therefore, the service rate is 0.

time t2: The service rate equals  $\frac{s1}{t2 - t1}$ .  
The absorption rate at this time equals the service rate.

time t3: The service rate equals  $\frac{s1 + s2}{t3 - t1}$ .  
Assume the address space is now swapped out.

time t4: The service rate calculated while swapped out is  $\frac{s1 + s2}{t4 - t1}$ .

time t5: The service rate is  $\frac{s1 + s2}{t5 - t1}$ .  
Assume the address space is swapped back into storage now. The service rate is reset to 0.

time t6: The service rate calculated during this in-storage interval is  $\frac{s3}{t6 - t5}$ .  
The absorption rate now is  $\frac{s3}{t6 - t5}$ .

time t7: The transaction completes. Its final service rate is  $\frac{s3 + s4}{t7 - t3}$ .

The total service used by the transaction is  $s1 + s2 + s3 + s4$ .  
Its total resident time is  $(t3 - t1) + (t7 - t5)$ .  
Its total active time is  $(t7 - t1)$ .

Since the transaction did not incur any long wait time, its total elapsed time is the same as its active time.

### Performance Objective and Workload Level

The service rate, however, is only one aspect of an address space's performance characteristics. It does not, for instance, allow consideration of the importance of a particular address space. That is, if address space A and address space B are both receiving 100 service units per second, this may represent "good" service for address space A but "bad" service for address space B.

This need for the inclusion of relative importance in the process of comparing the service received by competing address spaces leads to the more comprehensive concepts employed by the SRM: *workload levels* and *performance objectives*.

A workload level specification consists of a set of positive integers of increasing magnitude, such as:

workload levels = (1,50,100)

A performance objective consists of a set of service rates and is assigned a unique identifier, such as:

objective 1, service rates = (1000,500,0)

A performance objective associates a set of service rates with a set of workload levels such that there exists, by definition, a workload level number corresponding to each specified service rate. The graph of this relationship looks as follows:

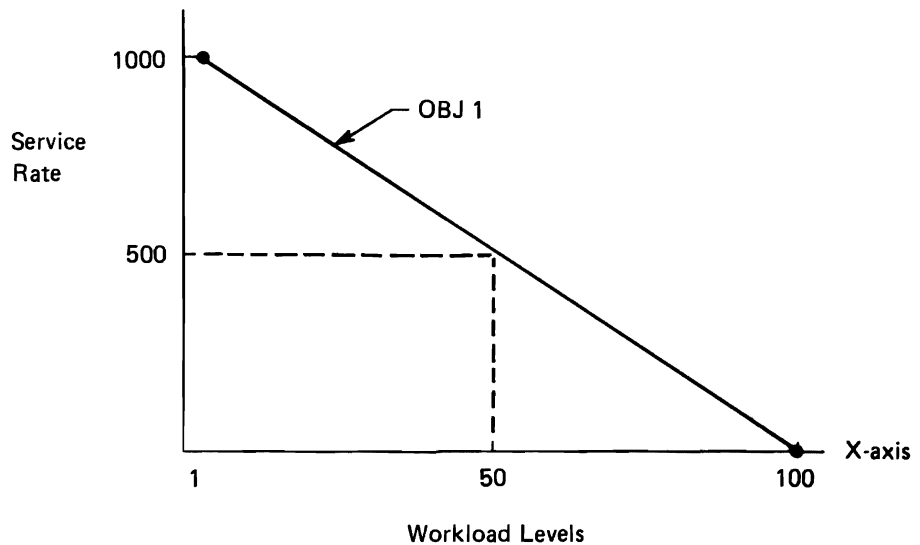


Figure 3-1. Relationship between Workload Levels and Service Rates

For every service rate on this graph, there is a corresponding workload level.

This graph is constructed by the SRM from the workload level numbers and service rates specified in the IPS. The *specified* service rates are used only to construct the graph, that is, to draw the line expressing the desired relationship between *measured* service rates and workload levels. The workload levels represent a fixed common scale into which an address space's measured service rate is mapped (the purpose of this mapping is discussed later). Thus, the measured service rates are the independent variables while the workload levels are the dependent variables. (Note that the x-axis in this case represents the dependent variable, contrary to conventional usage.)

Now let us choose a second performance objective:

objective 2, service rates = (2000,1000,0)

Figure 3-2 shows this objective plotted against the same workload level numbers (there can be only one workload level specification per IPS).

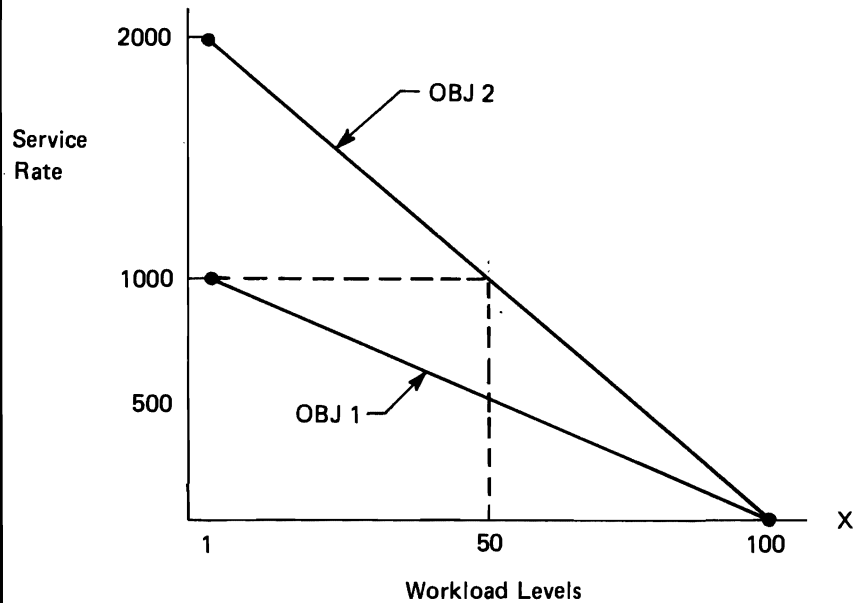


Figure 3-2. Comparing the Slopes of Two Performance Objectives

Now suppose that address space A is associated with performance objective 1 and address space B with performance objective 2, and that both address spaces, at some point in time, are receiving a service rate of 1000. For address space A, the corresponding workload level is 1, but for address space B it is 50.

These numbers, 1 and 50, are the basis for comparing the actual service rates. Thus, by associating address spaces with different performance objectives and plotting these against a common scale, the importance of each address space is factored into the comparison.

In this case, if address space A is in real storage (swapped in) and address space B is waiting to be swapped in, the SRM will assign a swap recommendation value of 50 to address space B and recommend that an exchange swap be performed (see Section 1, Swapping).

Consider a third performance objective:

objective 3, service rates = (2000,2000,2000)

Since this objective does not end with a zero service rate specification, the SRM will extrapolate to determine the zero point, or *cut-off level*, according to the following rules:

- The slope of the last specified segment of the graph is used and that segment is extended to the cut-off point (service rate = 0), if possible.
- The maximum cut-off point is  $3/2$  of the highest specified workload level number.

For performance objective 3, the cut-off level is  $3/2 * 100 = 150$ , resulting in the following graph when combined with Figure 3-2:

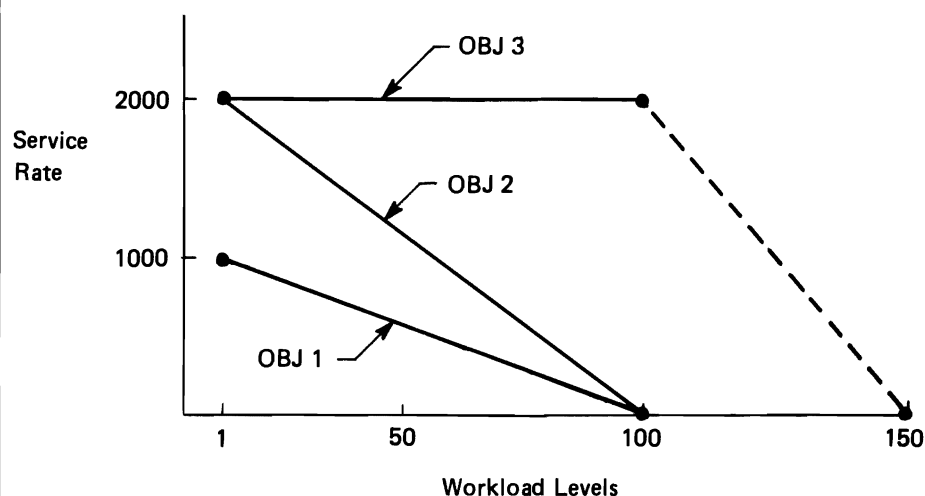


Figure 3-3. The Effect of Different Performance Objective Slopes on Service

For any service rate ( $< 2000$ ) on this graph, address space C, associated with objective 3, will have a workload level greater than address spaces A and B. In effect, this insulates address space C from exchange swap considerations relative to any address space associated with objective 1 or objective 2, within the same domain. (The workload level corresponding to a service rate of 2000 for objective 3 is 100, not 1 or 50.)

Note that performance objectives are not allowed to slope upward. That is, since each succeeding service rate within a performance objective is associated with a higher workload level, no service rate may exceed a preceding one.

Let us now examine how the address spaces of a domain compete with one another on the basis of workload levels.

Consider a domain with a target MPL of 1 and two address spaces, both associated with objective 1 of Figure 3-1, and let the abbreviation Adsp1 (obj1) denote that address space 1 is associated with objective 1.

Assume that Adsp1(obj1) becomes ready first and is therefore swapped in with a workload level of 100, corresponding to a service rate of 0. As it begins to accumulate service units (see definition of service), its service rate begins to increase, causing a corresponding decline in its workload level. Now the second address space, Adsp2(obj1), becomes ready and is assigned a workload level of 100, causing Adsp1(obj1) to be swapped out and Adsp2(obj1) to be swapped in. While the workload level of Adsp2(obj1) is now declining, the workload level of Adsp1(obj1) begins to increase again, since its time interval (see definition of service rate) continues to increase while it is swapped out and not accumulating service units. The result is another exchange swap.

This exchange between the two address spaces continues until one of them completes processing. The frequency of the exchange depends on the rate at which the respective workload levels exceed one another and also, of course, on the frequency of the SRM's monitoring of the situation (constant RMPTTOM, Section 6).

Let us now add an address space associated with objective 2 to this domain, Adsp3(obj2). Because of the steeper slope of objective 2, the workload level of Adsp3(obj2) decreases at a slower rate than the workload levels of Adsp1(obj1) and Adsp2(obj1), for corresponding increases in the service rate. The effect is to make Adsp3(obj2) more "important", since it will be kept in real storage for longer intervals than its rivals, assuming, of course, a similar rate of consumption of service units, and ignoring reasons for swaps other than the workload level (long waits, etc.)

If an address space associated with objective 3, Adsp4(obj3), is included in this domain, it can readily be seen that this address space will not be swapped out at all until it has completed processing, since its workload level is always greater than or equal to the workload levels of the other three address spaces (again ignoring other reasons for swaps).

Figure 3-4 below illustrates these points and provides a basis for further observations.

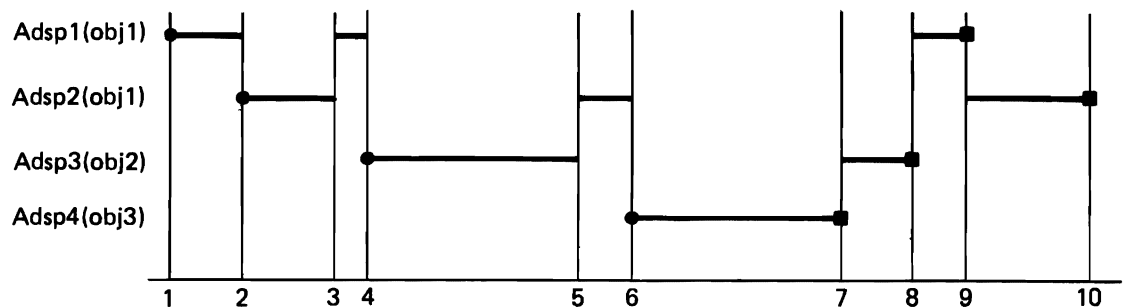


Figure 3-4. Competition for Resources Controlled by Performance Objectives

The details depicted in Figure 3-4 are arbitrary and depend entirely on such variables as the arrival times (that is, when address spaces become ready) and the amount of service required by each address space to complete processing. The concepts involved, however, will be the same.

The numbers denote the following events:

1. Adsp1(obj1) becomes ready and is swapped in with a workload level of 100.
2. Adsp2(obj1) becomes ready and is assigned a workload level of 100. Since the workload level of Adsp1(obj1) is now less than 99, Adsp1(obj1) is swapped out and Adsp2(obj1) is swapped in.
3. The workload level of Adsp2(obj1) has now declined to a level below that of Adsp1(obj1), which has increased while in the swapped out state. Therefore, Adsp1(obj1) preempts Adsp2(obj1).
4. Adsp3(obj2) becomes ready and preempts Adsp1(obj1) with a workload level of 100. Since the slope of objective 2 is steeper than that of objective 1, Adsp3(obj2) will stay swapped in for a longer interval.
5. The workload level of Adsp3(obj2) has fallen below that of Adsp2(obj1), causing it to be preempted by the latter.
6. Adsp4(obj3) becomes ready and preempts Adsp2(obj1) with a workload level of 150. It will remain in real storage until it completes because its workload level cannot fall below 100.
7. Adsp4(obj3) completes processing, allowing Adsp3(obj2) to be swapped in again, assuming that it now has the highest workload level.
8. Adsp3(obj2) completes and Adsp1(obj1) is swapped in, assuming that it has the higher workload level.
9. Adsp1(obj1) completes processing, allowing Adsp2(obj1) to be swapped in.
10. Adsp2(obj1) completes processing.

Figure 3-4 allows us to make the following observations:

- Address spaces associated with objective 3 can have a serious impact on the performance of address spaces associated with objective 1 and objective 2. This is especially true if the former represent long-running, batch-like work and the latter represent short, interactive TSO commands that require fast response times.
- If the reverse is true, that is, objective 3 is used for short TSO commands, and these arrive at a fairly regular rate, then the turnaround time for batch work may be very unpredictable.
- If batch-like work is associated with the same performance objective as short, interactive work, and objective 3 is not used, the long-running work may still be severely impacted since regularly arriving short work will constantly preempt it with a workload level of 100.



- Under the rules of competition described so far, a great deal of swapping takes place, causing a corresponding amount of overhead. This would be even more apparent if all address spaces were associated with either objective 1 or objective 2. Notice, however, that if objective 3 were the only objective used in this domain, swapping would be considerably reduced, due to the fact that much greater changes in the service rates would be required to effect similar changes in the workload levels. In other words, the frequency of swapping depends directly on the slope of the performance objective – the steeper the slope, the less frequent the incidence of swapping.

Note also that the competition for access to real storage is reduced with each increase in the target MPL of the domain, assuming a fixed maximum number of ready address spaces, and is eliminated altogether if the target MPL is equal to this maximum number.

It is often desirable to exercise a greater degree of control over the competition within a domain than is afforded by the use of performance objectives alone. Such additional control is provided through another IPS parameter, the *interval service value*.

#### **Interval Service Value (ISV)**

The purpose of the ISV is to enable an installation to control the frequency of swapping.

This is accomplished by superseding the previously discussed rules for competition within a domain for a specific interval, as follows: each time an address space is swapped in, it is assigned a service rate of zero, in effect assigning to that address space the highest possible workload level for the respective performance objective, that is, the cut-off workload level. The ISV specifies the number of service units an address space is allowed to accumulate before the workload level corresponding to the actual service rate applies.

To illustrate the use of the ISV, consider again a domain with address spaces Adsp1(obj1), Adsp2(obj1), Adsp3(obj2) and Adsp4(obj3). Assume that each address space is associated with an ISV of 400 service units, and that the number of service units required by each address space to complete is as follows:

Adsp1(obj1) – 400 service units  
 Adsp2(obj1) – 600 service units  
 Adsp3(obj2) – 800 service units  
 Adsp4(obj3) – 600 service units

Figure 3-5 illustrates the effect of the ISV on the competition between these address spaces.

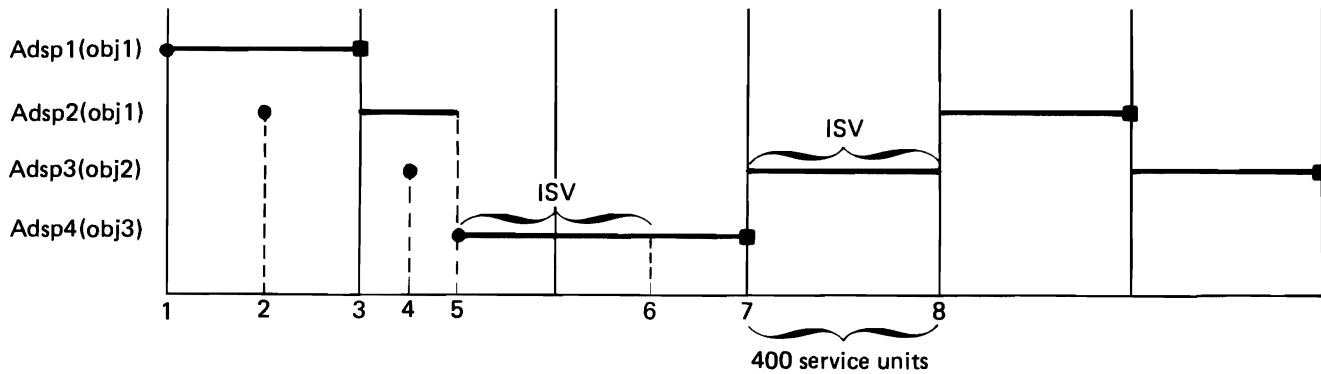


Figure 3-5. Competition for Resources Controlled by Performance Objectives and ISVs

The numbers denote the following events:

1. Adsp1(obj1) becomes ready and is swapped in. It remains in real storage for the length of its ISV and completes processing without being swapped.
2. Adsp2(obj1) becomes ready but does not preempt Adsp1(obj1) since the latter is in its ISV, that is, the two workload levels are equal.
3. Adsp1(obj1) completes and Adsp2(obj1) is swapped in.
4. Adsp3(obj2) becomes ready but cannot preempt Adsp2(obj1), since the latter is in its ISV. Notice that the only significant property of an objective, while the address space is in its ISV, is its cut-off workload level. The slope of the objective has no meaning during this interval.
5. Adsp4(obj3) becomes ready and preempts Adsp2(obj1), even though the latter is still in its ISV, due to the fact that objective 3 has a cut-off level of 150. The ISV, therefore, represents a guaranteed interval of service only when the cut-off levels of all objectives in the domain are equal.
6. Adsp4(obj3) comes out of its ISV but remains in storage since its workload level cannot decrease to less than 100.
7. Adsp4(obj3) completes processing and Adsp3(obj2) is swapped in. Notice that the service rate is computed in the usual manner while an address space is in the swapped out state. Since Adsp2(obj1) has already accumulated service units, its workload level is lower than that of Adsp3(obj2).
8. Adsp3(obj2) leaves its ISV and is preempted by Adsp2(obj1).

Figure 3-5 allows us to make the following observations:

- The frequency of swapping can be reduced by using the interval service value. Notice that Figure 3-5 shows fewer swapped in intervals than Figure 3-4.
- The ISV is most effective when the cut-off workload levels of all performance objectives in a domain are equal. Notice that Adsp4(obj3) causes Adsp2(obj1) to be swapped out while the latter is still in its ISV. The reason is the higher cut-off level of objective 3, that is, 150 versus 100.

- The effectiveness of the ISV is increased when its value is increased. For example, consider the same domain without Adsp4(obj3) and assume an ISV of 800. In this case, each of the three address spaces would complete processing without exchange swaps.

The use of the ISV will be discussed again later in this section and also in Section 3.

Up to this point we have discussed three performance characteristics that can be associated with an address space: the domain, the performance objective and the ISV. A fourth factor has already been discussed in Section 1 – the dispatching priority. Once an address space has been swapped into real storage, it competes with all other address spaces, regardless of domain, on the basis of the dispatching priorities. The higher the dispatching priority, the higher the rate at which an address space is allowed to consume resources. Thus, if an installation wishes to favor one type of work over another, it can assign a high dispatching priority in addition to favorable domain constraints, performance objectives and interval service values.

#### **Response – Throughput Bias (RTB)**

A swap recommendation based on the workload level of an address space is one of three possible factors contributing to the SRM's exchange swap decisions. The other two are recommendations from the CPU and I/O load adjusting functions (see "Functions", Section 1).

An installation can specify whether the latter two recommendations should be considered or not via the RTB parameter in the IPS. In addition, these load adjusting recommendations can be given a desired weight, or importance, via the resource factor coefficients in the OPT (see "OPT Concepts" in this section).

The RTB has only two possible values: zero and one. If the RTB is set to zero, the workload level recommendation is the sole basis for exchange swap decisions. The effect of this choice is to emphasize good response time as opposed to system throughput considerations, since system load adjusting recommendations will be ignored.

If the RTB is set to one, the CPU and I/O load adjusting recommendations are factored into the exchange swap decisions (see "OPT Concepts"). Since these recommendations can be weighted, an installation may specify, for instance, that under certain conditions an address space may "hog" the CPU, even though this may result in a degradation of response time for all other address spaces. In effect, setting RTB to one means that swap recommendations should be made in the interest of throughput, not response time.

### Performance Period

In the preceding examples, each address space has been associated with one set of performance characteristics from the time it becomes ready until processing is completed. Additional flexibility of control is gained by dividing this life span into distinct *performance periods*, and associating an address space with a different set of performance characteristics for the duration of each period. This ability is provided via the IPS.

The purpose of performance periods is to allow an installation to vary the performance characteristics of transactions as their execution characteristics change. For example, if a domain includes a variety of TSO users, transactions with greatly differing life spans will be competing with one another. If the majority of transactions are of short duration and these are to experience consistently good response time, it may not be satisfactory to keep one set of performance characteristics in effect for the entire life span of every transaction. By using performance periods, short transactions, for instance, can be favored over long transactions without prior knowledge of individual life spans. The following example illustrates this concept.

Consider a TSO domain with a variety of transactions whose life spans range from short to intermediate to long, where:

short is  $\leq 400$  service units,  
 intermediate is  $> 400$  and  $\leq 2000$  service units,  
 long is  $> 2000$  service units.

Assume that three sets of performance characteristics have been defined, as follows:

Set 1: TSO domain, objective 3, ISV=400 service units, dispatching priority = 7E, RTB=0

Set 2: TSO domain, objective 2, ISV=2000 service units, dispatching priority = 7D, RTB=0

Set 3: TSO domain, objective 1, ISV=2000 service units, dispatching priority = 76, RTB=0

Assume also that three performance periods have been defined, such that Set 1 is in effect during Period 1, Set 2 is in effect during Period 2, and Set 3 is in effect during Period 3, for all transactions in the TSO domain, and that the length of each period is as follows:

Period 1: 400 service units  
 Period 2: 1600 service units  
 Period 3: to end of transaction

Each transaction, therefore, is associated with the following composite set of characteristics:

- Set 1 for the first 400 service units, or to end of transaction
- Set 2 for the next 1600 service units, or to end of transaction
- Set 3 to end of transaction.

Such a composite set of performance characteristics, consisting of one or more performance periods, is called a *performance group*. A transaction is associated with a performance group via the latter's unique identifier, the *performance group number*.

### Performance Group

The performance group is the vehicle that enables an installation to associate a user's transaction(s) with a set of performance characteristics for each point in the life of the transaction(s), and thus to specify the treatment it wishes a job, job step, or time-sharing session to receive at all times. Batch job steps and TSO terminal sessions are associated with a performance group by the specification of a performance group number (PERFORM=nnn) on the JOB or EXEC statement, on the LOGON command, on the RESET command, or by default. (See "Assigning Performance Group Numbers" in Section 4.)

Let us now collect the specifications used so far in this section into a complete IPS, using the official syntax as described in Section 5:

```

CPU=10.0,IOC=5.0,MSO=3.0
WKL=(1,50,100)
OBJ=1,SRV=(1000,500,0)
OBJ=2,SRV=(2000,1000,0)
OBJ=3,SRV=(2000,2000,2000)
DMN=1,CNSTR=(1,2,1)
DMN=2,CNSTR=(4,8,10)
PGN=1,(DMN=1,OBJ=1,ISV=400, . . .)
PGN=2,(DMN=1,OBJ=2,ISV=400, . . .)
PGN=3,(DMN=1,OBJ=3,ISV=400, . . .)
PGN=4,(DMN=2,OBJ=3,ISV=400,APG=14,RTB=0,DUR=400)
      (DMN=2,OBJ=2,ISV=2000,APG=13,RTB=0,DUR=1600)
      (DMN=2,OBJ=1,ISV=2000,APG=6,RTB=0)

```

where:

Domain 1 (DMN1) is the domain used for the examples illustrated by Figure 3-4 and Figure 3-5;

Domain 2 is the TSO domain described under "Performance Period";

Performance groups 1, 2 and 3 (PGN1, . . .) represent the partial sets of characteristics illustrated by Figure 3-5;

Performance group 4 is the composite set described under "Performance Period".

We now return to the discussion of the TSO domain and performance group PGN4. The following observations can be made about transactions competing in this domain:

- During Period 1, all transactions are associated with performance objective OBJ3, which represents relatively “good” service (see Figure 3-3), and a high APG priority of 7E (70 + E = 7E (HEX)). An ISV of 400 ensures that no exchange swapping takes place at all during this period, since this value is equal to the duration of the period. Note again that each specification is in effect only for the duration of the respective period.
- Transactions requiring more than 400 service units are then switched to the less favorable performance objective OBJ2 (a lower cut-off level), and a lower dispatching priority of 7D.
- Transactions requiring more than 2000 service units are then switched to objective OBJ1, and the mean-time-to-wait APG priority (see “Functions”, Section 1).
- Under these conditions, intermediate and long commands (transactions completing during Period 2 and Period 3, respectively), will probably exhibit poor response times during periods of high terminal activity. Notice that transactions in Period 1 will preempt those in Period 2 and Period 3 even when the latter are in their ISV interval (see Figure 3-5). To prevent swapping during ISV intervals, the cut-off workload levels of the performance objectives must be equal, as in the following performance group (PGN 4 without objective OBJ3):

```
PGN=5, (DMN=2,OBJ=2, . . . .)
      (DMN=2,OBJ=2, . . . .)
      (DMN=2,OBJ=1, . . . .)
```

Under the conditions of performance group PGN5, the response time of short commands may become erratic since transactions in Period 2 and Period 3 now have a much higher chance of gaining access to real storage. This problem may be alleviated somewhat by decreasing the ISV for Period 3 to 1000 service units. Another solution would be to remove transactions entering Period 3 from the domain, as follows:

```
PGN=6, (DMN=2, . . . .)
      (DMN=2, . . . .)
      (DMN=1, . . . .)
```

It should be remembered that the purpose of the examples in this section is to illustrate concepts, not to serve as recommendations.

This completes the discussion of IPS concepts.

### OPT Concepts

The parameters discussed in this part are specified in the IEAOPTxx member of SYS1.PARMLIB.

#### Resource Factor Coefficients (RFC)

The purpose of the resource factor coefficients is to enable an installation to assign the desired weight to the swap recommendation values produced by the CPU and I/O load adjusting functions (See "Resource Use Functions", Section 1). If the response-throughput bias (RTB) in the IPS is set to one, the SRM will combine these weighted values with the workload level recommendation to determine the full swap recommendation value of an address space. This combined value is the basis for exchange swap decisions. It is computed by the following formula:

$$\begin{aligned} \text{swap recommendation value} = & \\ & \text{workload level recommendation} \\ & + (\text{CPU coefficient} * \text{CPU load adjusting recommendation}) \\ & + (\text{I/O coefficient} * \text{I/O load adjusting recommendation}) \end{aligned}$$

Setting either coefficient to zero eliminates that recommendation from consideration. Setting both coefficients to zero is equivalent to a specification of RTB=0 in the IPS.

The values generated by the load adjusting functions depend on the total CPU and I/O imbalance and on the percentage of that imbalance caused by a particular address space. Although the computations involved are beyond the scope of this discussion, the effects of specific coefficients on swap analysis can be studied by considering the maximum possible load adjusting values. These values are limited, by definition, to 1/5 of the highest cut-off workload level of all performance objectives.

For example, if the highest cut-off level is 100, the maximum CPU load adjusting recommendation value will be  $(1/5 * 100) = 20$ .

Consider an IPS with WKL=(1,100) and a performance objective with SRV=(500,0). The following graph illustrates the range of variations in the swap recommendation value due to a CPU coefficient of 1.

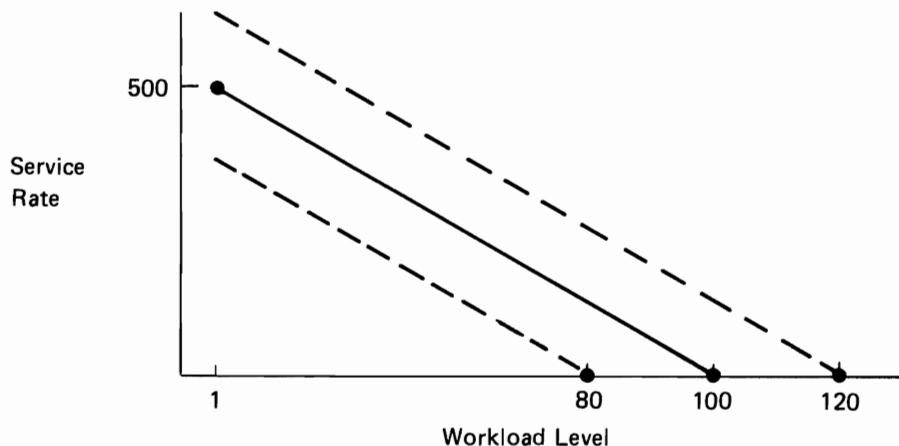


Figure 3-6. The Effect of RFCs on Swap Recommendation Values.

Figure 3-6 allows the following observations:

- An increase in the resource factor coefficients causes a corresponding decrease in the effect of workload level recommendations on exchange swap decisions.
- High resource factor coefficients can lead to a loss of control over the frequency of swapping, achieved via the interval service value (See Figure 3-5). That is, the differences in the swap recommendation values of address spaces will be such that the frequency of exchange swaps could approach a level equivalent to no ISV control.

These two observations indicate that care must be taken in selecting the coefficients, and that any positive effect achieved through their use should be balanced against the possible negative impact on workload level control.

- The SRM assigns either a positive or negative sign to the load adjusting recommendation values, depending on the particular imbalance. For instance, if the CPU is underutilized, it will select ready address spaces that are heavy CPU users and add the load adjusting value to their workload levels to increase their chances of getting swapped in. If the CPU is overutilized, the SRM will select swapped in address spaces that are heavy CPU users and subtract the load adjusting value from their workload levels to increase their chances of being swapped out.

#### **Resources Manager Constant (RMC)**

The purpose of the Enqueue Residence Value (ERV) is discussed under “Enqueue Delay Minimization”, Section 1.

### **Section 3: Guidelines and Examples**

This section discusses guidelines for:

- Defining installation requirements and objectives
- Selecting values for IPS and OPT parameters
- Evaluating and adjusting these values.

In addition, the default IPS and several IPS variations are presented and explained.

An installation’s individual needs and requirements may lead to ultimate specifications that differ greatly from the values presented here. The guidelines should, nevertheless, prove useful as a starting point.



## Defining Installation Requirements

Before specifying any parameters to the SRM, an installation must define response and throughput requirements for its various classifications of work.

Examples of specific questions that should be answered are listed below. The applicability of these questions will, of course, vary from installation to installation.

### Subsystems

*How many subsystems will be active at any one time and what are they ?*

*For IMS or VSPC, how many active regions will there be ?*

*Will the subsystem address space(s) be nonswappable ?*

*What is the desired response time and how will it be measured ?*

### Batch

*What is the required batch throughput or turnaround for various job classes ?*

*Are jobs being placed in the correct class ?*

This is an important question. The MVT scheme of job classing may not be appropriate in MVS because of differences in storage usage.

*How much service do these jobs require, and what service rate is needed to meet the turnaround requirement ?*

The MF/1 workload report or reduction of SMF data in type 5 records will provide the average service consumed by jobs of different classes. For installations new to MVS, the following suggestion is offered based on service definition coefficients of CPU=10.0, IOC=5.0, MSO=3.0:

Quick jobs use 10,000 service units or less.

Medium jobs use between 10,000 and 30,000 service units.

Long jobs use 30,000 units or more.

*What is the average number of ready jobs ?*

Most likely, this is the number of active initiators. In MVS, the number of initiators in real storage may be less than the number of MVT/SVS initiators. A few extra initiators may be started to decrease turnaround times.

### TSO

*What is the number of terminals ?*

*What is the average number of ready users ?*

As a guideline for installations new to TSO, assume two ready users for every ten users logged on. This average will depend on the type of terminal and on the type of TSO session (data entry, problem solving, program development).

*What is the required response time and expected transaction rate for different categories of TSO transactions at different times, such as peak hours ?*

*What is the expected response time for short transactions ?*

*How will this response time be measured ?*

*Should response time be different for select groups of TSO users ?*

*How should semi-trivial and non-trivial transactions be treated ?*

*How are they defined ?*

An installation using MVS will have MF/1 Workload Reports or SMF data in type 34 and 35 records available to help define trivial and non-trivial TSO work. For those installations with no previous MVS experience, the following suggestions may serve as guidelines, again based on service definition coefficients of CPU=10.0, IOC=5.0, MSO=3.0:

Short TSO commands use 600 service units or less.

Medium length commands use between 600 and 2,000 service units.

Long TSO commands use 2,000 service units or more.

*What is the required service rate for TSO users ?*

If 2-second response time (as reported by MF/1) is required for very short TSO commands (100 service units), the required service rate for such a transaction is 100/2 or 50 service units per second. Service rates for other types of transactions should be computed also.

### **General**

*What is the importance level of TSO, batch, IMS, and special batch classes in relation to one another ?*

*Which may be delayed or "tuned down" to satisfy other requirements ?*

In other words, which response requirements are fixed and which are variable ?

*What percentage of system resources should each group receive ?*

*How critical are certain resources – especially the CPU and channels ?*

If a user makes heavy demands on a resource, the installation will have to decide whether the user's response time requirement should be waived in favor of a more balanced system load.

Once these questions have been answered, and the installation is somewhat confident that its requirements can be met by the hardware, the installation is ready to begin writing an initial IPS and OPT.

## Preparing an Initial IPS and OPT

This section presents guidelines for selecting IPS and OPT parameter values. Where applicable, values are given which may be used as initial values when no previous performance data is available.

There are three approaches to preparing an initial IPS and OPT:

1. Use the default IPS and OPT
2. Modify the defaults
3. Create a new IPS and/or OPT.

It is suggested that installations new to MVS initially use the default IPS (IEAIPS00) provided with the system. This also applies to installations using the default IPS provided with Release 2 and 3 of OS/VS2. (See "Default IPS" in this section.)

Installations that have modified the default IPS provided with release 2 or 3 of MVS and created new PARMLIB members should consider replacing these, due to different functional characteristics of the new SRM.

**Work Sheets:** If an installation decides to modify the default IPS or create a new IPS, the following worksheets may be of value while reading this section. They have been included as an aid for recording parameters and values as they are defined. After concluding this section, the completed work sheets should enable an installation to write an IPS, in formal syntax, with a minimum amount of effort. (In most cases, only a subset of the available entries will be needed).

Worksheet Number 1 – Service Definition Coefficients

CPU \_\_\_\_\_

IOC \_\_\_\_\_

MSO \_\_\_\_\_

Domain Definitions				
Domain Number	Minimum MPL	Maximum MPL	Weighting Factor	Description

Worksheet Number 2 — Dispatching Priorities

Dispatching Priority		Associated User Types
	Dec (Hex)	
	255 (FF)	
	⋮	
	⋮	
	⋮	
	⋮	
Default APG Range	Fixed	127 (7F)
		126 (7E)
		125 (7D)
	Rotate	124 (7C)
		123 (7B)
		122 (7A)
	Fixed	121 (79)
		120 (78)
	Mean- Time- to- Wait	119 (77)
		118 (76)
	112 (70)	
	⋮	
	⋮	
	0 (00)	

### Worksheet Number 3 – Execution Characteristics Definition

<b>Class</b>	<b>Subclass</b>	<b>Duration</b>	<b>ISV Value</b>	<b>Objective Number</b>	<b>Comments</b>

Worksheet Number 4 – Objective Control Per Domain

Workload Levels \_\_\_\_\_

Maximum Absorption Rate \_\_\_\_\_

Highest Cut-off Workload Level \_\_\_\_\_

Domain Number 1

--

Domain Number 2

--

Domain Number 3

--

Domain Number 4

--

Domain Number 5

--

Domain Number 6

--

Domain Number 7

--

Domain Number 8

--

Domain Number 9

--

Worksheet Number 5 – Performance Group/Period Definition

Performance Group Number	Domain Number	Objective	Dispatching Priority	Interval Service Value	Duration	RTB (Default 0)	UNT (Default S)
1 Batch Default	Period	1				0	S
		2				0	S
		3				0	S
		4				0	S
		5				0	S
		6				0	S
		7				0	S
		8				0	S
2 TSO Default	Period	1				0	S
		2				0	S
		3				0	S
		4				0	S
		5				0	S
		6				0	S
		7				0	S
		8				0	S
	Period	1				0	S
		2				0	S
		3				0	S
		4				0	S
		5				0	S
		6				0	S
		7				0	S
		8				0	S
	Period	1				0	S
		2				0	S
		3				0	S
		4				0	S
		5				0	S
		6				0	S
		7				0	S
		8				0	S



### Selecting Service Definition Coefficients (Worksheet Number 1)

The purpose of the service definition coefficients (SDCs) is to allow the individual service components to be weighted. For example, there will probably be a greater demand for the resource in least supply. Using the coefficients, service provided by such a resource can be given added importance. This is not to imply that the service values are to be used for accounting purposes, since, as will be seen later, the service consumed by a job is not necessarily repeatable.

The coefficients should be high enough to yield a range of service rates sufficiently high for the workload management function to be effective, but they must not be so high that service rates become excessively large. This also results in ineffective workload management control.

Changes to the coefficients may necessitate changes to other parameters in the IPS that are dependent on service value specifications (for example, interval service value and duration).

An increase in a service definition coefficient will numerically raise the system service capacity and the service rate of users, though not, of course, affecting the system's physical capacity for work.

For example, consider the I/O component of service for the following:

- With IOC=1.0, a service rate of 100 represents 100 I/O requests/second.
- With IOC=2.0, a service rate of 100 represents 50 I/O requests/second.

Three alternatives exist for setting the SDCs:

1. Use the values in the default IPS (IEAIPSO0). These should generate reasonable service values for the workload management function.

The default coefficients are:

CPU=10.0  
IOC=5.0  
MSO=3.0

2. Use previous installation SDCs. The installation may wish to modify the coefficient for main storage usage. A coefficient of 1 to 3 should have a minimum effect on IPS parameters that are dependent on service value specifications.
3. Define new coefficients. To do this, a more detailed understanding of the individual service components is required. These are discussed separately in the following paragraphs.

**CPU Service:** The SRM calculates CPU service based on task execution time and CPU model, which should make the IPS independent of the CPU model.

The following table describes the CPU service consumed per second of task execution time by CPU model. The values listed are internal SRM constants. The “total system absorption rate” reported by RMF will not equal the values listed here because these do not include SRB-mode processing and certain other types of system processing.

CPU Model	Service Units Per Second of Task Execution Time	Seconds of Task Execution Time Per Service Unit
145	24.0	.0417
155	42.0	.0238
158	51.2	.0195
165	125.0	.0080
168	151.0	.0066

For installations with no prior service data the task time reported in type 4, 5, 34, and 35 SMF records could be converted to service units using the above table.

CPU time may not be identical for different runs of the same job step. One or more of the following factors may cause small variations in CPU time: CPU architecture (such as storage buffering), cycle stealing with integrated channels, and the amount of queue searching (see the publication *OS/VS2 SPL: System Management Facilities (SMF)*, GC28-0706).

**I/O Service:** One I/O service unit corresponds to one EXCP, as reported by SMF. The EXCP count does not include EXCPs issued by a program in system key. Hence, EXCP counts are not maintained for IMS. Note that the count maintained by the SRM wraps around at 65,536. It is reset at every swap analysis, swap out and job step termination. Nonswappable address spaces may exhibit erroneous counts because of these factors.

For a description of SMF EXCP counts, refer to the publication *OS/VS2 SPL: System Management Facilities (SMF)*, GC28-0706.

**Main Storage Service:** A program uses one storage service unit when it holds 50 pages (200K) for one CPU service unit. The amount of storage service available can be determined by calculating the number of pages available for problem program use (from RMF paging reports) and applying the following formula:

$$\text{storage service units} = \frac{\# \text{ pages} * \# \text{ CPU service units}}{50}$$

This formula has been simplified to allow the presentation of an example (see below). In the actual calculation, the SRM utilizes a continually updated variable called *page seconds*. This variable is the product of task execution time and the number of frames allocated to an address space, and it is updated every time the number of frames changes (due to page faults, stealing). The page seconds accumulated by a transaction are reported in SMF type 4 and 34 records.

The main storage component of service can affect the repeatability of service required by a transaction, since the total storage service available varies with the number of address spaces in real storage at any one time.

***For example:***

**Assume** 100 pages and 10 CPU service units are available for problem program usage.

**Case 1:** 1 address space uses all 100 pages for the entire amount of time (10 CPU service units).

The total amount of main storage service used would be:

$$\frac{100 * 10}{50} = 20 \text{ storage service units}$$

**Case 2:** 2 address spaces share the resources equally. That is, each address space uses 50 pages for 5 CPU service units.

The total amount of main storage service used by each address space would be:

$$\frac{50 * 5}{50} = 5 \text{ storage service units}$$

Since each address space used 5 storage service units, the total number of units used is 10. All pages and CPU services were used, and yet the total is less than in Case 1. Thus, the service rate for an address space decreases as the number of executing address spaces increases.

Once it is known how many CPU, I/O and storage service units are available, the coefficients can be selected to weight each service component. For example, if the intent is to equalize the importance of the service components, the coefficients may be determined by using the equations:

$$\text{CPU} * \text{CPU service units} = \text{IOC} * \text{I/O service units} = \text{MSO} * \text{storage service units}$$

Selecting a reasonable value for one coefficient, such as 10 for CPU, allows calculation of the other two values.

**Defining Domains and Their Constraints (Worksheet Number 1)**

The following suggestions for defining domains are based on separating an installation's work into four general classifications:

- Subsystems
- Batch
- TSO
- Special purpose

Each classification is discussed separately. It should be noted that as execution characteristics change, address spaces may be reclassified. For example, very long (to be defined later) TSO commands may be reclassified as batch work.

Guidelines are presented for selecting appropriate MPL values and weights for each type of domain. (Note that minimum MPL values should not be set too high. This may cause excessive paging overhead.)

**Subsystem Domains:** Domains should be defined for each separate subsystem, such as IMS, CICS, VSPC, etc. Associating a subsystem's address space(s) with a unique domain is useful because it enables RMF to report the service used by that subsystem.

Normally subsystems will be nonswappable due to the undesirability of incurring swapping overhead. Other reasons may also exist for making subsystem address space(s) nonswappable. For example, the IMS 1.1.1 resource serialization technique does not cause an address space holding a serialized resource to be swapped in should some other address space require that resource.

MPL specifications for domains of nonswappable subsystem address spaces have no meaning, since nonswappable address spaces are not counted by the SRM when determining the current MPL for a domain.

MPL specifications for domains of swappable subsystem address spaces should allow instant swap in of these address spaces when they become ready. Therefore, the minimum MPL should be greater than or equal to the number of address spaces associated with the subsystem. The maximum MPL value is not meaningful in this case. Since the MPL adjustment function is not utilized for subsystem domains with these recommended MPL values, the weight of such domains should be set to 1.

**Batch and TSO Domains:** There are crucial differences between the execution characteristics of batch and TSO work, such as the amount of service required to complete processing and the arrival rate of work. Batch jobs usually require considerably more service than most TSO commands and could therefore overload the system if the number of such jobs allowed in real storage were not controlled. Such limits, however, are not desirable for most TSO work. Since this control is achieved via the domain constraints, the two different requirements cannot be satisfied by one domain. Therefore, separate domains should be created for batch and TSO work.

In most cases, one domain should be sufficient for batch work. Performance objectives can be used to achieve the desired degree of control over individual jobs.

If all TSO commands require uniform service, one domain will probably suffice to provide good response time. However, if the TSO work consists of commands with diverse service needs, that is, both short and long-running commands, additional domains may have to be defined in order to satisfy these different requirements.

As discussed in Section 2, the maxMPL value is used to limit the number of address spaces allowed in real storage, while the purpose of the minMPL is to guarantee access to real storage for a fixed minimum number of address spaces.

The minMPL value can be used to implement fixed response requirements. If fast response, for example, is a fixed requirement for short TSO commands, the minMPL value should be set to 1 or 2 for every 10 logged on users. For instance, if 40 users are logged on, a minMPL value of 4 would be a good starting point.

For batch work, if there are no fixed throughput requirements, the SRM should be allowed to determine the target MPL in a range of 0-255. If, however, a specific number of jobs must be processed during a particular period of time, the minMPL should be raised slightly above zero (1 to 3) in order to guarantee access to real storage for some minimum number of batch jobs.

For long-running, batch-like TSO commands, the maxMPL value should be used to limit the number of such commands in real storage. This is advisable since such commands, although batch-like, do not have a predictable arrival rate and may therefore overload the system. A minMPL of 1 and maxMPL of 2 might be appropriate if such control is desired.

Domain weights have meaning only for domains whose target MPL is adjustable, and only when there is more than one such domain. Domains with fixed target MPLs should be given a weight of 1.

Domain weights have the effect of creating an ordered priority list according to which the SRM decides which domain's target MPL should be raised or lowered first. The differences between weight values should be sufficiently large to ensure that the computed contention indexes (see Section 2) actually reflect the installation's intentions. These indexes may be calculated beforehand by using representative values, as suggested by the following method:

For batch, assume that the average number of ready users is equal to the number of initiators and that the target MPL is 1.

For TSO, assume that the average number of ready users is equal to the target MPL, resulting in a contention index equal to the weight.

For example, assuming there are 8 initiators, the contention indexes would be:

TSO contention index = TSO weight

batch contention index = 8 \* batch weight

Therefore, if the TSO domain is to be favored over the batch domain, the weight for the TSO domain must be more than 8 times the batch domain weight.

Notice that the weight has no meaning when the minMPL is sufficiently high to eliminate MPL adjustment, as recommended for TSO domains with fast response requirements. The weight may be used, however, to favor batch work over long-running TSO work if both types are in unique domains.

**Special Purpose Domains:** A separate domain may be created to prevent ready work from executing. For example, the system operator may halt the execution of a job by associating it with a domain having a fixed target MPL of zero, that is, minMPL = maxMPL = 0.

### Selecting Dispatching Priorities (Worksheet Number 2)

The following recommendations are made for the selection of dispatching priorities:

- Address spaces with no fixed requirements, possibly long TSO transactions and long batch jobs, should be assigned to the mean-time-to-wait group of the APG to aid throughput.
- Address spaces that are likely to be CPU-bound should be assigned to the mean-time-to-wait group. When such an address space is executing (that is, looping), no address space with a lower priority can be dispatched. The mean-time-to-wait group allows the SRM to dynamically adjust the dispatching priority.
- The IMS control region should be assigned a high dispatching priority, possibly lower than that of TCAM, but higher than the dispatching priorities of initiators and short TSO transactions.
- The dispatching priorities of IMS message processor regions (MPRs) will depend on whether any work assigned to the APG should be favored over these MPRs. If, for example, short TSO transactions are to be favored over IMS MPRs but long TSO transactions are not, then the IMS MPRs must be assigned to the APG.
- Initiator dispatching priorities should be lower than those of short TSO transactions (first period transactions) and IMS MPRs, but probably above the mean-time-to-wait range in order to ensure fast allocation.
- The Rotate dispatching priority of the APG should be used to prevent one subsystem address space from dominating another.
- Note again that for nonswappable address spaces only the priority specified in the first period of the associated performance group is applicable.

### Defining Durations (Worksheet Number 3)

Durations may be used to subdivide a major class of work into subclasses. For example, transactions may be subdivided into short, medium and long. Examples of possible subdivisions by duration are shown in the following table. The values are those used in the default IPS (IEAIPS00).

(*Note:* Duration may be specified in real time units. However, this is not recommended except in special cases which will be discussed later).

Class	Subclass	Duration (Service Units)
Batch	Short	<10000
	Medium	10000 - 30000
	Long	>30000
TSO	Short	< 600
	Medium	600 - 2000
	Long	>2000

A subclass, in these examples, corresponds to a performance period, with the duration value defining the length of the period. The service units required by TSO sessions and batch jobs are reported in SMF type 4, 5, 34, and 35 records. An installation may use this data to define appropriate subclasses. For batch jobs, duration values may be selected to reflect initiator classes.

### Selecting ISV Values (Worksheet Number 3)

As discussed in Section 2, ISV values are used to control the frequency of swapping. An increase in ISV values reduces the number of exchange swaps. The effectiveness of ISV control over swapping is reduced if multiple performance objectives are used in a domain and these have different cut-off levels.

When duration specifications exist, ISV values for the respective periods will depend directly on the duration values. For example, if short TSO transactions have a duration of 600 service units and control swaps are to be prevented during this interval, the ISV value must be at least 600 service units. If an ISV value of 20,000 were associated with short batch transactions (see table), no swaps would occur during that period, since the transaction would switch periods after 10,000 service units. If the ISV value were set to 5000, one control swap would probably occur after 5000 service units, provided, of course, that other address spaces of the domain are ready to be swapped in.

ISV values are not meaningful for non-swappable address spaces and for domains whose minMPL is sufficiently high to allow all its ready address spaces to be swapped in at one time.

### Selecting Performance Objectives (Worksheet Number 4)

It is recommended that new performance objectives be defined. Previously defined objectives may no longer serve their intended purpose due to the functional differences of the new SRM.

To draw a performance objective, workload level numbers and service rates must be selected. For recommended workload level numbers, refer to "Default IPS" and "IPS Examples."

An appropriate set of service rates should take into consideration the total available system service as described under "Selecting Service Definition Coefficients." For example, assume a model 158 4-megabyte system with 300 EXCPs/second and 640 pages for problem program use. Using the default SDCs of CPU=10.0, IOC=5.0, MSO=3.0, the maximum service available would be:

$$(10.0*51.2) + (5.0*300) + (3.0*640*51.2/50) = 3978$$

Since only a fraction of this total is available to any single address space, a service rate of 1000 to 2000 would be a reasonable upper limit for performance objectives for this configuration.

Adjustments may be necessary after analysis of RMF workload reports showing the absorption rates of individual performance objectives. If the absorption rate of an objective approaches the maximum specified service rate, that objective should be redefined with a higher maximum service rate.

The following points should be considered when defining performance objectives:

- Should the cut-off workload level be equal to that of other objectives of the same domain, or should address spaces be preempted while in their ISV interval? For example, especially important batch jobs may be associated with an objective that allows them to preempt all other jobs in the domain.
- The slope of an objective is not meaningful when all transactions in a domain complete processing within their ISV interval.
- Drawing a steeper slope will cause recommendation values for swapped in address spaces not in their ISV to decrease less rapidly, thereby allowing more service to be accumulated prior to swapping. The steeper the slope of the objective, the less swapping will be incurred.
- Objectives should be kept simple.



For batch work, two objectives may provide the initial desired degree of control, while one objective may be sufficient for TSO work. Suggested variations for both types are described under "IPS Examples". One possible variation of the use of objectives is illustrated below.

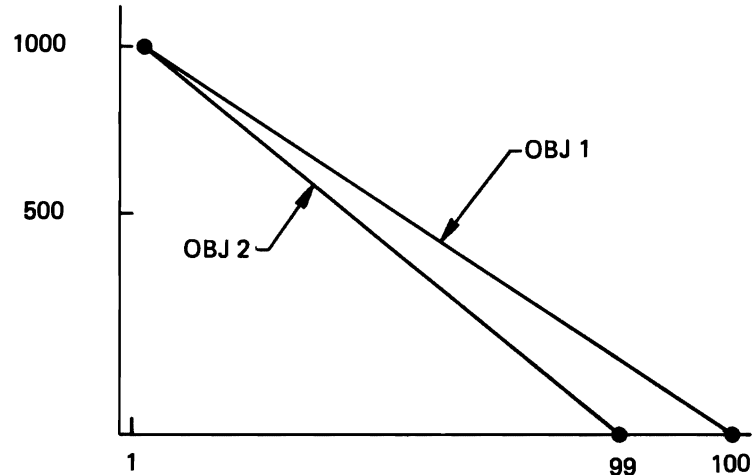


Figure 3-7. A Use of Different Cut-off Levels

**Explanation:**

Workload levels must differ by more than 1 to cause an exchange swap. Therefore, address spaces associated with the lower objective will not be preempted by address spaces associated with the higher objective while in their ISV intervals. That is, ISV preemption will not occur even though the cut-off levels of the two objectives are different.

If, however, the MPL adjustment function were to decrease the target MPL of this domain, the lower cut-off level of objective 2 would cause an address space associated with that objective to be swapped out. If the cut-off levels were equal, an address space associated with objective 1 would have the same chance to be selected for swap out.

This combination of objectives, therefore, accomplishes a dual purpose – ISV preemption is avoided, and the intended function of the higher objective is preserved.

**The RTB Parameter and the Resource Factor Coefficients**

It is recommended that the RTB be initially set to 0. Possible cases when the RTB should be set to 1 are discussed under "Evaluating and Adjusting the IPS and OPT".

The RTB function is useful only when CPU or logical channel utilizations are significantly out of balance. Under normal circumstances, an RTB specification of 1 will cause unnecessary overhead without accomplishing anything.

Also, as discussed in Section 2, the CPU and I/O load adjusting recommendation values are added to workload level recommendations. The resultant differences in swap recommendation values may be large enough to cause a loss of control over the frequency of swapping, that is, ISV control of swapping may no longer be effective.

### **Performance Groups (Worksheet Number 5)**

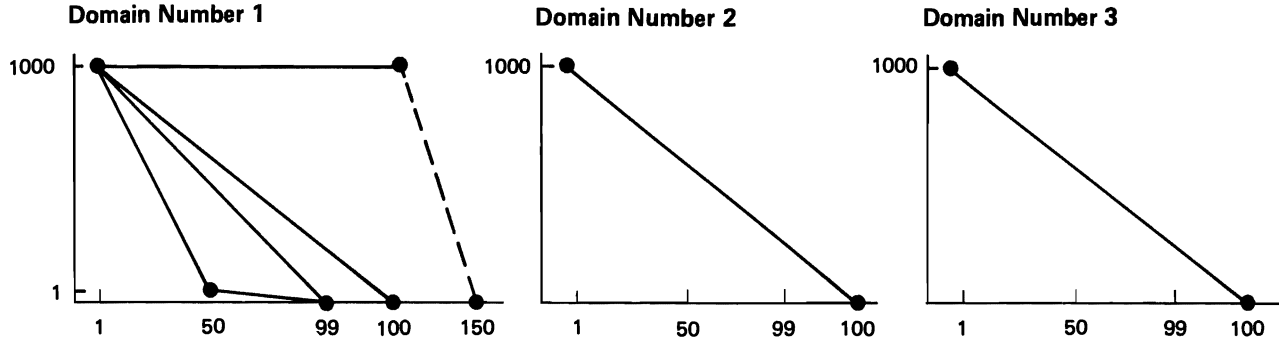
Reasons for assigning unique performance group numbers vary from installation to installation. The following are possible candidates for unique numbers:

- Each subsystem
- Individual subsystem address spaces
- Groups of users
- Departments
- Batch initiator classes
- Monitor programs

For batch work, the default performance group number is 1. For TSO, the default number is 2. These two performance groups must be specified in every IPS.

### **Default IPS**

The default IPS is based on the guidelines just presented. However, not all specific recommendations were used because this IPS must be acceptable for any CPU, storage, and I/O configuration. The philosophy of this default IPS is to provide highest priority (best response) to short TSO commands, then medium TSO commands, then long TSO commands, and to detain batch jobs whenever TSO service needs to be increased.



```

CPU=10.0, IOC=5.0, MSO=3.0          /* DEFAULT IPS - IEAIPS00 */
WKL=(1,50,99,100)

OBJ=1,SRV=(1000,*,*,0)              /* FIRST PERIOD BATCH (DMN1) */
OBJ=2,SRV=(1000,*,0)                /* SECOND PERIOD BATCH (DMN1) */
OBJ=3,SRV=(1000,*,*,0)              /* FIRST PERIOD TSO (DMN2) */
OBJ=4,SRV=(1000,*,*,0)              /* SECOND PERIOD TSO (DMN2) */
OBJ=5,SRV=(1000,*,*,0)              /* THIRD PERIOD TSO (DMN3) */
OBJ=6,SRV=(1000,1000,1000,1000)    /* HOT BATCH (DMN1) */
OBJ=7,SRV=(1000,1,0)                /* LOW PRIORITY BATCH (DMN1) */

DMN=1,CNSTR=(1,50,1)                /* BATCH */
DMN=2,CNSTR=(1,50,255)              /* SHORT AND MEDIUM TSO */
DMN=3,CNSTR=(1,50,16)               /* LONG TSO */

PGN=1, (DMN=1,APG=6,DUR=10K,ISV=10K,OBJ=1,RTB=0) /* BATCH -SHORT */
      (DMN=1,APG=6,ISV=10K,OBJ=2,RTB=0)          /* ----- -MEDIUM */

PGN=2, (DMN=2,APG=14,DUR=600,ISV=600,OBJ=3,RTB=0) /* TSO -SHORT */
      (DMN=2,APG=13,DUR=1400,ISV=2K,OBJ=4,RTB=0) /* ----- -MEDIUM */
      (DMN=3,APG=6,ISV=10K,OBJ=5,RTB=0)          /* ----- -LONG */

PGN=3, (DMN=1,APG=9,ISV=100K,OBJ=6,RTB=0)        /* HOT BATCH */

PGN=4, (DMN=1,APG=6,ISV=10K,OBJ=7,RRB=0)         /* LOW PRYTY BATCH */
    
```

Figure 3-8. The Default IPS

**Notes on the default IPS:**

The multiple identical objectives (1, 3, 4 and 5) are intended solely as a means to provide more detailed RMF workload reports.

Objective slopes are defined such that address spaces leaving their ISV will have recommendation values of at least 1 less than the cut-off level.

Jobs associated with performance group 3 will in effect run nonswappable due to the characteristics of performance objective 6.

## IPS Examples

### Example 1

Assume:

- A 168 TSO/Batch system with 3 megabytes of real storage.
- On the average, 30 users are logged on doing program development work, with compiles and test execution. Short commands have top priority. The compiles and tests may be deferred.
- Batch class X work has a 1-hour turnaround requirement. All other batch work is class A and has a 24-hour turnaround requirement.
- Five initiators are started to class XA.

```

CPU=10.0, IOC=8.0, MSO=1.1          /* IPS Example 1          */
WKL=(1,25,100)
OBJ=1,SRV=(2000,*,0)
OBJ=2,SRV=(1000,*,0)                /* Normal Batch          */
DMN=1,CNSTR=(1,10,5)                /* Batch-like Domain     */
DMN=2,CNSTR=(3,12,10)               /* Short TSO Domain      */
DMN=3,CNSTR=(2,2,1)                 /* Long TSO Domain       */
DMN=4,CNSTR=(3,3,1)                 /* Domain For Monitors   */

PGN=1,(OBJ=2,DMN=1,ISV=20K)         /* Batch, except class X */

PGN=2, (OBJ=1,DMN=2,DUR=800,ISV=2K,APG=14) /* TSO - short          */
      (OBJ=1,DMN=2,DUR=1200,ISV=2K,APG=13) /* TSO - medium         */
      (OBJ=1,DMN=3,DUR=10K,ISV=10K,APG=11) /* TSO - long           */
      (OBJ=1,DMN=1,ISV=20K)           /* Batch - like TSO     */

PGN=3, (OBJ=1,DMN=1,ISV=20K,APG=10)     /* Batch class X        */

PGN=4, (OBJ=1,DMN=4,APG=15)             /* Monitor programs     */

```

#### Notes on Example 1:

The values for the service definition coefficients were determined by starting with CPU=10 and then using the definition of the three service components to equalize their importance. For example, the value of MSO was calculated by using the equation:

$$10 * 151 = \frac{MSO * 460 * 151}{50}$$

The value of IOC was determined in the same manner.

TSO work is favored over batch work via higher dispatching priorities until it exceeds 12K service units, after which it is assigned a lower dispatching priority than class X batch work.

Two monitoring programs (for example, RMF and a storage monitor routine) are assigned to a unique performance group (PGN4) and domain (DMN4) to obtain RMF reports showing the amount of service they use. These programs are assured immediate swap-in (minMPL = 3) and in addition are assigned a high dispatching priority.

- 2) Decrease the target MPL for the short TSO transaction domain by lowering the maximum MPL. This should produce a steady number of swapped out TSO transactions, thus providing a more consistent delay for all transactions and the elimination of erratic response time. It may, of course, also mean an overall increase in response time and a lower transaction rate for all TSO transactions.
- 3) Create a staging queue for TSO transactions. In the TSO performance group, add a new period, possibly preceding the current first period. Associate this period with a domain having a minimum and maximum MPL of 0, and assign a real time duration of 1 or 2 seconds to the period. This provides a real time lag to all commands, thus lowering the transaction rate and eliminating the contention that has caused erratic response time. If small, this delay may not even be noticed at the terminal.

**Problem 2: TSO Response Time Is Too Fast**

**Detection:** The objective criteria established when defining installation requirements is the only means of judging this condition. Both SMF (records 34 and 35) and RMF provide service usage and transaction information.

**Possible Causes:**

- a) TSO domains are too heavily favored.  
As TSO address spaces become ready, they are being allowed instant access to system resources. This may be slowed in two ways:
  - 1) Introduce the delay (staging queue) as described in Problem 1.
  - 2) Lower the maximum MPL for the short TSO domain to the average number of ready address spaces as indicated by the RMF report. This eliminates the slight advantage the domain would have received in the instances where the target MPL had previously exceeded the average number of ready users.
  - 3) Lower the weighting factor of the short TSO domain. Contention indexes should be recalculated to ensure the appropriate domain priority scheme.
- b) TSO has too great a dispatching priority advantage. Possibly, long TSO transactions should use the same range of dispatching priorities as batch work. This will help delay long TSO response time, but will not slow down short TSO response. However, placing any address spaces at an equal or higher priority than short TSO transactions may cause erratic response time, and thus is a fairly drastic solution.

**Problem 3: TSO Response Time is Poor**

**Detection:** Once again, only the criteria established while defining installation requirements can identify this problem. SMF (records 34 and 35) and RMF provide service usage and transaction information.

**Possible Causes:**

- a) The maximum MPL for the short TSO transaction domain is too low, and/or the weight is too low.  
If there are always more users ready than the maximum MPL will allow into storage, the time each of them spends waiting to be swapped in may be causing the long response time. If the RMF trace report indicates a consistent supply of ready users in the TSO domain, the maximum MPL, and possibly the weight, should be increased to give TSO better response.
- b) The minimum MPL is too small.  
Under fluctuating load conditions, the MPL adjusting function may not be able to react quickly enough to high bursts of demand for resources, since it does not raise the target MPL to more than 1 above a weighted average of ready users. Hence, the target MPL will remain very stable.  
If this target is too low to allow users immediate access to the CPU, response time may be impacted. This would probably be perceived as erratic response time, as described above, but extreme cases could produce consistently slow response time. Raising the minimum MPL constraint to a large value – that is, close to the maximum number of ready users – will improve the response time.
- c) The dispatching priority scheme may be a problem.  
If short TSO transactions have a dispatching priority lower than other work, response time may be affected. Ensure that at least long TSO transactions are placed at a lower priority than short TSO transactions. Refer to the discussion under Problem 1.
- d) Long-running TSO commands may be interfering with short TSO response.  
Long TSO commands (especially batch-like work) can be detrimental to short TSO response time. Performance group periods should be used to alter domains, objectives, or dispatching priorities as a command ages. The duration values of each period should be examined to ensure that lengthy commands are not monopolizing the TSO domain.
- e) Paging overhead may be impacting TSO response.  
If RMF paging activity reports indicate a high Pageable System Area paging rate compared to the address space paging rate, TSO response may be delayed due to time spent waiting for pages. Frequently referenced modules have a tendency to remain in storage, but moderately referenced routines may be continually paged in and out. Examine the PLPA and consider “fixing” these moderately referenced routines.

**Example 2**

Assume:

- A 168 TSO/batch system with 4 megabytes of real storage.
- On the average, 40 users are logged on doing mainly data entry. Several users execute matrix manipulation and graph display jobs from a terminal.
- Batch class C has a 1-hour turnaround requirement.
- Batch class B has a 2-hour turnaround requirement.
- All other batch jobs are class A with 24-hour turnaround.
- Five initiators are started for class C, three for class B, and none for class A until TSO is shut down.

```

CPU=10.0, IOC=5.0, MSO=0.0                /* IPS Example 2    */
WKL=(1,50)
OBJ=1,SRV=(500,0)
OBJ=2,SRV=(1000,0)
OBJ=3,SRV=(2000,2000)                       /* Batch class A    */
DMN=1,CNSTR=(0,0,1)                         /* A delay domain   */
DMN=2,CNSTR=(4,40,7)                        /* Short TSO        */
DMN=3,CNSTR=(1,2,20)                        /* Long TSO         */
DMN=4,CNSTR=(3,10,2)                        /* Batch            */

PGN=1,(OBJ=1,DMN=4,ISV=5K)                 /* Started Tasks    */

PGN=2,(OBJ=1,DMN=1,DUR=2,UNT=R)            /* Trivial Delay    */
      (OBJ=1,DMN=2,DUR=400,ISV=400,APG=9) /* TSO — short     */
      (OBJ=1,DMN=2,DUR=600,ISV=1000,APG=8) /* TSO — medium    */
      (OBJ=1,DMN=3,ISV=2K)                 /* TSO — long      */

PGN=3,(OBJ=2,DMN=4,ISV=10K)                /* Batch class C    */

PGN=4,(OBJ=1,DMN=4,ISV=10K,DUR=3600,UNT=R) /* Batch class B    */
      (OBJ=2,DMN=4,ISV=10K)

PGN=5,(OBJ=3,DMN=4,ISV=100K)               /* Batch class A    */

```

**Notes on Example 2:**

MaxMPL for the short TSO domain (DMN2) is equal to the average number of logged on users.

MaxMPL for the batch domain (DMN4) is equal to the maximum number of started initiators (5+3+2).

After calculating contention index values, domain weights were assigned to allow long TSO work (DMN3) the greatest chance of having its target MPL raised.

Access to resources is delayed by 2 seconds for all TSO commands (DMN1). This method helps prevent erratic response by providing a more constant TSO load (that is, a constant arrival rate of TSO work).

Each batch initiator class is associated with a unique performance group. The default performance group (PGN1) is used only by started tasks.

If a class B job has not completed in one hour, it is given a special boost to enable it to complete in the remaining hour of its turnaround requirement (see PGN4).

Class A batch is run only at night, with no particular turnaround requirements. Jobs can execute to completion without swapping since no workload control is necessary. Therefore, a large ISV is used.

## Evaluating and Adjusting the IPS and OPT

This section is intended as an aid in interpreting measurement data and adjusting the IEAIPSxx and IEAOPTxx parameter values.

The first part of this section addresses possible causes for a failure to meet fixed response and throughput requirements. The analysis relies on MF/1 or RMF reports, and other measurement data as established by the installation when defining the fixed requirements.

The second part addresses the optimal use of the SRM control mechanisms, that is, how to achieve fixed requirements with a minimum of swapping and paging overhead, and how to maximize throughput via job mix control.

Note that the controls offered by the SRM are only one aspect of the tuning process. Therefore, adjustment of IPS and/or OPT parameters may not be effective if basic system tuning is needed.

### Fixed Requirements Are Not Being Met

**Problem 1:** Short TSO Response Time is Erratic.

**Detection:** User feedback or personal experience while using TSO will be the best detection method. MF/1 and RMF report only average response times.

#### *Possible Causes:*

- a) The dispatching queue is not ordered properly.  
If short TSO transactions are placed on the dispatching queue behind relatively long transactions, long response times can occur. Use the APG feature in the IPS to place long TSO transactions at a lower dispatching priority than first period TSO transactions, and ensure that the long transactions are indeed switching to a new period (DUR value for short TSO transactions is not too large).

Also, examine the use of dispatching priorities higher than or equal to the short TSO transaction priority. There may be dispatching queue interference from memory create processing and from the initiator priority as specified in "SYS1.PROCLIB(INIT)".

- b) The number of TSO users waiting to be swapped in is not uniform.  
Sporadic waiting periods translate into user-perceived erratic response time. RMF tracing reports will indicate the number of users swapped out for the short TSO transaction domain. If this value fluctuates, there are several possible solutions:
- 1) Increase the target MPL for the short TSO transaction domain by raising the minimum MPL above the observed average number of ready address spaces. The minimum MPL may need to be increased several times, even to the mean observed maximum number of ready address spaces. This will help ensure that when a TSO address space becomes ready, it will be swapped in. This will provide consistent short TSO transaction response, but may also mean that short TSO transactions will consume more system resources than is tolerable, introducing, for example, sudden heavy page demands.



**Problem 4: IMS Response Time or Transaction Rate Is Too Good.**

**Detection:** The IMS statistical program, or any other method decided upon when specifying installation requirements, will indicate whether this is a problem.

**Possible Causes:**

Dispatching priority is too high.

Whether IMS is swappable or non-swappable, if its dispatching priority is placed in the APG range and controlled by the SRM, other work may be placed above or equal to the IMS priority, thus causing IMS to slow down.

**Problem 5: Batch Turnaround Time Is Not Good Enough.**

**Detection:** SMF type 5 records provide elapsed time information for all jobs. MF/1 and RMF provide average elapsed times. The installation will have specified the criteria for elapsed times when establishing fixed requirements.

**Possible Causes:**

a) Job classing may be ineffective.

If tape merges, file sorts, or other long jobs are run in a job class with jobs that have fast turnaround requirements, they could monopolize the initiators while short jobs wait on the queue for selection. SMF type 5 records provide sufficient information to determine whether jobs are indeed waiting a long time to be initiated. Ensure that jobs are placed in the proper initiator class.

b) TSO or IMS may be using more of the system than planned.

If this is true, batch work will have less resources available. Either reinvestigate fixed requirements, or refer to appropriate problems listed in this section. If resources are available and the batch domain is consistently at its maximum MPL, raise the maximum.

c) Swapping may be slowing down throughput.

The initial ISV for the batch performance group should be large enough to allow the majority of short jobs to complete without a swap. This is an unlikely cause of this problem since swapping is a control that executes in a range of seconds, while throughput is measured in minutes or hours. However, if an excess of initiators are started, there may be inordinate waiting times when swapped out.

d) Jobs that don't have a fast turnaround requirement may be interfering with jobs that do.

Within the batch domain, use objectives to distinguish between the performance needs of various types of work. The properties of slopes, ISVs, and cut-off points can help ensure that turnaround requirements are met.

## Optimizing the Use of Control Mechanisms

### Problem 6: Swapping May Be Causing Excessive Overhead.

**Detection:** The RMF paging activity report lists total swap out counts. These counts will indicate the causes contributing to swapping overhead. Some of these can be attributed to the SRM, some to applications. Since swapping is a costly control mechanism, the number of unnecessary swaps should be kept at a minimum.

#### *Possible Causes:*

- a) "Input terminal wait" swap count is high.  
In a TSO environment, the vast majority of swaps should be of this type. This is not a problem.
- b) "Output terminal wait" swap count is above 0.
  1. TSO commands or CLISTs may produce output so fast that the TIOC does not have sufficient buffers available or sufficient buffers may not have been allocated initially. This causes the address space to be automatically swapped out of storage until the output buffers have been emptied. The SRM treats this condition as the end of a transaction. If the excess swaps are counted here, the problem may be eliminated by increasing the OWAITHI value in the IKJPRMxx PARMLIB member.
  2. Another application-induced practice that may cause an output wait swap is the issuance of a TPUT SVC with the 'HOLD' keyword.
- c) "Long wait" swap count is high.  
Applications that expect to wait for long periods of time (WAIT macro with the LONG option, STIMER for greater than 0.5 seconds) will be swapped out. Investigate the possibility of rewriting such applications.
- d) "Detected wait" swap count is large.  
These swap outs occur when an address space in storage has been non-dispatchable for a fixed period of time as, for instance, when an initiator finishes with a job and then finds no other available jobs on the queue. If this count is larger than the number of swaps that can be attributed to idle initiators, investigate which jobs are waiting and correct the responsible program(s).
- e) "Unilateral" swap count is high.  
In a batch environment, this count may equal the number of ended batch jobs. If the count is higher than this, the extra swaps may have been caused by the staging of work through several domains with improper minimum MPL values.

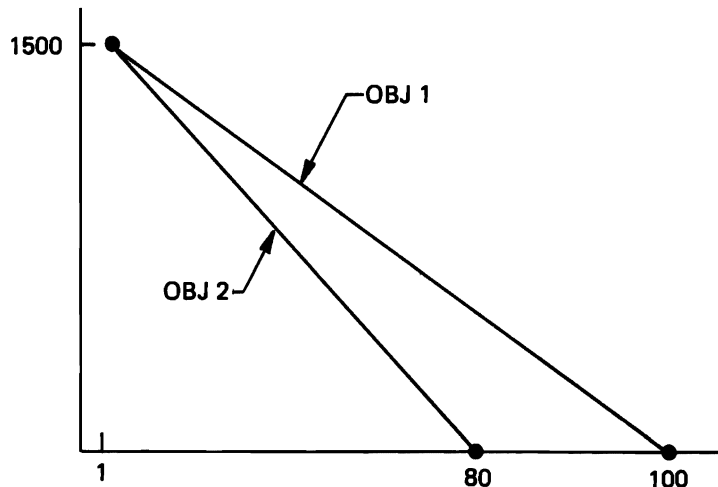
For example, if medium TSO transactions are switched from a domain with a high target MPL into a domain with  $\text{minMPL} = \text{maxMPL} = 1$ , only one medium transaction will be allowed in storage at a time. The others will be swapped out because the target MPL is exceeded, that is, unilateral swap outs will occur.

The RMF workload activity report indicates, by performance group period, the ratio of swaps to ended transactions. This can assist in finding the period causing extra swapping.

Note, however, that the technique of delay for TSO as described under Problem 1 will not induce unilateral swaps because the delay is incurred in the first period, before the TSO address space is swapped into storage.

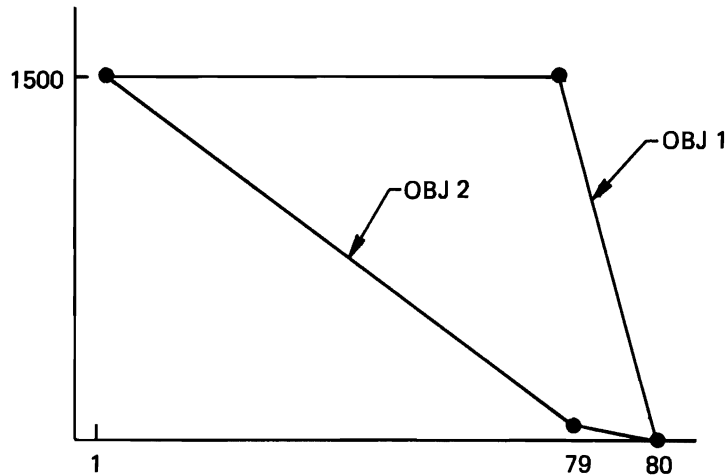
- f) "Exchange on recommendation value" swap count is high. These control swaps are dictated by the IPS and may be excessive for several reasons.
- 1) An ISV value is too small.  
Remember that the ISV is the primary determinant of swapping frequency. It should be large enough to accommodate transactions with such severe response time requirements that swapping is an inappropriate control. RMF workload reports that summarize by domain and period will help locate the area causing the extra swapping.
  - 2) Different cut-off workload levels exist in a domain.  
The use of different cut-off workload levels is a valid workload management technique. It provides a simple priority scheme within a domain. However, address spaces associated with the objective having the higher cut-off point may be constantly preempting address spaces on other objectives, even in their ISV. If this preemption is frequent, there may be a problem in domain definition.

For example:



In a batch domain, a 'hot' job on objective 1 will preempt normal jobs on objective 2, and cause an exchange swap. If an installation has many hot jobs, however, they may be candidates for a domain of their own to prevent interference with normal batch work.

Another solution would be to use the same cut-off workload levels, but utilize objective slopes to indicate job priority. Now the hot job on objective 1 will not preempt normal batch until the latter leaves its ISV, after which the hot job will be swapped in and execute to completion.



- g) "ENQ Exchange" swap out count is not close to zero. High contention for serial resources is causing extra swapping. Even if this count is low, there may still be an enqueue bottleneck if the "unilateral swap count" in a batch environment is much higher than the number of ended batch transactions. This is possible because the SRM will exceed the target MPL, if necessary, to ensure that a resource holder is swapped in for its ERV time. Investigate enqueue contention in the system.

#### Problem 7: Paging Overhead Is High.

**Detection:** The RMF paging activity report shows the total system non-swap page-in and page-out values. If the sum of these values indicates that significantly more than 5% of CPU time is used for paging, the installation is probably incurring excessive overhead. Five percent is a rate of approximately 15 pages per second for a 158 UP and approximately 30 pages per second for a 168 UP.

In addition to this report, RMF traces the highest system UIC and the count of available frames. A UIC value that is consistently 0 could indicate an overcommitment of storage, since this means that no pages are allowed to go unreferenced for 1 second. This condition defeats the SRM's storage management algorithm, that is, no distinction between the age of frames is possible.

#### Possible Causes:

- a) Overcommitment of real storage, probably due to large minimum MPL values. The SRM will adjust each domain's target MPL based on system contention indicators and the constraint values. If the paging rate indicates the need to decrease the total MPL, but all target MPLs are currently at the minimum MPL values, the SRM cannot respond. The installation should either increase the amount of real storage, or decrease the minimum MPL value for at least one domain.

## b) Examine the use of the link pack area (LPA).

Investigate the feasibility of packing and repackaging to obtain more efficient use of storage.

**Problem 8: Resources are Unused.**

**Detection:** RMF reports indicate system resource usage. If these show a high CPU wait time, little paging, or low channel activity, the system may not be fully utilized.

**Possible Causes:**

## a) The maximum MPL for domains is too low.

If all domains have a target MPL equal to the maximum value, the SRM is unable to increase system utilization by raising the target MPL of a domain. Therefore, resources will be underutilized. Increasing the maximum MPL of at least one domain that consistently has ready work waiting should alleviate this.

## b) Some resource is a bottleneck.

If RMF indicates several underutilized resources, but shows one to be consistently overutilized, increasing the MPL of a domain will probably not help. Instead, the service definition coefficients may be altered by raising the coefficient of the overutilized resource. This will have the effect of increasing the service rate for users of that resource, thus producing a lower workload level and causing earlier swap outs.

*Note:* Changing the service definition coefficients may require changing other service dependent parameters in the IPS (ISV, DUR and OBJ). Also, RMF and SMF transaction and service statistics will differ from previous reports.

If the overutilized resource can be associated with a specific performance group or period, the RTB for that period or performance group may be set to 1 to allow the load adjusting functions to eliminate the bottleneck. In conjunction with these changes, the resource factor coefficient of the overutilized resource may need to be raised to give more weight to the load adjusting swap recommendations.

**Section 4: Installation Management Controls**

This section contains information about JCL statements and operator commands for SRM-related installation management functions.

**Assigning Performance Group Numbers**

The PERFORM parameter on the JOB or EXEC JCL statement is used to associate a performance group with a job or job step (see the publication *OS/VS2 JCL*, GC28-0692).

The PERFORM parameter value on the JOB statement associates all steps of a job with a performance group. If no PERFORM parameter appears on the JOB statement, the parameter value on the EXEC statement will be used for the associated job step.

If an invalid performance group is specified, a warning message indicating non-verification and default substitution will be issued. The default for non-TSO jobs is 1; for TSO jobs it is 2.

If PERFORM is specified for a procedure, the specified value is effective for the entire procedure. If PERFORM.procstepname is coded for a procedure, the value is effective only for the procedure step named.

If no PERFORM parameter is specified on either the JOB or EXEC statements, a default will be assigned. If JES2 is the primary I/O subsystem, the default values may be specified by the installation in the JES2 initialization parameters. (See the parameters &STC, &TSO, &X in the publication *OS/VS2 SPL: JES2*, GC23-0001. Defaults may be assigned for started tasks, time sharing users or any batch class.)

If a performance group is not specified via the PERFORM parameter and is not assigned by JES2, the IBM-supplied default will be assigned.

For TSO users, the PERFORM parameter may also be specified on the LOGON command. (See the publication *OS/VS2 TSO Command Language Reference*, GC28-0646). If the user requests a performance group not specified in the UADS, the IBM-supplied default of 2 is assigned.

### Assigning Dispatching Priorities

The DPRTY parameter, which may be included on the EXEC JCL statement, is used to assign a dispatching priority to a job step. (See the publication *OS/VS2 JCL*, GC28-0692.) Job steps that do not include the DPRTY parameter on their EXEC statements are automatically assigned an APG priority.

If the job step is assigned an APG priority, the specific priority value is obtained from the IPS specification. If no value is specified in the IPS for the corresponding period, the default is 6 (mean-time-to-wait).

### Operator Commands Related to the SRM

The system operator can directly influence the SRM's control of specific jobs or groups of jobs by entering commands from the console. The exact format of these commands is defined in the publication *Operator's Library: OS/VS2 MVS System Commands*, GC38-0229.

The RESET command is used to change the performance group of an executing job.

The SETDMN command provides a way for altering the constraint values on a domain's multiprogramming level. The information from this command is valid only for the life of the IPL — it does not change fields in the IPS member.

The SET command with the IPS parameter is used to switch to a different IPS after an IPL. The SRM will base all control decisions for existing and future jobs on the parameters in the new IPS.

To understand whether the system is running effectively with the current IPS, an installation needs dynamic system status information. The DISPLAY command with the DOMAIN keyword provides a snapshot of an individual domain's status.

## Section 5: SRM Parameters

This section describes the syntax rules, keywords, and the value ranges and defaults for the parameters specified in the SYS1.PARMLIB members IEAIPSxx and IEAOPTxx.

### Syntax Rules

Keywords, appearing in upper case letters, must be written exactly as indicated. No imbedded blanks are permitted in the keywords, and the keywords may not span logical records. Information is written on 80 byte logical records. The data may extend from byte 1 through byte 71; the contents of bytes 72 through 80 will be ignored.

Any number of blanks may follow a keyword. All required keywords must be specified in the sequence shown, except where noted, and must be separated by a delimiter. A delimiter, shown in the syntax descriptions as a comma, can be a comma followed by one or more blanks, or it can be any non-zero number of blanks.

Comments are permitted whenever blanks are allowed. The general format of a comment is:

```
/*character string*/
```

The slash and asterisk must be immediately adjacent. The character string may contain any characters except the \*/ combination.

### IEAIPSxx Parameters

The IPS contains five categories of information. They must be specified in the order shown.

1. Service definition coefficients:

```
[CPU=xx.x] [IOC=xx.x] [MSO=xx.x]
```

(Note: these keywords can be specified in any order.)

2. Workload levels:

```
WKL=(xxx,xxx[, . . .])
```

3. Performance objectives:

```
OBJ=xx, SRV=(xxx[, . . .])
```

The maximum number of performance objectives that can be specified per IPS is 64.

## 4. Domains:

DMN=xxx, CNSTR=(xxx,xxx,xxx)

The maximum number of domains that can be specified per IPS is 128.

## 5. Performance groups:

PGN=xx, (DMN=xx,OBJ=xx,DUR=xxxxxxxxx

$\left[ ,\text{UNT}=\left\{ \frac{\text{S}}{\text{R}} \right\} \right] [,\text{ISV=xxxxxx}] [,\text{RTB=x}]$

$[,\text{APG=xx}]$

$[(. . .)]$

The keywords within a set of parentheses represent a performance group period.

The maximum number of performance group periods that can be specified per performance group is 8. The maximum number of performance groups that can be specified per IPS is 255. (*Note:* The keywords comprising a performance group period can be specified in any order.)



*Parameter Descriptions*

Keyword	Meaning	Value Range	Default Value									
APG	<p>Specifies the dispatching priority within the APG range that will be assigned to a transaction.</p> <p>The APG is divided into three distinct groups:</p> <table border="0"> <tr> <td>Mean-time-to-wait</td> <td>0-6</td> </tr> <tr> <td>Rotate</td> <td>11</td> </tr> <tr> <td>Fixed</td> <td>7-10, 12-15</td> </tr> </table> <p><i>Example:</i> APG=12</p> <p>The resultant dispatching priority, assuming the default APG range, is 7C.</p>	Mean-time-to-wait	0-6	Rotate	11	Fixed	7-10, 12-15	0-15	6			
Mean-time-to-wait	0-6											
Rotate	11											
Fixed	7-10, 12-15											
CNSTR	<p>Specifies the constraints of a domain, that is, the minimum MPL, the maximum MPL, and the weighting factor, respectively. All three constraint values must be specified. The value ranges for the constraints are:</p> <table border="0"> <tr> <td>minimum MPL</td> <td>0-255</td> <td>none</td> </tr> <tr> <td>maximum MPL</td> <td>0-255</td> <td>none</td> </tr> <tr> <td>weighting factor</td> <td>1-255</td> <td>none</td> </tr> </table> <p><i>Example:</i> DMN=2,CNSTR=(7,10,15). This indicates, for domain number 2, a minimum MPL of 7, a maximum MPL of 10, and a weighting factor of 15.</p>	minimum MPL	0-255	none	maximum MPL	0-255	none	weighting factor	1-255	none		
minimum MPL	0-255	none										
maximum MPL	0-255	none										
weighting factor	1-255	none										
CPU	<p>Indicates the number by which accumulated CPU service units will be multiplied (weighted).</p> <p><i>Example:</i> CPU=1.0</p>	0.0 - 99.9	1.0									
DMN	<p>Specifies a unique number that is used to associate a domain with a transaction.</p> <p><i>Example:</i> DMN=5</p> <p><i>Note:</i></p> <p>If domain 1 is not specified in the IPS, it will be defined automatically with constraints of (1,255,1).</p> <p>If no domains are defined in the IPS, the domain for each performance group period will default to 1, with constraints of (1,255,1).</p> <p>If a domain is specified in the IPS, the first period of a performance group must contain a domain specification.</p> <p>If no domain is specified in subsequent periods, the last previously specified domain number will be assigned.</p>	1-128	(see Note)									

Keyword	Meaning	Value Range	Default Value
DUR	<p>Specifies the length of the performance group period in units indicated by the UNT keyword.</p> <p><i>Example:</i> DUR=400</p> <p>Every performance group period must have the DUR parameter specified except for the last performance group period in a performance group.</p> <p>*If UNT=R is specified for a performance group period and a DUR value greater than 1,000,000 is assigned, 1,000,000 is used in place of the assigned value.</p>	0-999999999 or 0-999999K*	none
IOC	<p>Indicates the number by which accumulated I/O service units will be multiplied (weighted).</p> <p><i>Example:</i> IOC=1.0</p>	0.0 - 99.9	none
ISV	<p>The interval service value specifies the minimum number of service units that a transaction should receive during each interval of real storage occupancy.</p> <p><i>Example:</i> ISV=800</p>	100 - 999999 or 100 - 999K	100
MSO	<p>Indicates the number by which accumulated storage service units will be multiplied (weighted).</p> <p><i>Example:</i> MSO=1.0</p>	0.0 - 99.9	1.0
OBJ	<p>Specifies a unique number that associates the performance objective with a particular period in the processing of a transaction.</p> <p><i>Example:</i> OBJ=8,SRV=(200,100,0)</p>	1-64	none
PGN	<p>Specifies a unique identifier for a performance group definition. This number is used to associate a job, job step or time-sharing session with the respective performance group.</p> <p><i>Example:</i> PGN=3</p> <p><i>Note:</i> Each IEAIPSxx member must have performance groups 1 and 2 specified.</p>	1-255	none
RTB	<p>Specifies the response/throughput bias for a particular performance group period.</p> <p>A value of "1" indicates that the CPU and I/O load adjusting recommendations are to be added to the workload recommendation during swap analysis. A value of "0" indicates that the CPU and I/O load adjusting recommendations are to be ignored.</p>	0-1	0
SRV	<p>Specifies the service rates that define performance objectives. The number of service rates may be equal to or less than the number of workload level values.</p> <p><i>Example:</i> OBJ=3,SRV=(1000,500,0)</p>	0-9999	none

**Notes:**

1. If more service rates than workload level numbers are specified, the extra service rates are ignored.
2. An asterisk (\*) can appear in place of a numerical SRV value. When it appears between two numerical values (for example SRV=(100,\* ,0)), it indicates that a linear graph connects the point before the asterisk with the point after the asterisk. Thus, with workload levels WKL=(10,20,30), SRV=(100,\* ,0) is equivalent to SRV=(100,50,0). If no number follows the asterisk, the line joining the previous two values is extended. Thus, with WKL=(10,20,30), SRV=(100,75,\* ) is equivalent to SRV=(100,75,50). If only one value precedes the asterisk, and no values follow the asterisk, the single value is repeated. Thus, with WKL=(10,20,30), SRV=(100,\* ) is equivalent to SRV=(100,100).

<b>Keyword</b>	<b>Meaning</b>	<b>Value Range</b>	<b>Default Value</b>
UNT	<p>Specifies the units for the DUR parameter. "S" indicates service units; "R" indicates real time in seconds. In the following example, the period duration for objective 6 is 4 seconds; for objective 7, 4000 service units.</p> <pre>PGN=2, (DMN=1,OBJ=6,DUR=4,UNT=R)       (DMN=1,OBJ=7,DUR=4000)       ( . . . )</pre>	S,R	S
WKL	<p>Specifies a series of positive numbers of increasing value. They provide a reference for defining performance objectives. There can be up to 32 numbers in any workload level specification. The following example illustrates a workload specification and two related performance objectives.</p> <pre>WKL=(1,50,99,100) OBJ=3,SRV=(400,300,* ,0) OBJ=4,SRV=(400,300,0)</pre>	1-128	none

**IEAOPTxx Parameters**

The OPT contains two categories of information:

1. Resource factor coefficients:  
[RFC [CPU=x.x] [,IOC=x.x]]
2. Resource manager constant:  
[RMC ERV=xxxxxx]

(*Note:* refer to the syntax rules described at the beginning of this section.)

***Parameter Descriptions***

<b>Keyword</b>	<b>Meaning</b>	<b>Value Range</b>	<b>Default Value</b>
CPU	Specifies the weighting factor by which the recommendation of the CPU load adjusting function is to be multiplied, when an address space is evaluated for swapping. <i>Example:</i> CPU=3.0	0.0 - 9.9	0.1
IOC	Specifies the weighting factor by which the recommendation of the I/O load adjusting function is to be multiplied, when an address space is evaluated for swapping. <i>Example:</i> IOC=2.0	0.0 - 9.9	0.1
ERV	Specifies the "enqueue residence value". This is a multiplication factor used to calculate the amount of task execution time during which an address space should not be swapped out, if the address space is enqueued on a system resource needed by another address space. <i>Example:</i> ERV=2	0-999999	1

***Note:***

The SRM determines the execution time by multiplying the ERV by the model-dependent time needed to accumulate 1 CPU service unit.

In the example given, if an address space can consume 1 service unit in 10 milliseconds, it will be allowed to execute for 20 milliseconds, when it is enqueued on a resource requested by other address spaces, before it will be eligible for swap-out.

## Section 6: SRM Constants

This section lists internal SRM constants which an installation with unique resource management requirements may want to change. Before making such a change, it will, in most cases, be advisable to study related segments of code in order to be able to predict the impact of the contemplated change on internal SRM algorithms and, therefore, on SRM evaluations and decisions.

In the table of constants, internal control blocks are denoted by their acronyms, as follows:

- RCT – Resources Control Table
- RMPT – Resources Manager Parameter Table
- MCT – Storage Management Control Table
- ICT – I/O Control Table
- PVT – Page Vector Table
- CCT – CPU Management Control Table

Constant	Control Block	Related Function	Purpose	Value
RCCUICTL	RCT	MPL adjustment	UIC low threshold	2
RCCUICTH	RCT	MPL adjustment	UIC high threshold	4
RCCCPUTL	RCT	MPL adjustment	CPU utilization low threshold	96% (times a scaling factor of 16)
RCCCPUTH	RCT	MPL adjustment	CPU utilization high threshold	100% (times a scaling factor of 16)
RCCPRTL	RCT	MPL adjustment	Paging rate low threshold	100 pages/second
RCCPTRTH	RCT	MPL adjustment	Paging rate high threshold	100 pages/second
RCCASMTL	RCT	MPL adjustment	ASM queue length low threshold	7
RCCASMTH	RCT	MPL adjustment	ASM queue length high threshold	10
RCCTOTUT	RCT	MPL adjustment	Average total logical channel utilization threshold	20%
RCCLCHUT	RCT	MPL adjustment	Single logical channel utilization threshold	30%
RCCLCHRR	RCT	MPL adjustment	Single logical channel EXCP request rate threshold	50 EXCPs/second
RMPTTOM	RMPT	SRM invocation	Minimum SRM invocation interval	500 milliseconds
RMPTXCHT	RMPT	Swap analysis	Exchange swap threshold	1 (times a scaling factor of 256)
RMPTSAET	RMPT	Swap analysis	Swap evaluation threshold	1000 milliseconds
MCCPLUS	MCT	Page stealing	Available frame queue delta for stealing	10
MCCINTMX	MCT	Page stealing	Maximum UIC updating interval	2000 milliseconds
MCCUICBD	MCT	Page stealing	Maximum system UIC before UIC updating interval changes	8
MCCASMT1	MCT	Auxiliary storage shortage detection	Auxiliary storage shortage threshold	70%
MCCASMT2	MCT	Auxiliary storage shortage detection	Critical auxiliary storage shortage threshold	85%
ICCMNSWP	ICT	I/O load adjusting	Minimum swap out time for I/O imbalance correction	30000 milliseconds
ICCMNIOR	ICT	I/O load adjusting	Minimum I/O rate for user I/O monitoring	5 EXCPs/second
PVTPERFX	PVT	Pageable storage shortage detection	Maximum fixed frame threshold	80%

## VS2.03.807

Constant	Control Block	Related Function	Purpose	Value
PVTPEROK	PVT	Pageable storage shortage detection	Acceptable fixed frame threshold	60%
ICCED\$UT	ICT	Device allocation	Projected impact on logical channel utilization of a previously allocated data set	5%
ICCNHI1	ICT	I/O load adjusting	Percentage of delayed I/O requests above which a logical channel is considered to be overutilized for a uniprocessor system	70%
ICCNHI2	ICT	I/O load adjusting	Percentage of delayed I/O requests above which a logical channel is considered to be overutilized for a multiprocessor system	80%
ICCNLO1	ICT	I/O load adjusting	Percentage of delayed I/O requests below which a logical channel is considered to be underutilized for a uniprocessor system	30%
ICCNLO2	ICT	I/O load adjusting	Percentage of delayed I/O requests below which a logical channel is considered to be underutilized for a multiprocessor system	40%
ICCSIGUP	ICT	I/O load adjusting	Percentage of I/O requests to a logical channel that determines a heavy user of that channel	5%
ICCMNLIN	ICT	I/O load adjusting	Minimum elapsed time before a logical channel utilization computation will take place	2000 milliseconds
ICCMNUIN	ICT	I/O load adjusting	Minimum elapsed time before a user I/O rate computation will take place	10000 milliseconds
ICCMXICT	ICT	I/O load adjusting	Maximum elapsed time a heavy I/O user may remain in real storage without having its I/O usage monitored	60000 milliseconds
CCCUTHIT	CCT	CPU load adjusting	Percentage of time a CPU was not in the wait state, above which the CPU will be considered overutilized	100%
CCCUTLOT	CCT	CPU load adjusting	Percentage of time a CPU was not in the wait state, below which the CPU will be considered underutilized	80%
CCCSIGUR	CCT	CPU load adjusting	Minimum mean-time-to-wait to be considered a heavy CPU user	28 milliseconds
CCCAPDIV	CCT	APG	Range of a subgroup within the mean-time-to-wait group (value given is for Model 155 Mod 11)	4 milliseconds
CCCAPRHT	CCT	APG	Percentage of APG users whose priority must change at an invocation of this function, before the invocation interval is shortened	60%
CCCAPMET	CCT	APG	Minimum user execution time before a new dispatching priority will be computed for the user	200 milliseconds
CCCAPRLT	CCT	APG	Percentage of APG users whose priority must change at an invocation of this function, before the invocation interval is lengthened	20%
CCCAPROT	CCT	APG	Rotate dispatching priority in APG	11
CCCAPGDP	CCT	APG	Default mean-time-to-wait dispatching priority in APG	6

## Part 4: How to Use the System Activity Measurement Facility (MF/1)

The System Activity Measurement Facility (MF/1) is an analysis tool which an installation may use to monitor selected areas of system activity, and obtain feedback in the form of SMF records and/or formatted reports. MF/1 permits the gathering of information on the following classes of system activity, either individually or in combination:

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

With MF/1, an installation can monitor the utilization of individual CPU's, channels and devices, in order to identify system components whose overall utilization is exceptional. The installation can further identify periods of system activity during which the utilization of particular resources are exceptional. Finally, the installation can relate the service being provided to different classes of users to the specifications provided in the Installation Performance Specification (IPS).

MF/1 is a software monitoring tool. Thus it is limited to reporting on system activity as that activity is communicated to the system (for example, by the setting of flags). As a result of this indirect reporting, MF/1's statistically sampled values can approach in accuracy only the internal system indications, not necessarily the external system activity itself. For example, if a CPU is disabled so that the freeing of a device (device end interruption) cannot be communicated to the system, the device will appear busy for a longer period of time than it would if it were measured by a hardware measuring instrument.

In an installation using the Mass Storage System (MSS), Mass Storage System Trace Report programs are used to monitor the MSS. These programs are described in *OS/VS Mass Storage System (MSS) Services for Space Management*.

### MF/1 Operation

MF/1 is always generated with the system, but its operation is completely optional. The system operator initiates MF/1 monitoring with the START command. MF/1 can also be started as a batch job. When MF/1 is not operating, it will cause little performance or storage overhead. When it is operating, the storage and performance overhead will depend on the set of options which were specified by the operator.

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

- The “parmvalue” 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, if used (the IBM-supplied procedure does not use it).
- The MF/1 partitioned data set member (typically a SYS1.PARMLIB member).

If a parameter is not specified in any of these sources, a program default value is used.

The MF/1 cataloged procedure is named MF1, and resides in the SYS1.PROCLIB data set. The IBM-supplied procedure is:

```
//IEFPROC EXEC PGM=IRBMFMFC,DPRTY=(15,15)
//IEFRDER DD DSN=SYS1.PARMLIB,DISP=SHR
//SYSUDUMP DD SYSOUT=A
```

The IEFRDER DD statement names the MF/1 library partitioned data set which may contain any number of members, each containing card-image MF/1 control input. The member names must be in the form IRBMF1nn, where nn is a decimal digit field. The default value for nn, “00”, may be changed by the MEMBER parameter, as indicated below in the explanation of the MEMBER parameter.

The START and STOP commands used to initiate and terminate MF/1 operation have the following syntax:

```
START MF1 [.identifier] [,devicename] [,volumeserial] [,parmvalue]
          [,keyword=option, . . .]
```

**Note:** The identifier, although optional during the starting of MF/1, must have been specified at START in order to STOP MF/1 from the operator’s console.

```
STOP [MF1.] identifier      (The STOP command will not take effect until
                             after the “MF/1 ACTIVE” message has been
                             received by the operator.)
```

The “parmvalue” field may be unused (if MF/1 control parameters are to be obtained from a lower priority source), or may contain one or more MF/1 options, to be used to control MF/1 for the duration of its execution. If non-alphameric characters are used in this field (for example, when more than one option is specified, and commas are therefore used as separators), the “parmvalue” field must be enclosed in parentheses.

The “devicename” and “volumeserial” fields are used only when the operator desires to override the specifications of the IEFRDER DD statement in the MF/1 cataloged procedure. When these fields are left blank (as will normally be the case), commas must be inserted to indicate their absence if MF/1 parameters are to be specified in the “parmvalue” field.



The “keyword=option” field may be used to specify any special keyword syntax desired for the IEFORDER DD statement, or any symbolic parameter keyword defined when a user-written cataloged procedure replaces the IBM-supplied one. Normally, this field will not be used.

An example of a typical operator request for MF/1 activity, specifying three MF/1 control parameters, is:

```
START MF1.EXAMPLE,,,(WKLD(SYSTEM), CYCLE(100),  
DEVICE(NOCOMM))
```

An example of starting MF/1 as a batch job is:

```
//MF1 JOB (accounting information)  
//MF1 EXEC MF1
```

## MF/1 Options

For a more detailed explanation of parameter selection and syntax rules, refer to the IRBMFxx member description in Part 2: System Initialization.

The MF/1 control options are summarized below. The underlined values are the program default values, and appear in the IBM-supplied SYS1.PARMLIB member IRBMF100. The values of these options to control an MF/1 execution are taken from the input sources in the following priority order:

1. START command “parmvalue” field
2. EXEC statement PARM field
3. MF/1 partitioned data set member

An option explicitly specified in the START command will take priority over any conflicting specifications of that parameter in the EXEC statement, or the MF/1 partitioned data set member. An option explicitly specified in the EXEC statement will take priority over a conflicting specification in the MF/1 partitioned data set member. If there are parameters for which no values are specified in the START command, the EXEC statement, or the MF/1 partitioned data set member, a program default value will be used.

OPTION	FUNCTION
{ CPU } { NOCPU }	Specifies whether or not system CPU activity is to be monitored by MF/1.
{ CHAN } { NOCHAN }	Specifies whether or not system channel activity is to be monitored by MF/1.
{ DEVICE (device list) } { NODEVICE }	Specifies whether or not system device activity is to be monitored by MF/1. If DEVICE is specified, a device list must indicate the classes of devices that will be monitored.

**OPTION**

**FUNCTION**

Device list choices:

$\left\{ \begin{array}{l} \text{CHRDR} \\ \text{NOCHRDR} \end{array} \right\}$	Character reader devices
$\left\{ \begin{array}{l} \text{COMM} \\ \text{NOCOMM} \end{array} \right\}$	Communications equipment
$\left\{ \begin{array}{l} \text{DASD} \\ \text{NODASD} \end{array} \right\}$	Direct access storage devices
$\left\{ \begin{array}{l} \text{GRAPH} \\ \text{NOGRAPH} \end{array} \right\}$	Graphics devices
$\left\{ \begin{array}{l} \text{TAPE} \\ \text{NOTAPE} \end{array} \right\}$	Magnetic tape devices
$\left\{ \begin{array}{l} \text{UNITR} \\ \text{NOUNITR} \end{array} \right\}$	Unit record devices
$\left\{ \begin{array}{l} \text{PAGING} \\ \text{NOPAGING} \end{array} \right\}$	Specifies whether or not system paging activity is to be monitored by MF/1.
$\left\{ \begin{array}{l} \text{WKLD} \\ \text{NOWKLD} \end{array} \right\} \left\{ \begin{array}{l} \text{(PERIOD)} \\ \text{(GROUP)} \\ \text{(SYSTEM)} \end{array} \right\}$	Specifies whether or not system workload activity is to be monitored by MF/1. If WKLD is specified, the level of detail of the report to be produced must be indicated by the option in the parenthesis. PERIOD requests the most detailed workload activity reporting, GROUP requests an intermediate level of detail, and SYSTEM requests the least detail.
CYCLE(value)	Specifies the frequency at which sampling observations are made of channel and device data. The range is from 50 to 999 milliseconds. The default is 250 milliseconds.
INTERVAL $\left\{ \begin{array}{l} \text{value} \\ \text{value M} \end{array} \right\}$	Specifies the interval at which all data will be gathered for report formatting and/or SMF record writing. The range is from 1 to 60 minutes. The default is 15M.
MEMBER(nn)	The value specified by this parameter is appended to IRBMF1 to form the name of the partitioned data set member that contains the MF/1 options. The default is 00, indicating member IRBMF100 in the partitioned data set named on the IEFORDER DD statement in the MF/1 cataloged procedure (normally SYS1.PARMLIB). This parameter may be specified in the PARM field of the START command or EXEC statement, but should <i>not</i> be specified within a partitioned data set member.
$\left\{ \begin{array}{l} \text{OPTIONS or OPTN} \\ \text{NOOPTIONS or NOOPTN} \end{array} \right\}$	Specifies whether or not a list of the keyword options to be used will be printed at the operator's console at MF/1 initialization. If the list is printed (OPTIONS or OPTN specified), the operator will be able to respond with any desired changes to the option list.  <b>Note:</b> If you don't plan to specify any options at MF/1 start time, you can avoid unnecessary console output and delay in activating MF/1 by specifying NOOPTIONS in the MF/1 partitioned data set member. If any syntax errors are detected by MF/1, OPTIONS is forced.
$\left\{ \begin{array}{l} \text{RECORD} \\ \text{NORECORD} \end{array} \right\}$	Specifies whether or not monitored data is to be written to the SMF data set. If SMF records are to be written, SMF data recording must have been specified during system initialization (MAN=ALL).

OPTION	FUNCTION
{ <u>REPORT</u> { <u>REALTIME</u> }   { <u>NOREPORT</u> { <u>DEFER</u> } }	Specifies whether or not printed reports of the monitored data are to be produced. If the reports are to be produced (REPORT specified), the REALTIME/DEFER options specify whether the reports are to be printed when formatted, at the conclusion of a data gathering interval (REALTIME specified), or printed after MF/1 processing terminates (DEFER specified).
{ <u>STOP</u> { value }   { <u>NOSTOP</u> { value M } }   { { value H } }	Specifies the desired time duration for MF/1 activity, in minutes or hours. The range is from one minute to one week (168 hours or 10,080 minutes). The operator STOP command will terminate execution at any time, regardless of the value for this parameter. The default value is 15M. The units default is M (minutes).
SYSOUT(class)	Specifies the SYSOUT class to which formatted reports are directed. Class A is the default.

### Conflicts Between Options

After the operator enters the START command to begin MF/1 execution, the control options are merged from all sources. After this merge has been performed, it is possible that some conflicts may arise in that some parameters should not be used in conjunction to control a single MF/1 execution. When any of these situations occur, MF/1 selects compatible values for these parameters, and notifies the operator of the selection. The possible conflicts are as follows:

CONFLICT	PROBLEM	MF/1 RESOLUTION
NOREPORT and NORECORD specified	No way for installation to obtain measurement data	Change NOREPORT to REPORT (DEFER)
STOP value specified is less than INTERVAL	Indicates MF/1 termination prior to obtaining any data	Set STOP value equal to INTERVAL value
REPORT(DEFER) and NOSTOP specified	SYSOUT will become cluttered with unprinted reports	Set STOP value equal to INTERVAL value

### Need for Careful Choice of Certain Parameters and Options

Care should be taken in specifying certain groups of MF/1 parameters. Though not causing conflicts, injudicious choices of these parameters can cause undesirable results. These include poor choices of: device class selection for monitoring devices, INTERVAL and CYCLE values, and STOP, INTERVAL, and REPORT values.

#### INTERVAL and CYCLE Parameters

Much of the data in the channel and device activity reports is statistically sampled. Since according to statistical theory the accuracy of sampled data increases with the number of samples taken of random events, an installation would expect to observe more precise results with decreased CYCLE time (for a fixed INTERVAL value), or with increased INTERVAL length (for a fixed CYCLE value). For example, 400 samples taken of random independent events provide a value which, with 90% Confidence, should fall within 4% of the true value; 1,600 samples of random independent events decrease to 2% the expected range of error, with 90% confidence.

However, pure statistical predictions are not always applicable to a software measurement tool such as MF/1 because the assumptions on which they are based (unbiased random independent samples and an infinite population) might not hold in an operating environment. Bias might occur because MF/1 samples internal indications of external system events. Thus MF/1 values might not precisely approach the values measured by a hardware measurement tool. The independence assumption becomes less and less realistic as CYCLE gets very small. As CYCLE gets smaller, each sample is more likely to find the system performing the same functions as in the previous sample; therefore, the new sample adds little additional information. The use of a smaller CYCLE value (while holding INTERVAL constant) should not be detrimental to accuracy, but any increase in accuracy might be of questionable benefit when compared with the system overhead that is introduced. A reasonable minimum CYCLE is a function of the timing characteristics of the hardware being measured.

Sample size can be increased by increasing INTERVAL length, by decreasing CYCLE length, or both. When decreasing CYCLE length, the installation should be aware that the increase in system degradation due to MF/1 operation could become significant, especially in the area of device measurements. At each sampling cycle, there is system overhead introduced due to the processing of the periodic interruption, and due to the data collection that occurs after the interruption. Device data collection is the major contributor to this type of overhead. Installations with a large number of UCBs associated with devices generated with the system can use the following approximation of the total overhead (number of instructions executed) associated with each sampling cycle:

$$\text{Number of instructions} \cong (10 \times D_1) + (30 \times D_2),$$

where  $D_1$  = number of UCBs associated with offline devices for monitored device classes, and  
 $D_2$  = number of UCBs associated with online devices for monitored device classes.

Thus, particularly when the number of UCBs for monitored device classes is large, the CYCLE value should not be made too small. In particular, avoid specifying a CYCLE value of 50 milliseconds.

The total overhead should be considered when the installation specifies the CYCLE value, so that total overhead for MF/1 sampling will not distort the normal system environment being measured.

*Device Class Selection:* Since MF/1 overhead is directly related to the number of device classes being monitored, the DEVICE option device list should include only those devices that require monitoring. For example, if the user is not interested in monitoring character readers and communications equipment, NOCHRDR and NOCOMM should be specified in the device list. This specification will eliminate MF/1 overhead for those device classes.

### **STOP, INTERVAL, and REPORT Parameters**

As was mentioned above, the specification of NOSTOP along with REPORT (DEFER) is considered a conflict by MF/1, because of the possible filling up of SYSOUT spool space. A similar problem can occur when the STOP value specified is very large (the largest specifiable value is 168 hours), the INTERVAL value is small, and REPORT (DEFER) is specified.

## MF/1 Reports and SMF Records

MF/1 produces feedback on the system activity data it monitors in the form of printed reports and/or SMF records. These reports/records are produced for each measurement interval and when MF/1 is terminated with an operator STOP command. The printed reports which can be produced by MF/1 are:

- CPU Activity
- Channel Activity
- Unit Record Device Activity
- Graphic Device Activity
- Direct Access Device Activity
- Communication Equipment Activity
- Magnetic Tape Device Activity
- Character Reader Device Activity
- Paging Activity
- Workload Activity (by performance group period, by performance group, or for the system as a whole)

The SMF records which can be produced by MF/1 are:

- SMF70 – CPU Activity
- SMF71 – Paging Activity
- SMF72 – Workload Activity (one record for each performance group)
- SMF73 – Channel Activity
- SMF74 – I/O Device Activity (one record for each device class for which data gathering was requested)

Each SMF record contains information similar to the contents of the corresponding formatted report. Totals, averages, and percentages are not explicitly contained in the SMF records, but may be calculated from the explicit data. The precise formats of the SMF records are contained in *OS/VS System Management Facilities (SMF)*. Elaboration of particular fields in these records may be obtained by consulting the descriptions of the corresponding fields in the following printed report descriptions:

Each printed report has the following header information in common:

- *Type of Operating System*: The label OS/VS<sub>n</sub>, where n is the system indication.
- *Release Number*: The label RELEASE nn.11, where nn and 11 are the number and level.
- *Date of Measurement*: The date on which the measurements were started, in the form DATE mm/dd/yy.
- *Time of Day*: The time of day that the measurements were started, in the form TIME hh.mm.ss.
- *Measurement Interval*: The length of the interval after which unique sets of measurements are printed out, in the form INTERVAL mm.ss.ttt. The end of the measurement interval is the sum of the recorded start time and the interval length.
- *Report Title*: The identification of the report, as one of the ten MF/1 report types.
- *Page Number*: The page number of the report output, in the form PAGE xxxx.

- *System ID*: The four-character SMF identifier associated with this system at system generation or initialization. The format is SYSTEM ID cccc.
- *Report Version*: The two-digit revision level of the report, in the form MF/1 VERSION nn.

All calculated numerical values in all MF/1 reports are rounded to the nearest printable value, unless otherwise noted in an individual report description. Asterisks are printed for numbers too large for the print field.

## 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. A sample of this report is contained in Figure 4-1. Entries are included for each CPU which has been online at the end of at least one reporting interval since MF/1 was started. Numerical data is not printed for CPUs which were offline at the end of the reporting interval, or had any VARY activity during the interval. CPUs for which data is not printed have one of the following messages in the first data field:

- |              |   |
|--------------|---|
| NOW ONLINE:  | Means that the CPU was varied online during this interval and is online at the end of the interval.   |
| NOW OFFLINE: | Means that the CPU was varied offline during this interval and is offline at the end of the interval. |
| OFFLINE:     | Means that the CPU has been offline for the entire interval.  |

The data fields in this report include:

- *CPU Number*: An integer (either 0 or 1), identifying the CPU in the system. This integer is the one produced by the STORE CPU ADDRESS (STAP) instruction (zero for a uniprocessor).
- *Wait Time*: The amount of time during which the CPU is not executing instructions (PSW wait state bit is on) in hours, minutes, seconds, and thousandths of a second.
- *Wait Time Percentage*: The fraction of the reporting interval that is represented by wait time. The range of values is 00.00 to 100.00.
- *CPU Serial Number*: The 6-digit number obtained by the store CPU ID instruction (STIDP). This number coincides with the physical serial number stamped on the CPU and, in conjunction with the model number, permits unique identification of the CPU.

C P U A C T I V I T Y				INTERVAL 04.00.144	PAGE 1
OS/VS2	SYSTEM ID H158	DATE 1/17/74			
RELEASE 02.0	MF/1 VERSION 01	TIME 13.35.37			
CPU NUMBER	WAIT TIME HH.MM.SS.TTT	WAIT TIME PERCENTAGE	CPU SERIAL NUMBER		
0	00.00.03.403	1.41	010060		
AVERAGE		1.41			

Figure 4-1. CPU Activity Report

## Channel Activity Report

This report provides information about channel loading for all channels in the system. Channel entries are included for each channel in the system which has been online at least one observation cycle since MF/1 was started. Data is not formatted for channels which were offline at the end of the reporting interval, or had any VARY activity during the interval. Channels for which data is not printed have one of the following messages in the first data field:

- |              |  |
|--------------|--|
| NOW ONLINE:  | Means that the channel was varied online during this interval and was online at the end of the interval.   |
| NOW OFFLINE: | Means that the channel was varied offline during this interval and was offline at the end of the interval. |
| OFFLINE:     | Means that the channel has been offline for the entire interval.   |

Figure 4-2 contains a sample channel activity report. The data fields in this report include:

- *CPU Number*: An integer (either 0 or 1), identifying the subject CPU by internal address.
- *Channel Number and Type*: The channel number is the hexadecimal representation of the external channel address. Channel types are identified as BYTE MPX, SELECTOR, or BLOCK MPX for byte multiplexor, selector, and block multiplexor, respectively.
- *Channel Activity Count*: The number of successful Start I/Os issued to the channel during the report interval. "Sense" Start I/Os are not counted; however, redundant, successful Start I/O Fast Release instructions (condition code zero) *are* counted. Six significant digits are printed. A scale factor of M (millions) may be printed after the number. The range is zero to 999,999M.



- **Percent Channel Busy:** The percentage of time during the report interval in which the channel was busy. The percentage is derived by dividing the number of sampling observations during the reporting interval in which the channel was busy, by the total number of observations during the reporting interval. An observation is made at periods determined by the CYCLE keyword. The accuracy obtainable for “percent channel busy” is dependent upon the number of observations made (see discussion under “INTERVAL and CYCLE Parameters” above). A channel is considered busy if operating in burst mode at a sampling observation. Percent channel busy values are not produced for byte multiplexor channels.
- **Percent Channel Busy and CPU Waiting:** The percentage of time during the reporting interval during which the channel was busy and, simultaneously, the CPU was in the wait state. (The CPU waiting portion of this reported value does not include the execution of the MF/1 sampling routine.). This value is used as a measure of I/O boundedness, and is calculated by dividing the number of observations during the reporting interval in which the channel was in burst mode (condition code 2 from a test channel instruction) and the CPU was simultaneously in the wait state, by the total number of observations during the reporting interval. An observation is made at periods determined by the CYCLE keyword. The accuracy obtainable for “percent channel busy and CPU waiting” is dependent upon the number of observations made (see discussion under “INTERVAL and CYCLE Parameters” above).

This value is not produced for byte multiplexor channels.

The sample channel activity report also shows the standard heading CYCLE, which indicates the requested time between sampling observations. This is the value specified by the CYCLE keyword. The range of values is 0.050 to 0.999 seconds.

C H A N N E L   A C T I V I T Y				PAGE 1	
OS/VS2 RELEASE 02.0		SYSTEM ID DE26 MF/1 VERSION 01	DATE 12/26/73 TIME 12.08.46	INTERVAL 01.00.050 CYCLE 0.250 SECONDS	
CPU AND TYPE	CHANNEL NUMBER	CHANNEL ACTIVITY COUNT	PERCENT CHANNEL BUSY	PERCENT CHAN BUSY & CPU WAIT	
0	0	BYTE MPX 4			
	1	BLOCK MPX 74			
	2	BLOCK MPX NOW ONLINE	3.32	2.49	
	3	BLOCK MPX NOW OFFLINE			
	TOTAL	78			

Figure 4-2. Channel Activity Report

D I R E C T   A C C E S S   D E V I C E   A C T I V I T Y				PAGE 1	
OS/VS2 RELEASE 02.0		SYSTEM ID DE26 MF/1 VERSION 01	DATE 12/26/73 TIME 12.07.46	INTERVAL 00.59.976 CYCLE 0.250 SECONDS	
DEVICE ADDRESS	VOLUME SERIAL	DEVICE ACTIVITY COUNT	PERCENT BUSY	AVERAGE QLENGTH	
130	KBD044	0	0.00	0.00	
150	SPOOL1	15	0.83	0.00	
152	SG2003	26	4.17	0.01	
153	SG9001	18	2.50	0.00	
230		OFFLINE			
231		OFFLINE			
	TOTAL	59			

Figure 4-3. Direct Access Device Activity Report

## I/O Device Activity Reports

The reports provide information about I/O device loading for all devices in the device classes selected by device class keywords. In addition to device activity measures, the report contains a measure of contention delay which is tabulated by device, but may be caused by hardware contention for any unit (channel, control unit, or device) in the path to the device. Device entries are included for each device which has been online at least once since MF/1 was started. Data is not formatted for devices which were offline at the end of the reporting interval, or had any VARY activity or dynamic device reconfiguration during the interval. Devices for which data is not printed have one of the following messages in the first data field:

NOW ONLINE:	Means that the device was varied online during this interval and was online at the end of the interval.
NOW OFFLINE:	Means that the device was varied offline during this interval and was offline at the end of the interval.
OFFLINE:	Means that the device has been offline for the entire interval.

*Note:* MF/1 sampling overhead is directly related to the number of devices in the device classes being monitored, whether online or not. For example, if an installation is interested in monitoring only tape devices, and specifies 'NODASD', then a large number of direct access devices does not increase MF/1 sampling overhead.

Figure 4-3 contains a sample I/O device activity report (here, the Direct Access Device Report; for the 2305 Fixed Head Storage Device, multiple device entries are made and the reported data reflects each device exposure). The data fields in the report include:

- *Device Address:* The unique three character hexadecimal device address which identifies a physical I/O device.
- *Volume Serial Number:* The six character volume serial number of the volume mounted on the device at the end of the reporting interval. This field appears only in reports for direct access and magnetic tape devices.
- *Device Activity Count:* The number of successful Start I/Os issued to the device during the report interval. "Sense" Start I/Os are not counted; however, redundant, successful (condition code zero) Start I/O Fast Release instructions *are* counted.
- *Percent Busy:* The percentage of time during the report interval in which the device was busy. The percentage is derived by dividing the number of observations during the reporting interval in which the device was busy, by the total number of observations during the reporting interval. Observations are made at periods determined by the CYCLE keyword. The accuracy obtainable for "percent busy" is dependent upon the number of observations made (see discussion under "INTERVAL and CYCLE Parameters" above).

- *Average Queue Length:* The average number of requests which are enqueued and waiting to be serviced on the subject device at any given time. This value is a measure of contention at any of the hardware units (channel, control unit, or device) in the path to the device. It is calculated by dividing the accumulation of the requests enqueued against this device at CYCLE observations, by the total number of observations made during the report interval. The accuracy obtainable for “average queue length” is dependent upon the number of observations made (see discussion under “INTERVAL and CYCLE Parameters” above).

The sample I/O device activity report also shows the standard heading CYCLE, which indicates the requested time between sampling observations. This is the value specified by the CYCLE keyword. The range of values is 0.050 to 0.999 seconds.

### Paging Activity Report

This report provides detailed information about the demands made on the system paging facilities and the utilization of real and auxiliary storage during the reporting interval. Figure 4-4 contains a sample paging activity report. Explanations of the fields appearing in the report are as follows:

#### Main Storage Paging Rates

- *Pageable System Areas:* These are areas of main storage not associated with a single address space. They include the pageable link pack area, the link pack area directory, the link pack area extension, the pageable BLDL list, and the common storage area. Virtual I/O and non-virtual I/O paging rates are provided for these areas, though the Virtual I/O counts for these areas will always be zero in MVS. (In MVS, Virtual I/O is applicable only to address spaces.)
- *Address Spaces:* These are areas of main storage associated with individual address spaces. Virtual I/O and non-virtual I/O paging rates are provided for these areas.
- *Page Reclaim Rate:* This is the per-second rate of page frames that are disconnected (stolen) from an address space or the system pageable area, but are retrieved for reuse before being re-allocated. The reclaim rate shows the number of non-initial-reference page faults or logical GETs which do not require reading from auxiliary storage. The range of values is 0.00 to 999,999.
- *Page Reclaim Percentage:* The percentage of the total system reclaim rate for each part of the total.
- *Swap Page-in Rate:* The per-second rate of pages read into real storage, as a result of address space swap-ins.
- *Non-swap Page-in Rate:* The per-second rate of pages read into real storage, exclusive of address space swap-ins.
- *Total Page-in Rate:* The per-second rate of total pages read into real storage. This rate is the sum of swap and non-swap page-in rates.

- **Total Page-in Percentage:** The percentage of the total system page-in rate, for each part of the total.
- **Swap Page-out Rate:** The per-second rate of pages written to auxiliary storage as a result of address space swap-outs.
- **Non-swap Page-out Rate:** The per-second rate of paging out of real storage, exclusive of address space swap-outs.
- **Total Page-out Rate:** The per-second rate of total pages written to auxiliary storage. This is the sum of the non-swap and swap page-out rates.
- **Total Page-out Percentages:** The percentage of the total page-out rate for each part of the total.

The ranges for all the above rates are from 0.00 to 999,999 and are expressed as an *average rate for the interval*. Explanations of the meanings of the various swapping sub-categories follow:

- **Non-VIO Page-in:** A page transfer from auxiliary to real storage which occurs as a result of a page fault, PGLOAD, or PGFIX, except those for pages of VIO data sets. Note that if there are concurrent requests for the same page, the first generates a page-in, and all the rest are considered reclaims.
- **Non-VIO Page-out:** A page transfer from real to auxiliary storage that occurs as a result of a PGOUT (including page stealing and other RSM generated page-outs), for pages other than those of a VIO data set.
- **Non-VIO Reclaim:** A request for a page as a result of a page fault, PGLOAD or PGFIX, which is satisfied without starting a new page-in, except those recovered by explicit VIO reclaim.
- **VIO Page-in:** The transfer of a VIO data set page from auxiliary to real storage, resulting from a page fault or a PGLOAD on a VIO window. VIO pages which are swapped in are not included.
- **VIO Page-out:** A transfer from real to auxiliary storage of a VIO data set page as a result of a PGOUT (including stealing and other RSM generated page-outs) on a VIO window page. VIO pages transferred as a result of a swap-out are not included.
- **VIO Reclaim:** A VIO request for a real storage data set page which was satisfied without page-in by means of the explicit VIO reclaim interface.

#### **Auxiliary Storage User Pool**

The user pool is one of three pools of paging data sets that comprise the total paging space. The other two pools are the system pool and the duplex pool. The duplex pool contains a copy of the system pool.

The following values are *snapshots of the end of the measurement interval*. These values are probably different from the average during the interval. In particular, the swap-in of the MF/1 address space will decrease the number of unused frames.

- *Page Slots*: The heading of auxiliary storage page slots for each category. Its values consist of snapshots observed at the end of the measurement interval. The range of values is from 0 to 99,999,999.
- *Available Slots*: The number of page slots that do not contain any data pages and that are available for use. (When the auxiliary storage user pool contains overflow from the system pool of paging data sets, the count of available slots will be higher than the actual number of available slots.)
- *VIO Slots*: The number of auxiliary storage page slots that contain pages for VIO data sets.
- *Non-VIO Slots*: The number of auxiliary storage page slots in the user pool that contain pages that belong to address-space virtual storage.
- *Unavailable Slots*: The number of auxiliary storage page slots that do not contain any data pages, and not available for use because of permanent I/O errors.
- *Total Slots*: The sum of all auxiliary storage page slots in the user pool of paging data sets.
- *Percent Page Slots*: The percent of total auxiliary page slots in each category.

### Pageable Main Storage Counts

The following values are *snapshots at the end of the measurement interval*. These values are probably different from the average during the interval. In particular, the swap-in of the MF/1 address space will decrease the number of unused frames.

- *Unused Frames*: The number of pageable real storage frames not allocated to any address space, or to the pageable system area.
- *Data Pages*: The number of pageable real storage frames allocated to swapped in address spaces, common area, or dynamic SQA.
- *Total Frames*: The sum of all real storage frames in the system.
- *Page Frames*: The number of real storage frames in each category. It is observed at the end of the measurement interval. The range of values is 0 to 4,096.
- *Percent Page Frames*: The percent of total real storage frames in each category.

### Swap Counts

- *Swaps*: The number of address space swap sequences, where a swap sequence consists of an address space swap out and swap in. The range of values is 0 to 9,999,999.
- *Average Pages per Swap-out*: The average number of pages swapped in for each address space swap in. The range of values is 0 to 4,096.
- *Average Pages per Swap-in*: The average number of pages swapped in for each address space swap in. The range of values is 0 to 4,096.

PAGING ACTIVITY										PAGE
OS/V52		SYSTEM ID JALL		DATE 1/11/74		INTERVAL 15.00.459				1
RELEASE 02.0		MF/1 VERSION 01		TIME 16.25.08						
MAIN STORAGE PAGING RATES										
PER SECOND										
CATEGORY	RATE	PAGE RECLAIMS			PAGE IN			PAGE OUT		
		PERCENT OF TOTAL SUM	NON SWAP	TOTAL RATE	PERCENT OF TOTAL SUM	NON SWAP	TOTAL RATE	PERCENT OF TOTAL SUM	NON SWAP	TOTAL RATE
PAGEABLE SYSTEM AREAS										
VIO	0.00	0	0.00	0.00	0	0.00	0.00	0.00	0.00	0
NON VIO	1.83	63	8.48	8.48	62	8.48	0.79	0.79	0.79	11
SUM	1.83	63	8.48	8.48	62	8.48	0.79	0.79	0.79	11
ADDRESS SPACES										
VIO	0.06	2	1.71	1.71	12	1.71	2.19	2.19	2.19	31
NON VIO	1.01	35	0.13	3.49	26	3.49	0.02	4.01	4.03	57
SUM	1.07	37	0.13	5.20	38	5.20	0.02	6.19	6.22	89
TOTAL SYSTEM										
VIO	0.06	2	1.71	1.71	12	1.71	2.19	2.19	2.19	31
NON VIO	2.84	98	0.13	11.98	88	11.98	0.02	4.80	4.82	69
SUM	2.89	100	0.13	13.68	100	13.68	0.02	6.99	7.01	100
AUXILIARY STORAGE USER POOL										
PAGEABLE MAIN STORAGE COUNTS										
SWAP COUNTS										
PAGE SLOTS		PERCENT		PAGE FRAMES		PERCENT		SWAPS		
AVAILABLE SLOTS	14,321	93	UNUSED FRAMES	145	50	SWAPS	6			
VIO SLOTS	7	0	DATA PAGES	145	50	AVERAGE PAGES PER SWAP OUT	3			
NON-VIO SLOTS	1,100	7	TOTAL FRAMES	290	100	AVERAGE PAGES PER SWAP IN	19			
UNAVAILABLE SLOTS	0	0								
TOTAL SLOTS	15,428	100								

Figure 4-4. Paging Activity Report

## Workload Activity Report

This report provides 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 installation performance specification (IPS). The report may be produced in any one of three different levels of detail, specified by the parameters indicated below from the most detailed report to the least detailed report:

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

The *performance group period report* is the most detailed workload activity report provided. It prints workload information broken down by performance group period – that is, information is provided for transactions associated with each performance group period defined in the IPS. The *performance group report* provides an intermediate level of detail. It provides information for all transactions associated with each performance group defined in the IPS. The *system summary report* is the least detailed report provided. It provides information applicable to the system as a whole during the measurement interval.

The workload activity report fields provide two sets of information at each report level:

- *Active Transactions:* Information on active transactions includes only data for transactions that were active during the reported MF/1 interval. There are four fields: Service in Interval; Average Transaction Service Rate; Workload Level; Average Transactions.
- *Ended Transactions:* Information on ended transactions includes data for transactions that may have been active in previous MF/1 intervals and for only part of the currently reported interval. Ended transactions include two fields: Ended Transactions and Average Time of Ended Transactions.

Figure 4-5 contains a sample workload activity report, at the performance group period level. Explanations of the fields appearing in the report follow:

- *Performance Group Number:* (This is provided in the performance group and performance group period level reports only.) – The identification of the installation defined performance group which defines the prescribed treatment for a class of users. The range of values is 1 to 255.
- *Performance Group Period:* (This is provided in the performance group period level report only.) – A number identifying an identifiable portion of the life of a transaction, which the installation has singled out to be associated with a particular performance objective. The range of values is 1 to 8.
- *Service in Interval:* The total number of service units absorbed by transactions during the measurement interval. The range of values is 0 to  $2^{40}$ . Only 7 significant digits are printed in the report. Higher values are shown as millions of service units by an M following the number.
- *Average Transaction Service Rate:* The total number of service units absorbed by all transactions in the period, group, or system, divided by the active time of all transactions. The entire “transaction active” time is equal to the total transaction active time accumulated by all associated transactions. Each



transaction's "transaction active" time is equal to the total time during which the transaction was in real storage, added to any swapped-out time during which the transaction was *not* in a "wait" state. It does not include time between job steps, for batch transactions. Notice that the service rate provided to *all* transactions (in the period, group, or system) is equal to the "average transaction service rate" times the "average transactions". The range of values is 0 to  $10^{13}$  minus one. Only 6 significant digits are printed. Higher values are shown as millions of service units per second, by an M following the number. (This value is not rounded; it is truncated.)

- **Workload Level:** The level in the IPS at which associated performance objectives are being satisfied. The level indicated for each performance group period is the normalized service rate for all transactions in that period. The levels given for the performance group and the system total are averages of the "period" workload levels weighted by accumulated service. The range of values is 1.00 to 224.00. The label EST is shown after the value for estimated values. An estimated value is given where the same service rate is specified in the IPS for more than one workload level – that is, when the performance objective is horizontally graphed.
- **Average Transactions:** The average number of transactions simultaneously active in the measurement interval. It is the quotient of the total active time spent by all transactions (in the performance group period/performance group/system) divided by the measurement interval time. The range of values is 0.00 to  $2^{24}$ . Only 6 significant digits are printed. Higher values are shown as millions of transactions by an M following the number.
- **Ended Transactions:** The total number of transactions which terminated during the measurement interval. The range of values is 0 to  $2^{40}$ . Only 6 significant digits are printed. Higher values are shown as millions of transactions by an M following the number.
- **Average Time of Ended Transactions:** The average transaction elapsed time of all transactions which terminated during the measurement interval. A transaction's "transaction elapsed" time is the real time interval between the start and end of the transaction. The maximum number of hours is 999.

The standard report heading appears with one additional heading:

**IPS ID:** This is the member of SYS1.PARMLIB that contains the controlling IPS (i.e., IEAIPSxx).

Data values in this report are not printed for performance groups or performance group periods which would have all values equal to zero. The word ZEROS is printed in the first field following the performance group or performance group period number.

The workload activity report may be a "short" report (it may not cover the entire reporting interval) if the Workload Manager of the SRM stopped collecting statistics as a result of a change in the IPS during the report interval. If this is the case, the INTERVAL field of the header contains the time interval during which statistics were being collected.

OS/VS2  
RELEASE 02.0

PERFORM GROUP NUMBER	PERFORM GROUP PERIOD	SERVICE IN INTERVAL	AVERAGE TRANS. SRV RATE	WORKLOAD LEVEL	AVERAGE TRANS-ACTIONS	ENDED TRANS-ACTIONS	AVERAGE TIME OF ENDED TRANS. HHH.MM.SS.TTT
001	1	118	0	50.00 EST	1.00	0	000.00.00.000
001	ALL	118	0	50.00	1.00	0	000.00.00.000
002 - 018	ZEROS						
121	1 2 - 3	102 ZEROS	7	38.64	0.05	8	000.00.01.624
121	ALL	102	7	38.64	0.05	8	000.00.01.624
122	1 2 - 3	98 ZEROS	7	38.64	0.06	8	000.00.01.748
122	ALL	98	7	38.64	0.06	8	000.00.01.748
123	1 2 - 3	95 ZEROS	9	38.25	0.04	8	000.00.01.220
123	ALL	95	9	38.25	0.04	8	000.00.01.220
124	1 2 - 3	96 ZEROS	9	38.25	0.04	8	000.00.01.287
124	ALL	96	9	38.25	0.04	8	000.00.01.287
125	1 2 - 3	70 ZEROS	8	38.44	0.03	3	000.00.02.721
125	ALL	70	8	38.44	0.03	3	000.00.02.721

Figure 4-5. Workload Activity Report (Part 1 of 2)

W O R K L O A D   A C T I V I T Y									
OS/VS2 RELEASE 02.0		SYSTEM ID JA25 MF/1 VERSION 01		DATE 1/25/74 TIME 19.42.39		INTERVAL 04.00.170 IEAIPST2		PAGE 2	
PERFORM GROUP NUMBER	PERFORM GROUP PERIOD	SERVICE IN INTERVAL	AVERAGE TRANS. SRV RATE	WORKLOAD LEVEL	AVERAGE TRANS- ACTIONS	ENDED TRANS- ACTIONS	AVERAGE TIME OF ENDED TRANS. HHH.MM.SS.TTT		
126	1 2 - 3	68 ZEROS	8	38.44	0.03	3	000.00.02.750		
126	ALL	68	8	38.44	0.03	3	000.00.02.750		
127	1 2 - 3	63 ZEROS	7	38.64	0.03	3	000.00.02.656		
127	ALL	63	7	38.64	0.03	3	000.00.02.656		
128	1 2 - 3	56 ZEROS	17	36.69	0.01	3	000.00.01.083		
128	ALL	56	17	36.69	0.01	3	000.00.01.083		
SYSTEM TOTAL		766	2	40.11	1.31	44	000.00.01.697		

Figure 4-5. Workload Activity Report (Part 2 of 2)

## Using MF/1

The uses to which MF/1 can be put are dependent upon the needs and goals of the individual installation. The following sections indicate hypothetical examples of use of each of the MF/1 report types. The examples are not meant to reflect actual figures or relationships, but rather to indicate the possibilities for tuning that MF/1 can provide. The indicated methods of analysis are not the only ones, and are not necessarily the best ones, but are suggestive of the possibilities that MF/1 provides.

### Using the CPU Activity Report

An installation wishing to monitor its CPU utilization during the afternoon hours of 1:00 to 3:00 (its peak period) requests the collection of CPU usage data by having the operator enter the following command at 1:00 P.M.

```
START MF1.EXAMPLE, , ,(NOPAGING,NOCHAN,NOWKLD,
NODEVICE, INTERVAL(30M), STOP(2H))
```

Assuming that the installation is running with the IBM-supplied MF/1 cataloged procedure and the IBM-supplied IRBMF100 parmlib member, the following options will control the MF/1 execution:

```
CPU
NOCHAN
NODEVICE
NOPAGING
NOWKLD
CYCLE(250)
INTERVAL (30M)
MEMBER (00)
OPTIONS
NORECORD
REPORT(DEFER)
STOP(2H)
SYSOUT(A)
```

The CYCLE Parameter will have no effect upon execution, since neither channel nor device statistics are being gathered.

This list of options will result in 4 CPU activity reports being printed after MF/1 termination, based upon statistics gathered each half-hour for two hours.

The installation finds, from averaging CPU wait time percentages, that the CPU is in the wait state 30% of the time during this peak period. Determining that this is unacceptable, the installation creates a new PARMLIB member IEAOPTxx and specifies the CPU resource factor coefficient (RFC) in this new member as:

```
CPU = 9.9
```

Since previously the value of this coefficient was equal to 0.1, the installation reasons that this change will cause the SRM to weight CPU imbalances more heavily in making swapping decisions. The installation places the new member into

operation, and observes the effect upon the system by requesting a full complement of MF/1 reports for a subsequent 1:00 to 3:00 peak period. To obtain these reports, the operator enters:

```
START MF1.EXAMPLE, , ,(INTERVAL(30M),STOP(2H))
```

This execution produces 4 reports each of CPU, channel, device, paging, and workload activity. Analyzing the CPU activity report, the installation finds that CPU utilization has been raised to 100%. However, analyzing the workload activity report (see Workload Activity Report, below), the installation finds that the service rate provided to a high priority group of users has been seriously degraded. The installation makes the inference that this resulted from the RFC change. Therefore, the installation decreases the CPU resource factor coefficient as follows:

```
CPU = 5.0
```

After placing the modified IEAOPTxx member into operation and monitoring during a subsequent peak period, the installation finds that the service rate provided to the high priority performance group is at an acceptable level. Furthermore the CPU utilization, though not so high as at the last monitoring, is still substantially above the original level.

### Using the Channel Activity Report

An installation wishes to analyze the utilization of its I/O channels. It decides to try to obtain about 2400 samples as a basis for the printed figures in each printed channel report. To achieve this number of samples in each printed report, a CYCLE value of 500 milliseconds is chosen (2 samples per second), and a reporting INTERVAL length of 20 minutes (1200 seconds, 2400 samples) is selected. (See discussion under "Interval and Cycle Parameters", above.) The system operator enters the following command to initiate MF/1 operation, at four different periods during the day:

```
START MF1.EXAMPLE, , ,(NOCPU,NOPAGING,NOWKLD,  
NODEVICE, INTERVAL(20M),CYCLE(500),STOP(1H))
```

Assuming that the installation is running with the IBM-supplied MF/1 cataloged procedure and IRBMF100 parmlib member, the following options will control the MF/1 operation:

```
NOCPU  
CHAN  
NODEVICE  
NOPAGING  
NOWKLD  
CYCLE(500)  
INTERVAL(20M)  
MEMBER(00)  
OPTION  
NORECORD  
REPORT(DEFER)  
STOP(1H)  
SYSOUT(A)
```

This list of options will result in three channel activity reports being printed after MF/1 termination, based upon statistics gathered every twenty minutes for one hour.

The installation finds, in comparing the *percent channel busy* indications from each channel, during each of the four one-hour periods of MF/1 activity, that one particular channel (channel 2) has a continually significantly higher percent channel busy figure than the other channels. The installation attempts to correct this situation by increasing the value of the I/O resource factor coefficient in the parmlib system tuning parameters member (parmlib member IEAOPTxx) as follows:

| IOC = 5.0 (previously, the value was 0.1).

After an IPL with the modified IEAOPTxx member, the installation monitors the effect of the change upon the system by again requesting that the operator enter the previous command four times during the day, each time obtaining three printed channel activity reports.

Again, the installation finds that the utilization of channel 2 is too high. This indicates that the SRM has been unable to improve the imbalance by its swapping recommendations. To investigate the problem further, the installation requests device activity reports, as explained below.

### Using the Device Activity Report

In pursuing the problem outlined under "Channel Activity Report" the installation notes that both tape and direct access devices are connected to channel 2. Therefore, the installation requests that the operator enter the following START MF/1 command a number of times, to analyze the device activity on this channel:

```
START MF1.EXAMPLE, , ,(NOCPU,NOCHAN,DEVICE(NOUNITR,
NOCOMM,NOGRAPH,NOCHRDR) NOPAGING,NOWKLD,
INTERVAL(20M),CYCLE(500),STOP (1H))
```

Assuming IBM supplied MF/1 cataloged procedure and IRBMF100 parmlib member, the following options will control the MF/1 operation:

```
NOCPU
NOCHAN
DEVICE (NOCHRDR,NOCOMM,DASD,NOGRAPH,TAPE,NOUNITR)
NOPAGING
NOWKLD
CYCLE(500)
INTERVAL (20M)
MEMBER (00)
OPTION
NORECORD
REPORT (DEFER)
STOP(1H)
SYSOUT (A)
```

This list of options will result in three tape device activity and three direct access device activity reports being printed after MF/1 termination, based upon statistics gathered every twenty minutes for 1 hour.

The installation finds that two of the direct access devices on channel 2 both have high percent busy figures, and high average queue length figures. The installation checks the volumes indicated as mounted on these devices, and finds several frequently used system data sets on each. After relocating some of these data sets, subsequent MF/1 analysis shows a better balance of I/O device and channel utilization.

### Using the Paging Activity Report

This report provides a wide breadth of information not susceptible to simple interpretation. An installation could use it, for example, to analyze the amount of swapping overhead that is occurring. It could also analyze the change to this overhead that is produced when certain System Resources Manager parameters are varied (for example, the *interval service value* (ISV) in the IPS). The installation might similarly wish to analyze the percentage of auxiliary storage slots that are unused, in view of the relation of this percentage to the efficiency of operation of the Auxiliary Storage Manager. (A low percentage of unused slots could result in an excessive amount of ASM overhead, in trying to access these unused slots.) Alternatively, the installation might wish to compare paging rates indicated by this report with the profile of user activity provided by the Workload Activity Report, to analyze possible correlations between high paging rates and high percentages of active users from certain performance groups.

## Using the Workload Activity Report

This report, like the paging report, is susceptible to a wide range of uses. Some examples follow:

- The distribution of transactions among performance groups is found under the column labeled “average transactions” and in the row labeled “ALL” in the performance group period column. The distribution can give a general profile of the system workload at different times of the day. In contrast, the distribution of total service among performance groups (found under the “service in interval” column, and in the row labeled “ALL” in the performance group period column) can tell the installation which performance groups are really getting the bulk of the system resources.
- The total service provided to all users in the measurement interval is found under the “service in interval” column and in the “system total” row. This information can provide a basis for comparing system throughput under different workload conditions. By comparing this figure for similar MF/1 reporting interval lengths, the installation can determine whether changes made to improve the system configuration have had the desired effect on throughput.



**This blank leaf represents page 229 which was deleted by Supervisor Performance #2.**



## Part 5: System Performance Factors

Part 5 contains discussions of factors that affect system performance and discussions of certain tools that can measure performance. Each discussion includes guidelines, and rationale. Some discussions also include questions and answers. Since MVS is significantly different from MVT, the discussions emphasize new aspects of the system, such as VIO, VSAM catalog, and the pageable link pack area. The performance information is based on design analysis; that is, on projections of how the system is supposed to work. Some of these ideas will change as system experience is gained. These new insights, based on system measurements, will first appear in Installation Newsletters as tuning guidelines.

The following performance topics are included in Part 5, in this order:

- The Pageable Link Pack Area: Its Advantages and Uses
- VIO Performance
- Device Allocation Performance
- VSAM Catalog Performance
- How SMF Can Supplement MF/1
- The Use of GTF to Track Sysevents
- TCAM Tuning Considerations
- TSO and Batch Service Trade-offs Via the IPS
- Miscellaneous Performance Guidelines

The information previously here and through page 239 in the base publication is now incorporated in “Part 2.1: Auxiliary Storage Management Initialization”.

**This blank leaf represents pages 232 - 239 which were deleted by Supervisor Performance #2.**



## The Pageable Link Pack Area: Its Advantages and Uses

The pageable link pack area (PLPA) contains most of the frequently used control program routines. All IBM-supplied transient SVC routines and appendages now reside in the PLPA. The result is that the transient area contention problem of MVT has been eliminated.

It's desirable to place all commonly used reentrant LINKLIB and CMDLIB modules\* in the PLPA because of the following advantages. (An opposing philosophy on the use of the *fixed* LPA is described later in this topic.)

- The length of time that a page occupies real storage depends on its frequency of use. If the page is not used over a period of time, the system will reuse (steal) the real storage frame that the page occupies.
- The most frequently used PLPA modules in any given time period will tend to remain in real storage.
- PLPA paged-in modules avoid Program Fetch overhead.
- Two or more programs that need the same PLPA module share the common PLPA code, thus reducing the demand for real storage.
- The main cost of unused PLPA modules is paging space, since only auxiliary storage is involved when modules are not being used.
- All modules in the PLPA are treated as reentrant in MVS; that is, PLPA pages in real storage are recognized by the system, the "change" bit is ignored, and PLPA page-out is avoided. This action reduces the overall paging rate compared with modules in other libraries.

### Recommendations

The following recommendations should improve LPA performance:

- Copy frequently and moderately used reentrant\*\* modules from LINKLIB and CMDLIB to LPALIB. *OS/VS2 System Programming Library: Storage Estimates* lists some of the control program modules that are already packaged in LPALIB and those that may be placed there. Low-usage TSO command processors and low-usage user programs probably should not be placed in LPALIB. The reasoning is as follows.

---

\*See an alternative suggestion on the placement of some PLPA modules and CMDLIB modules later in this topic.

\*\*Preferably the modules should be refreshable, since a reentrant module is allowed to modify itself if it holds a global (cross-memory) lock during the modification.

## Pageable Link Pack Area (continued)

When code in the PLPA is not continually referenced by multiple address spaces, it will tend to be stolen. The reason is that individual address spaces typically get swapped out (without the PLPA pages they reference) and, as a result, one address space can't maintain the referencing frequency necessary to prevent page stealing. When an address space is swapped back in, and most of the code it references has been stolen, the result is a very time-consuming serial sequence of page faults.

- An alternative philosophy is to allow low-usage CMDLIB and low-usage user modules to remain in their libraries and be fetched into the user's subpools, instead of being placed in the PLPA. The result would be that the TSO commands and user code would be swapped in and out with the address space. This action would virtually eliminate the serial page-fault delays. A secondary advantage is that the SRM could more accurately estimate the real storage implications of swapping such an address space into or out of real storage.
- Minimize page faults and disk arm movement by specifying module packing through the IEAPAK00 member of parmlib. Module packing reduces page faults by placing in the same virtual page those small modules (less than 4K bytes) that refer to each other frequently. In addition, module groups that frequently refer to each other but which exceed 4K bytes in combined size can be placed in adjacent (4K) auxiliary storage slots to reduce seek time.\* Thus, use of the PAK list should improve performance compared with the simple loading of the PLPA from LPALIB. (See the description of parmlib member IEAPAK00 in the Initialization chapter. The IEAPAK00 description includes a complete list of module names and their functions that are included in the IBM-supplied member.)

If your installation uses ISAM, you may wish to add the following six pack-groups to parmlib member IEAPAK00. Use the syntax rules listed under that member in the Initialization chapter.

```
(IGG0192A,IGG0192B,IGG0192C),  
(IGG0192F,IGG0192G,IGG0195T,IGG0195U),  
(IGG0196C,IGG0196D,IGG0195G,IGG0196G,IGG0195D),  
(IGG01928,IGG01929,IGG01924),  
(IGG0202I,IGG0202N,IGG0202J),  
(IGG0202K,IGG0202L,IGG0202M)
```

- Use within reason those parmlib lists that speed the search for listed modules at the expense of the unlisted modules. The lists include the modified link pack area list (IEALPAxx), the fixed LPA list (IEAFIXxx), and the load list (IEALOD00). (For information on these lists, see IEALPAxx, IEAFIXxx, and IEALOD00 in the Initialization chapter.) NIP builds and chains contents directory entries (CDES) for modules in these lists, one CDE per module.

---

\*Adjacent slot location, although likely, is not guaranteed. For further information, see "How the PLPA Is Loaded" later in this topic.

## Pageable Link Pack Area (continued)

Later, when a module is requested, the Program Manager serially searches the active LPA. The active LPA is ordered as follows:

1. modules loaded from PLPA
2. IEALOD00 list
3. fixed link pack area list
4. modified link pack area list

**Note:** This sequence is a part of the overall module search sequence listed later in this topic under the heading “The VS2 Module Search Sequence”.

If the CDE search is unsuccessful, the Program Manager then uses a hashing technique to search the LPA directory on direct access. Note that modules in the modified LPA list, the fixed LPA list, and the IEALOD00 list tend to be found fairly rapidly (except for very long lists), but at the expense of modules that *must* be found in the LPA directory. It's undesirable to make these parmlib lists excessively long, since the CDE search will be prolonged and the LPA directory search will be delayed. It's probably best to limit the three lists to moderate-use modules, since high usage PLPA modules will tend to remain in real storage and thus in the active link pack area list.

- You may use the *fixed* LPA to buy reduced page-fault overhead and increased performance at the expense of some real storage. Such tradeoff would be desirable with a system that tends to be CPU bound. The condition of being CPU bound implies that real storage is not also overcommitted. Therefore, it would seem desirable to divert some real storage from possible use by additional address spaces, and use the diverted storage for LPA module residence. (You can run MF/1 to determine the relative availability of CPU.) Implement the tradeoff by placing moderate-usage LPALIB modules in the fixed LPA (via parmlib member IEAFIXxx), instead of in the pageable LPA. High usage PLPA modules probably need not be listed in IEAFIXxx, since they may be referenced frequently enough to remain in real storage anyway. A large fixed LPA means less real storage will be available for pageable programs. Accordingly, the System Resource Manager will maintain fewer address spaces in real core than would otherwise be the case. No loss in throughput should occur, however, as long as CPU utilization remains reasonably high.

Note that this suggestion represents an opposite philosophy to that described earlier, in which *all* moderate-use reentrant modules would be placed in LPALIB.

The STIMER and TTIMER modules (IGC0004F and IGC0004G, respectively) are especially good ones to put in the fixed LPA because their design takes advantage of that placement to significantly improve the performance of the timer function. To place the modules in the fixed LPA, include their names in the IEAFIXxx member.



## Pageable Link Pack Area (continued)

### The VS2 Module Search Sequence

The Program Manager searches for a requested module (requested in EP\* form without DCB) in the following sequence. Note that the search of the active LPA queue in real storage takes place before the search of the PLPA directory.

1. job pack area queue (JPAQ)
2. tasklib(s), steplib, joblib
3. active CDE queue (via a serial search). This queue includes:
  - a. modules loaded from PLPA
  - b. IEALOD00 list
  - c. fixed LPA (FLPA) list
  - d. LPA extension (also called the modified LPA, or MLPA)
4. pageable LPA (via a hashed search of the LPA directory)
5. LINKLIB and libraries concatenated to it via LINKLSTxx members

### How the PLPA Is Loaded

The Program Manager's resource initialization module (or RIM) "loads" PLPA from high virtual addresses to low. It starts with the pack groups listed in the IEAPAK00 list, continues with the nonpack modules in LPALIB, and finishes with the link pack directory.

The size of each pack group is determined, and each page is filled from low address to high. There is never any overlap of pack groups; that is, the second of two pack groups always starts at the next page boundary, possibly leaving a gap that the RIM will try to fill later with small non-pack modules.

---

\*EP means the entry point parameter of an ATTACH, LINK, LOAD, or XCTL macro.

## Pageable Link Pack Area (continued)

The LPALIB modules that are not listed in IEAPAK00 are loaded next, from high virtual address to low, starting with the modules that are larger than 4K. The modules sized between 3K and 4K are then loaded. These modules may either fill 3K–4K gaps within a previously loaded pack group or be placed on a new virtual page. Lastly, the smaller modules are loaded in descending size order to try to fill gaps in the loaded pack groups. In each case the search is for a gap greater than each size value in bytes, as follows:

4096  
3584  
3072  
2560  
2048  
1536  
1024  
512  
256  
128  
64  
0

After all modules have been loaded, the link pack directory is built.

During the load process, the Program Manager RIM forces page-outs of 100K bytes at a time. The Auxiliary Storage Manager (ASM) does the actual loading of the paging data set. (Real frame stealing may interfere with this process.) The forced-out pages will probably be placed in contiguous auxiliary storage slots, from low slot address to high, on the page data set(s) that will hold the PLPA\*.

Figure 5-1 illustrates the relative positions of two pack groups, composed of 20 module each, plus 360 LPA modules (before backfilling) that are not in pack groups, and the LPA directory. The figure shows the relative positions of the modules and the LPA directory in virtual storage and in the PLPA page data set.

---

\*Ideally, the PLPA should be placed on a single page data set on a single device. This will happen if the first-named page data set (specified at IPL or sysgen) is sufficiently large to hold the entire PLPA. If it is not, the overflow will go on another page data set, possibly on another device. Since the LPA directory is paged out with the last 100K block, such PLPA separation could degrade performance, particularly if the overflow device is slower than the primary device.

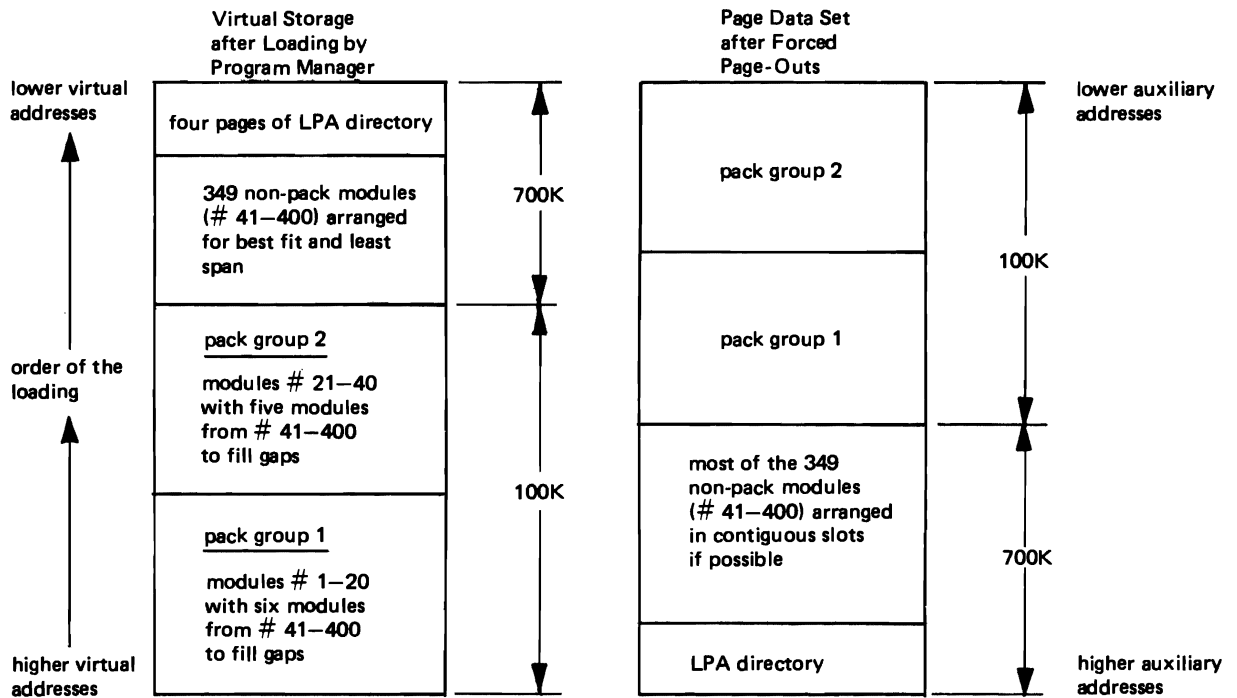
## Pageable Link Pack Area (continued)

### SYS1.LPALIB

Contains: modules 1–400

### SYS1.PARMLIB (IEAPAK00)

Contains: Pack Group 1 Names (modules 1–20)  
Pack Group 2 Names (modules 21–40)



**Note:** Modules are transferred to auxiliary storage in blocks of 100K bytes.

Figure 5-1. Illustration of the Loading of the PLPA

## VIO Performance

The use of VIO has both advantages and disadvantages. The installation must decide whether the advantages outweigh the disadvantages for its particular applications.

### Advantages

VIO offers the following advantages:

- Virtual I/O pages tend to stay in real storage, if pages are not frequently stolen without being reclaimed. This gives VIO at times the effect of a very large number of buffers. With real data sets, on the other hand, output is done immediately, and on input, buffers are reused.
- DADSM overhead is minimized.
- VIO can improve the CPU efficiency of a user program that has a relatively small blocking factor. With a simulated 3330, with 9 blocks or more per track, VIO uses fewer instructions per track than conventional I/O. With fewer than 9 blocks per track, conventional I/O may be more efficient than VIO, provided that VIO data does not remain in real storage and real I/O occurs to a paging data set.
- The conventional serialization required by Allocation is not required for VIO (see performance topic “Device Allocation”).
- The use of paging algorithms, such as ASM’s slot sorting, should speed up VIO operation, compared with conventional I/O (EXCP) by reducing device delays. For output, simulated data transfer can occur without waiting for real I/O to occur to paging space.
- VIO can facilitate improved DASD space management by the installation. If all temporary data sets use VIO, the installation can add to the paging space all of its DASD space previously used for temporary data sets. This should give the system at least as much VIO space as required. When VIO usage is low and other paging requirements are high, the space will automatically be used for non-VIO system paging needs.

## VIO Performance (continued)

Since auxiliary slots for VIO are allocated by ASM only as they are required (when the window is to be paged out) it may be possible to commit less DASD space for temporary data sets than with conventional I/O. Normally, a programmer allocates DASD space to handle a peak data set size, when in fact only a small portion of that space may be used. The space, however, remains reserved and cannot be used for other purposes. By using VIO, the programmer may still purposely overestimate data set size to insure he will not run out. However, he can be certain that only the space needed for data at any moment will be allocated.

By examining slot usage (through the MF/1 Paging Activity report), the installation may be able to remove some of the previously “wasted” DASD space. It would then be able to utilize the space for other purposes, and still support as many temporary data sets as it did previously. (See Part 4: How to Use the System Activity Measurement Facility, Paging Activity Report, for additional information. In using MF/1 data for this purpose, the user should note that the Auxiliary Storage User Pool portion of the MF/1 Paging Activity Report indicates a snapshot of slot usage at the *end* of the report interval. Slot usage is likely to vary during a report interval.)

- The JCL SPACE parameter is enforced for VIO, as it is for conventional access methods. There is also a default space allocation for VIO if the parameter is not specified. The fact that the SPACE parameter (specified or defaulted) is enforced gives the installation some protection against program bugs which could cause a write loop and saturate the paging space. Note that the maximum possible size for one VIO data set is a single volume on the simulated device.
- VIO alleviates the SCRATCH SYS Problem. If an abnormal system termination occurs, there are problems associated with the deallocation of existing real system-named temporaries. For a batch installation, a warm start must be done (with the same packs up) to cause deallocation. Otherwise, SCRATCH SYS must be run on all the packs containing the real temporary data sets. For TSO system-named temporaries, SCRATCH SYS is the only good way to deallocate real temporaries, since restart is not supported.

With VIO, deallocation of temporaries requires only a re-IPL that specifies CVIO (Clear VIO) at warm start or at cold start. All auxiliary slots containing VIO data set pages will be freed regardless of whether or not the volumes on which they reside are mounted.

- VIO is physical device independent. That is, under control of the paging I/O algorithms, the physical device chosen to contain the data set has no relation to the device being simulated. The device chosen will be the best one for the system at the time the I/O is required. No matter how device-dependent the user's code is, VIO will remain device independent.

## VIO Performance (continued)

### Disadvantages

These appear to be the disadvantages of VIO:

- VIO for large data sets (particularly if not in update processing) may be less efficient than conventional I/O. Pages from large VIO data sets will be readily stolen and the data forced out to paging space.
- VIO is less efficient than conventional I/O for physical data transfer, if the simulated block size is large, i.e., fewer than 9 blocks per track on a 3330. For example, on a 3330 VIO uses slightly fewer instructions to read or write a full track than does EXCP at 9 blocks per track, but with quartertrack blocking VIO requires approximately twice as many CPU instructions per track as does EXCP.
- The system data set SYS1.STGINDEX saves the page data set status of jobs that are eligible for automatic checkpoint or step restart. If SYS1.STGINDEX fills up, the job step can not be restarted.

### VIO Performance Considerations

To optimize VIO performance, try to make use of these performance considerations:

- VIO is most efficient for small data sets used for update processing. Such data sets may refer to data pages frequently enough to allow page reclaim. Page reclaim consists of the reuse of a page of data in real storage before it can be written out to paging space. You can measure the relative success of page reclaims by running MF/1, and noting the number of VIO page reclaims listed in the Paging Activity report.
- Allocate VIO data sets by cylinder to minimize overhead. VIO simulates the physical device by software coding. I/O requests are checked for validity against the data set extents. For data sets allocated by cylinders, a check is made for SEEK or SEEK CYLINDER channel command codes. However, for block allocation (the default) or for track allocation, all channel command codes that can cause a reference to any track other than the current must be validity checked which causes additional overhead. This includes SEEK, SEEK CYLINDER, SEEK HEAD, all multi-track commands, overflow READS, etc.
- Use VIO for blocks of data 1200 bytes or less, and IOS for blocks of 3200 bytes or more. Measurements have shown that VIO uses fewer CPU instructions to process a 1200 byte (or less) block of data than IOS; IOS uses fewer instructions than VIO to process a 3200 byte (or more) block of data. For the range of block sizes between 1200 and 3200, VIO should be used if the system has light paging activity.

## VIO Performance (continued)

- It's best to choose a block size whose multiple (plus block headers and window control record) almost completely fill the pages that comprise the window. The VIO "window" that is written out to paging space is always equal to the *used* portion of the track size (data plus track overhead) rounded up to the next multiple of 4K. Although the window, when full, is always four pages for a 3330, the amount of data transferred depends on the number of bytes of the data on the simulated track. Avoid a block size that requires an additional page for the window without significantly increasing the data being transferred. For example, a block size of 2035 uses a three-page window to transfer 12,210 bytes of useful data, plus 8 bytes for each block header and 30 bytes for a window control record, at six blocks per track. (This is a total of 12,288 bytes.) In contrast, a block size of 2036 bytes uses a four-page window to transfer 12,216 bytes of useful data, plus 8-byte block headers and a 30-byte window control record, for a total of 12,294 bytes, also at six blocks per track. However, the 2036-byte block size needs an extra page in the window, consisting of 4090 wasted bytes, to transfer 6 additional bytes of useful data.
- VIO data sets ought not be allocated and held for long time sharing sessions. Since temporary data sets are usually freed only at the end of a TSO session, the total paging space in use can become excessive. The installation can control the total allocated VIO space per TSO session, possibly by means of a DAIR exit.
- Consider chained scheduling for non-VIO data sets. The use of OPTCD=C, together with an increase in the number of buffers (for example, BUFNO=6), has been found to significantly reduce system instructions in some batch environments.

## How to Specify VIO Data Sets

The following are the requirements for specifying VIO data sets:

- The UNITNAME macro at sysgen must specify VIO=YES. Both generic and esoteric names are accepted. With an esoteric name, the first unit in the list must be a direct access device. A generic name must be DASD and should not specify a unit list.
- The data set must be specified or defaulted as temporary (NEW,DELETE or NEW,PASS).
- The volume request must be non-specific (i.e., no volume serial number).
- The total SPACE parameter – primary and used secondary – can not exceed one volume on the device being simulated. The primary space specification will be accepted up to a value of one volume on the simulated device. The default space value is (1000,(10,50)).
- The data set should have no dsname or should be specified by &name.
- VIO cannot be used for VSAM or ISAM data sets.
- The unit count subparameter of the DD UNIT parameter is ignored.

## VIO Performance (continued)

### Questions and Answers

The following questions are those that customers have asked regarding VIO.

*Q: What is the effect of VIO usage when going from a paging load of light to heavy? Is there a mechanism or procedure to offset any negative effect?*

*A: The effect consists of possibly increased I/O and CPU overhead. Both factors slow VIO performance. However, CPU overhead may not increase proportionately with increased paging, since the Auxiliary Storage Manager may be able to chain more pages together for I/O. One answer would be to provide more paging devices, faster paging devices, and different channel paths. Another solution would be to obtain more real storage.*

*Q: Are there any advantages in simulating a 2314 for VIO data sets if the paging device is a 3330 or 2305, even though no 2314s may actually be in the system?*

*A: Yes, if you have code that depends on the device being a 2314. Be sure that an IODEVICE macro at sysgen specified a 2314, at a unique address, even though there are no physical 2314s on the system.*

*Q: Does VIO fill the window for an input data set as an anticipatory paging operation?*

*A: No. The window is filled only as the result of input requests.*

*Q: The track capacity of a simulated 3330 for VIO requires a four-page window. However, if I block my data 1680 bytes per block, the seven blocks will fit in the first three pages of the window. Are all four pages paged out?*

*A: No. Only those pages that have data are paged out.*

*Q: Can a job that runs ADDRSPC=REAL use VIO?*

*A: Yes.*



## VIO Performance (continued)

*Q: If paging error occurs with a VIO data set, does a user SYNAD exit get control?*

*A: No. (For discussion of paging error handling, see the Questions and Answers section in the topic "Guidelines for the Use of Paging Data Sets".)*

*Q: Is there a limit to the size of a VIO data set? If so, how is it specified?*

*A: The size of a VIO data set can be limited via the JCL SPACE parameter in the same manner as with non-VIO data sets. The maximum size of a VIO data set is one volume of the device type being simulated.*

*Q: Do you have one window per active VIO data set or one per address space?*

*A: One per data set. This applies no matter how many DCBs the user may have open to the data set.*

*Q: What happens when the user buffer size is greater than VIO window size?*

*A: A buffer size (block size) that is larger than the window size can be valid only if the track overflow feature is being simulated, since the window size is equal to one track of the device being simulated, rounded up to the next 4K. If track overflow is not being simulated, an error indication (blocksize larger than physical track size) will be returned.*

*Q: Can UNITNAME macro at sysgen contain VIO=YES for a device not installed at the installation?*

*A: A device can be simulated that is specified at sysgen, via both the UNITNAME and IODEVICE macros, but which is not physically online to the system.*

*Q: Does PCI work with VIO?*

*A: Yes. Existing programs that use the PCI feature will work with VIO. However, the user should understand that the I/O is being simulated, and the PCI interrupts and associated processing have no relation to the real I/O to the paging data sets.*

*Q: Is the window block-paged or individually paged?*

*A: The window is block paged on output but is individually paged on input, via a page fault.*

## Device Allocation Performance

This topic discusses how an installation can improve performance through the allocation of devices, data sets, and volumes. The first section describes the order in which allocation requests are serviced. The remainder of the topic provides guidelines for improving allocation response.

**Note:** For SRM influences on Device Allocation, see “Device Allocation” in the chapter “The System Resources Manager”. For Auxiliary Storage Manager influences on the selection of paging devices, see “Part 2.1: Auxiliary Storage Management Initialization.”

### The Order in Which Allocation Requests Are Serviced

The order in which allocation requests are handled by the system affects the processing time and the degree of serialization of particular allocations. To reduce serialization, allocate your data sets, volumes, and devices from the categories high on the following list, if possible. As you move down the list, the degree of serialization and the processing time increase.

1. VIO data sets, JES2 or JES3 data sets, and dummy data sets.
2. Permanently resident or reserved direct-access volumes (see the VATLSTxx description in the Initialization chapter for information on the specification of these volumes).
3. Teleprocessing devices; and generic device types as specified in the device precedence list, except devices which hold permanently resident or reserved DASD volumes. (See the guideline later in this topic on the use of the DEVPREF keyword of the SCHEDULR macro to specify the device precedence list.)
4. Unmounted nonspecific direct access volumes. These requests cause serialization with other allocations until the operator mounts the volumes.
5. Offline devices and devices allocated to another job. These requests require operator interaction.

## Device Allocation Performance (continued)

### Guidelines for Improving Allocation Response

The following suggestions should help your installation to make best use of the redesigned Device Allocation routines:

- Within the limit of page space availability, encourage the use of VIO data sets. (For further information, see the topic entitled “VIO Performance”.)
- Set up a sufficient number of permanently resident and reserved DASD volumes on line, to avoid contention for a few volumes of these types. You can check for contention by running MF/1 to obtain device activity reports. The volumes should be spread across channels so that the System Resource Manager can balance the channel load.
- Use the UNITNAME sysgen macro to define separate esoteric subgroups within major generic device types, so that different subsets of users can request separate subgroups of devices. The purpose is to minimize contention for the same devices among the various subsets of users. For example, an installation whose batch and time sharing users request allocation of 3330's could separate the two types of user requests as follows:

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

The effect of this specification is that allocations to SYSBATCH serialize only requests for units 330–333, instead of the entire 3330 generic. Similarly, allocations to SYSTSO serialize only requests for units 334–337.

- Use the DEVPREF keyword of the SCHEDULR sysgen macro to minimize contention for the fastest devices. The DEVPREF keyword sets up the device preference table. The table determines the order in which device types will be selected by Allocation if a request is eligible for more than one device type (e.g., UNIT=SYSDA). If the keyword is not specified, the default device preference table lists the fastest generics first. If such specification causes heavy contention for the fastest eligible devices, you can specify the DEVPREF list so that generics with many devices (and many channels) are listed first and are therefore given preference. As a secondary consideration, the increased number of preferred units and channels will give the System Resource Manager a large selection for its choices.
- Keep all operable devices online if possible. (This is old advice and does not depend on the redesign of Device Allocation.)
- Try to avoid the use of specific unit address (e.g., UNIT=253) in DD statements for volumes that are neither permanently resident nor reserved. A specification of specific unit address serializes the request on the entire device type. For example, if unit 253 is a 3330, a specific unit request (UNIT=253) will be serialized with other requests for any 3330. Instead of using specific unit address, use subsets of the generic device type, as suggested earlier in this topic.

## Device Allocation Performance (continued)

- Resolve the question whether the operator should respond HOLD or NOHOLD when a job must wait for other jobs to free devices or volumes, and a message is issued to the operator. The criteria for resolving the question are:

**HOLD** This means that the job should wait while holding devices and volumes already allocated to the job. Select this option if the needed resources are constantly being freed, and allocation requests for other jobs will probably not be held up by the requests made for this job. This job can hold up other requests in either of two ways: it has already allocated units needed for another job, or its allocation requests are serialized on devices it is waiting for.

**NOHOLD** This means that the job waits without holding devices and volumes already allocated to the job. Select this option if the needed resources may not be freed for some time, and allocation requests for this job are likely to hold up requests issued for other jobs.

*Note:* Requests for dynamic allocation are not held up by requests waiting for batch allocation, even though the jobs awaiting batch allocation are holding resources.

- Free data sets, volumes, or devices before the end of a job step or TSO session, if possible. The freed resources can then be used for other jobs or sessions. You can free the resource when a data set is closed, by specifying FREE=CLOSE on the associated DD statement. (This option is a new facility in MVS.) Note that when subsequent steps of a job require the same data set, the resource must be reallocated prior to being reaccessed (or else the OPEN fails). Use discretion when freeing the resources, however, because once a resource is freed, its continuing availability cannot be guaranteed.
- Invoke Dynamic Allocation from a batch job by means of a new application of SVC 99. (The details are described in *OS/VS2 System Programming Library: Job Management*.) The advantage is that the batch job allocates the resource only when it is needed, and frees it as soon as it is not needed. (FREE=CLOSE can also free the resource, if it is specified on a DD statement.) Resources are thus more readily available to other requesting jobs. A disadvantage is that the batch job must handle a return code if the requested resource is not available. (With conventional allocation via DD statements, the system would cause the job to wait for the requested resource(s) to become available.) Note, however, that an authorized program need not handle a return code if a requested resource is not available. The authorized program can request a “wait for the resource” when it invokes Dynamic Allocation. (Unfortunately, there is no deadlock detection in this case.)

### Device Allocation Performance (continued)

- Premount all private volumes, including private catalogs, before running the jobs that request these volumes. (This is another piece of old advice.) Note, however, that AVR (Automatic Volume Recognition) is no longer optional.

### TSO Allocation Suggestions

The following suggestions should improve TSO allocations during TSO sessions although they may extend log-on times:

- DD statements that a user wants in all his TSO sessions should be placed in a LOGON procedure. This technique has these advantages:
  1. Allows volumes to be mounted.
  2. Provides recovery from an offline device condition. Messages tell the operator to VARY the device online.
  3. Saves repeated allocation and freeing of the same data set by successive commands in the same TSO session.
- The DYNAMNBR parameter value in the EXEC statement should be carefully chosen. The value should be large enough so that it is not readily exceeded by dynamic allocation requests. Note that the maximum number of concurrently allocated resources for any TSO session is 1635.

The topic “How the SRM Allocation Algorithm Affects I/O Load Adjusting”, formerly here, is now rendered obsolete by the SRM redesign.

## Device Allocation Performance (continued)

In practice, it is not always possible to restrict a job's I/O dependencies to a single channel. However, any reduction in the job's I/O dependencies will improve the job as a candidate for correcting I/O load imbalances. Accordingly, the SRM Device Allocation algorithm selects channels with the objective of minimizing a job's logical channel dependencies.

The same strategy should be followed by job submitters. This is particularly important in the case of jobs which do a considerable amount of I/O over a sustained period of time. It is of course important that not all jobs choose their data sets through the same logical channel(s). This suggests some understanding between the installation and its heavy I/O users, supplemented by the use of job classes. The job classes must ensure a good distribution of jobs with different logical channel dependencies.

An important secondary benefit is realized from this allocation strategy. When a particular channel has an uncorrectable overload, every address space which depends on this channel has its processing slowed by the slow channel response. If many of the executing address spaces are slowed by such a dependency, the result is a significant drop in throughput, accompanied by a drop in the utilization of other system resources. This drop occurs because many address spaces spend much time waiting on the overloaded channel. If, however, some jobs do not depend on the overloaded channel, they will not be slowed by the overload, and they will be able to take advantage of the lowered demand for other resources. Consequently they will run faster than would normally be the case.

## Questions and Answers

The following Allocation question has been asked by customers:

- Q: The Device Allocation algorithm in the System Resources Manager tries to minimize the number of logical channels used by a batch job or TSO session. What happens if a job's execution time depends on channel separation?*
- A: Use specific unit allocation or an esoteric name eligible to a single channel.*

## VSAM Catalog Performance

*Note:* These guidelines pertain to VSAM catalogs only, not to OS CVOLs. For information on OS CVOL usage under the VSAM master catalog, refer to *OS/VS2 Using OS Catalog Management With the Master Catalog: CVOL Processor*. Additional catalog information can be found in the *OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide*. Additional tuning aids are described in "Chapter 3: Catalog Conversion" of the *OS/VS2 Conversion Notebook*.

The performance of VSAM catalogs can be improved by adhering to the following guidelines:

- Choose a primary space value large enough to hold the data to be cataloged, but do not grossly overspecify. Excessive unused space wastes seek time. Secondary space allocation should however be specified to avoid reorganization of the catalog at inconvenient times if the primary space becomes full. Try to arrive at a value for primary space such that secondary space is used only for occasional overflow. A formula is available in *OS/VS2 System Programming Library: Storage Estimates* for the calculation.
- Specify a sufficiently large BUFFERSPACE value in the DEFINE MASTERCATALOG and DEFINE USERCATALOG commands\* for each shared VSAM catalog in order to maximize the number of concurrent searches (LOCATEs) of the VSAM master catalog and each job-shared VSAM user catalog. If, however, a VSAM user catalog is not shared across jobs, there is no need to provide the capability for concurrent searches.

The buffer (defined by BUFFERSPACE) is used for the reading of the index and data portions of the catalog. The minimum or defaulted specification of 3K bytes of pageable virtual storage allows two concurrent searches of the catalog. Each additional 1K bytes permits one more concurrent search, up to a maximum of seven concurrent searches at a BUFFERSPACE value of 8K. (Thus, a BUFFERSPACE value of 4K would allow three concurrent searches.) A maximum of one catalog search per address space at one time is possible, except if there is user subtasking, in which case more than one concurrent catalog search per address space is possible.

*Note:* The BUFFERSPACE value should be specified as or defaulted to the minimum (3K) for a VSAM user catalog that is not shared across jobs.

---

\*The DEFINE commands are described in *OS/VS2 Access Method Services*. The STEPCAT or JOBCAT DD statement for a VSAM user catalog can also specify buffer size, via the BUFSP subparameter of the AMP parameter. (See *OS/VS2 JCL*.) If both the DEFINE command and JCL specify buffer size, the larger specification overrides the smaller. In other words, the JCL specification can increase the buffer size but not decrease it.

## VSAM Catalog Performance (continued)

- Although a large BUFFERSPACE value is good for shared VSAM user catalogs, don't overspecify. There are three reasons:
  1. A BUFFERSPACE value of more than 8K is automatically reduced to 8K without increasing the number of concurrent catalog searches beyond the maximum of seven.
  2. Each increase in BUFFERSPACE value beyond the minimum (3K) requires additional fixed storage, at the rate of 224 bytes of fixed storage for each 1K bytes of increased specification. Thus, a BUFFERSPACE value of 8K would require 1568K bytes ( $448 + 5 \times 224$ ) of fixed storage.
  3. Concurrent searches can occur only when the catalog is being referenced (for example, by Allocation or Open). Concurrent searches will result only to the extent that such referencing occurs simultaneously in two or more address spaces. If a catalog is not shared, there will be no concurrent searches.
- I/O operations to a catalog are performed by use of a control block called a Request Parameter List (RPL). The number of RPLs determines the maximum number of concurrent Locates that can be done. When all the RPLs are in use, the next Locate will cause an exclusive enqueue on the RPL resource. The dequeue will not be done until this Locate completes, thus allowing only one Locate at this time. The number of RPLs is determined by the BUFFERSPACE value specified on the DEFINE MASTERCATALOG or DEFINE USERCATALOG command, or as altered by the ALTER CATALOG command. The larger the BUFFERSPACE size, the greater the number of RPLs. Therefore, in systems where storage is not a critical resource, the BUFFERSPACE parameter value should be increased to 8K for more efficient Locate performance. In systems where storage is at a premium, the default value of 3072 bytes is recommended.
- The installation may wish to use the same VSAM catalog for jobs that normally run concurrently. (Note that although concurrent *searches* can be shared, *updates* always require exclusive access to the catalog.) You may connect concurrently running jobs to the same VSAM user catalog (e.g., ten jobs to one catalog) in order to optimize the use of fixed storage and reduce execution time. One way this can be done is by specifying the same catalog name in the JOBCAT DD or STEPCAT DD statements for the concurrent jobs. In this case, specify a relatively large BUFFERSPACE value, up to 8K. The disadvantage of catalog sharing, however, is that the jobs will contend for the use of the shared catalog.



## VSAM Catalog Performance (continued)

- Avoid a lot of user catalogs in systems where storage is a critical resource. Improved response times have been observed in a two megabyte environment when the number of user catalogs has been reduced from six to two. Fewer user catalogs will require less fixed storage, and can improve response times.
- You may improve catalog search speed by placing all data sets of a job or step, if possible, in the same VSAM user catalog. The catalog should preferably be identified by a JOBCAT or STEPCAT DD statement. Such identification is useful because the catalog search begins with catalogs identified by JOBCAT or STEPCAT DD statements. However, even without such identification some search time can be saved, if data sets used in job steps or time sharing sessions have the same highest level dsname qualifier. In this case, the VSAM master catalog will be searched the first time that a data set must be found. Subsequent searches will bypass the master catalog and go directly to the appropriate user catalog.

## VSAM Catalog Performance (continued)

### Questions and Answers

The following questions on the VSAM catalog have been asked by customers.

*Q: Will connected CVOLs and STEPCAT or JOBCAT catalogs be mounted at interpreter time?*

*A: In general, no. The catalogs are mounted at the time they are allocated.*

*Q: How serious is the loss of the VSAM master catalog to MVS?*

*A: If there are no VSAM data sets represented in the master catalog, other than paging, swapping, the catalog itself, and SYS1.STGINDEX, then the impact is similar to losing an OS master catalog. If VSAM data sets exist, then VSAM recovery techniques and considerations apply. (See *OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide*.) The VSAM master catalog, as well as paging data sets, swapping data sets, and SYS1.STGINDEX, must be rebuilt on an MVS system. Non-VSAM data sets still have VTOC entries and can be recataloged.*

If the VSAM master catalog can't be opened, or if certain system data sets can't be found (via LOCATE macros), the system can't be initialized. If there are catalog problems after system initialization, the impact to the system depends on the degree of reliance on the catalog and on the fraction of the catalog that is unusable. (It is unlikely that the entire master catalog will become unusable at the same time.)

*Q: Under what circumstances should an installation convert its OS CVOLs to VSAM user catalogs?*

*A1: Use CVOL support and do not immediately convert CVOLs to VSAM user catalogs, if you have varied systems, some under MVT and one or more under MVS, and you wish to use your data on both systems. As applications are moved to the MVS system(s), gradually move the data sets to VSAM user catalogs for improved performance and easier maintenance\*.*

*A2: Convert OS CVOLs to VSAM user catalogs, via the CONVERTCAT utility, for improved performance and easier maintenance\*, if you have one or more MVS systems that have users with CVOLs, and there are no longer any MVT systems at your installation. Such conversion is desirable even though the VSAM master catalog *could* continue to point to the CVOLs.*

---

\*Utilities IEHLIST and IEHPROGM do not have full function for OS CVOLs under MVS.

## How SMF Can Supplement MF/1

MF/1 records or reports can be used to identify intervals during which the utilization of certain system resources has been unusual. MF/1 data includes information on CPU, the paging subsystem, channels and devices, pageable real storage, and workload activity. SMF can be used concurrently to list information that describes the workload processed during the same time interval. There may be a correlation between unusual resource utilization and the processing of particular batch jobs or time sharing sessions.

MF/1 can be used to determine the average system workload level over a relatively long period of time. SMF can be used during the same period to identify the exceptional jobs, jobsteps, and TSO sessions that received service rates significantly different from those defined in the IPS at that workload level. Exceptional jobs or TSO sessions are those that either receive unusual service or place exceptional demands on system resources. System resources consist of the paging subsystem, the CPU, channels, devices, real storage, etc.

Exceptional jobs or TSO sessions can be identified by summarizing SMF statistics\*. When these jobs and sessions have been identified, the installation can investigate the reasons for the exceptional service or demands, and then make changes. For example, a change to the blocking factor may turn a heavy I/O-demand job into an average I/O-demand job receiving average service.

MF/1 and SMF together can be used for at least the following purposes:

- Comparison of paging rates for a problem program and the system
- Comparison of I/O activity for a problem program and the system
- Comparison of problem program service versus total service
- Determining changes to the system configuration

### Comparison of Paging Rates for a Problem Program and the System

This program-versus-system comparison can suggest paging problems within the problem program. The program's paging rate is indicated by SMF record types 4 and 34 (see *OS/VS System Management Facilities (SMF)*). The installation can calculate paging rate for the jobstep from the formula:

Paging rate = jobstep pages in and out / CPU time for the jobstep.

A similar rate for the system can be determined during the same time interval from SMF record types 70 and 71. Type 70 contains the length of the measurement interval and the CPU wait time during the interval. Type 71 contains paging data, including page-ins and page-outs, and the interval length. The system calculation is basically the same as the job step calculation stated above, except that CPU time must first be determined by subtracting CPU wait time from the total time interval.

---

\*The Statistics Gathering Program (SGP) is available as a data reduction tool for SMF data. The program is distributed as field-released program No. 5798-AYY.

## **How SMF Can Supplement MF/1 (continued)**

### **Comparison of I/O Activity for a Problem Program and the System**

This program-versus-system comparison on first observation seems to be one of apples to oranges, since MF/1 counts SIOs, and SMF counts EXCPs (which include SVCO's, PCI interrupts, channel-end interrupts, and abnormal-end interrupts). It is possible, however, to obtain a ratio of EXCPs to SIOs by using the sum of the jobstep EXCPs from SMF and the sum of the SIO counts from MF/1 for the same measurement period. When you have established this I/O ratio, you may be able to detect those jobsteps whose I/O operations seem excessive. Appropriate corrective action can then be taken.

### **Comparison of Problem Program Service and Total Service**

The installation can compare the service given to particular performance groups versus total system service by examining SMF record types 5 and 35, and comparing their data with that in record type 72 which is produced by MF/1. Record types 5 and 35 contain the number of service units required by a job or TSO session. Record type 72 gives the total service provided for all jobs and TSO sessions. By comparing the data from these two sources, the installation can determine whether service is being distributed according to the goals of the installation.

### **Determining Changes to the System Configuration**

SMF provides information on the system configuration at IPL and changes to that configuration while the system is running. The data is contained in record types 0, 8, 9, 10, 11, 22, 70, 73, and 74. This information is important during analysis of MF/1 data, since it can explain significant changes in the MF/1 output. Explanation of these changes might otherwise be left to speculation or to an exhaustive study of the operator's console output.

## The Use of GTF to Track Sysevents

System components issue the SYSEVENT macro to inform the System Resource Manager (SRM) that the status of an address space or a system resource has changed. (The SYSEVENT is somewhat analogous to the TSEVENT macro in MVT.) The SRM is informed of critical changes to the availability of real storage frames, auxiliary storage slots, and SQA virtual space. In addition, the SYSEVENT may request that the SRM perform a service. For example, the REQSERVC sysevent (code X'26') is issued to request that the SRM obtain service data related to a particular address space. (See Figure 5-3 at the end of this topic for descriptions of the various SYSEVENT codes.)

The installation can monitor SYSEVENT activity during a selected time interval by starting GTF with the SRM option. The SRM option causes GTF to write a trace record each time a SYSEVENT macro is issued. (Information on how to use GTF is available in *OS/VS2 System Programming Library: Service Aids*.) The SYSEVENT trace record contains a time stamp, an address space control block (ASCB) address, a CPU identification (CPUID), a jobname (if applicable, and if comprehensive trace records have been requested), and contents of regs 0, 1, and 15 which contain information peculiar to the particular SYSEVENT. The information includes the SYSEVENT code that indicates why the SRM was invoked. (See Figure 5-2 for the format of the SRM trace record.)

Field Name	length	reserved	record identifier (AID) X'FF' (See note 1)	format identifier (FID) X'04' (See note 2)	timestamp	event identifier (EID) X'4001' (See note 3)	ASCB addr	CPUID	jobname (See note 4)	R15 contents	R0 contents	R1 contents
Length	2	2	1	1	8	2	4	2	8	4	4	4
Hex Offset	0	02	04	05	06	0E	10	14	16	1E	22	26

**Notes:**

1. The record identifier (AID) in GTF records is a one-byte hexadecimal number that identifies the record as a trace record or a control record.
2. The format identifier (FID) in GTF records is a one-byte hexadecimal number that is used to determine the name of the AMDPRDMP EDIT module that will format the record.
3. The event identifier (EID) in GTF trace records is a two-byte hexadecimal number that defines the event that caused the record to be built. The EID is in the form cddd where c is the event class (0-F) and ddd is the ID of the event within the class.
4. Jobname appears only in the comprehensive trace record. Comprehensive format must be requested when GTF is started.

**Figure 5-2. SRM Trace Record Format**

## The Use of GTF to Track Sysevents (continued)

The installation should reduce the trace data to determine the utilization of system resources and the service given to particular jobs or TSO sessions. The reduced data can supplement that obtained from MF/1 and SMF (see the performance topic "How SMF Can Supplement MF/1"). For example, the trace data can indicate when shortages in paging space occur. This information can be used with SMF data to learn the identify of jobs and TSO sessions active at the time of the shortage. Possible corrective action may include the addition of more paging space, or external scheduling at the installation to prevent all the heavy paging jobs (possibly VIO users) from running at the same time.

The sysevent data can be used to analyze the effect of the system running with insufficient SQA virtual storage. Distributions can be produced which identify the frequency and durations of the shortage periods. These intervals can be correlated with MF/1 data to determine how the SQA shortage affects CPU utilization, channel utilization, and service unit utilization. If the shortage time is excessive or if it causes a significant degradation of performance, the installation can make more space available for SQA virtual storage.

Similarly, the sysevent data can be used to analyze the effect of the system running with a shortage of available page frames, i.e., in an AVQLow\* environment. If the shortage time is excessive or if it impacts performance, the installation may wish to modify the Real Storage Manager (RSM) constants that define the shortage.

A sysevent trace tape can be sorted by address space ID (ASID), sysevent type, or by time of occurrence, etc. The sort yields a sequence of tabulated records for each job or TSO session. The time difference between the individual printed records is the time between significant stages of a job or TSO session. The stages include such items as address space creation, job selection, (START, MOUNT, or LOGON issued), initiator attaching a task, initiator detaching a task, and job termination. The sequence also contains a record for each time the address space is swapped into or out of real storage. For a TSO session, the sequence contains a record of each time the address space enters wait state and each time it becomes ready.

---

\*SYSEVENT X'17'

## The Use of GTF to Track Sysevents (continued)

Through appropriate reduction of the data, the installation can obtain some meaningful time distributions. Some examples are:

- time needed to swap an address space in
- time needed to swap an address space out
- time address spaces spend in real storage
- time address spaces spend out of real storage
- user think time
- system response rate
- transaction arrival rate
- transaction processing rates
- job processing rates
- TSO session durations

Information is also provided in the trace which identifies the names of time sharing commands and subcommands (see sysevent code 00 in Figure 5-3). This information should help the installation to produce time distributions for the individual TSO commands and subcommands.

Figure 5-3 describes the meaning and input information (register contents) for each sysevent type.

Figures 5-4 through 5-8 show sample printouts of a data-reduced sysevent trace. Note that unusual data are circled and annotated.

---

<b>Sysevent Code:</b>	00
<b>Mnemonic:</b>	TSEVENT 00 (PPMODE)
<b>Meaning of Mnemonic:</b>	A time sharing command, or a subcommand of EDIT or TEST, is to be executed.
<b>Circumstances:</b>	The TSO Terminal Monitor Program or the EDIT/TEST command processor issues this sysevent when the command or subcommand is about to be executed. It causes no action on the part of the SRM.
<b>Inputs:</b>	Reg 0, bytes 0–3: ASID Reg 1, bytes 0–3: Contains the first four characters of the command or subcommand name. Reg 15: Contains the last four characters of the command or subcommand name.
<b>Outputs:</b>	None

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 1 of 24)

## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	01
<b>Mnemonic:</b>	TIMEREXP
<b>Meaning of Mnemonic:</b>	SRM time interval expired.
<b>Purpose:</b>	Assures that the SRM will gain control at least once within a predetermined real time interval.
<b>Circumstances:</b>	The timer routines have recognized that the SRM time interval has just popped (elapsed), or TOD clock initialization has occurred. At the time the sysevent is issued, the SRM's timer queue element has been removed from the timer Queue.
<b>Inputs:</b>	Reg 0, byte 3: Sysevent code Reg 1, byte 3: Contains X'01' if entry is from system TOD clock initialization. Contains X'00' otherwise.
<b>Outputs:</b>	None.

---

<b>Sysevent code:</b>	02
<b>Mnemonic:</b>	TERMWAIT
<b>Meaning of Mnemonic:</b>	Terminal wait.
<b>Purpose:</b>	Indicates that a TSO Session has entered terminal wait.
<b>Circumstances:</b>	A TSO Session is in terminal wait after the issuance of a TGET or a TPUT. Receipt of the TERMWAIT sysevent indicates to the SRM that the current transaction for a TSO address space should be ended, provided that the address space has entered long wait status and is swappable. Note that the occurrence of this sysevent does not guarantee that the entire address space is in a long wait status. This determination can only be made by Quiesce.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID. Reg 0, byte 3: Sysevent code. Reg 1, byte 0: contains X'00' if input terminal wait; contains X'80' if output terminal wait.
<b>Outputs:</b>	None.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 2 of 24)



### The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	03
<b>Mnemonic:</b>	NIOWAIT
<b>Meaning of Mnemonic:</b>	Address space suspected of being in long wait.
<b>Purpose:</b>	Indicates to SRM when an address space is suspected of having entered long wait.
<b>Circumstances:</b>	Some task in the address space just entered long wait. Occurrence of this sysevent does not guarantee that the entire address space is in a long wait status. This determination can be made only by Quiesce. The time spent by a swappable address space in long wait will not be considered part of the current transaction for that address space.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code
<b>Outputs:</b>	None.

---

<b>Sysevent Code:</b>	04
<b>Mnemonic:</b>	USERRDY
<b>Meaning of Mnemonic:</b>	User ready.
<b>Purpose:</b>	Indicates that a swapped out address space in a wait state or an address space for which Quiesce is running has at least one dispatchable unit (SRB)* which is ready to run.
<b>Circumstances:</b>	Something has occurred causing a dispatchable unit (SRB) to be scheduled to this address space.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID. Reg 0, byte 3: Sysevent code.
<b>Outputs:</b>	None.

\*SRB means system resource block.

Figure 5-3. Descriptions of SYSEVENT Codes (Part 3 of 24)

## The Use of GTF to Track Sysevents (continued)

<b>Sysevent Code 05:</b>	This code is not used.
<b>Sysevent Code:</b>	06
<b>Mnemonic:</b>	MEMCREAT
<b>Meaning of Mnemonic:</b>	Address space create.
<b>Purpose:</b>	Indicates that a new address space is about to be created. Indicates the type of origin of the new address space (i.e., START, LOGON, MOUNT). Gives the SRM a chance to prohibit the creation of the address space.
<b>Circumstances:</b>	At the earliest point where the ASID is known and the space for the ASCB has been obtained.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID. Reg 0, byte 3: Sysevent code. Reg 1, byte 3: contains X'01' if START; X'02' if LOGON; X'03' if MOUNT.
<b>Outputs:</b>	Reg 1, byte 0: contains X'00' if acceptable to proceed; contains X'80' if the address space should not be created because of a resource shortage as determined by the SRM.
<b>Sysevent Code:</b>	07
<b>Mnemonic:</b>	MEMDEL
<b>Meaning of Mnemonic:</b>	Address space delete.
<b>Purpose:</b>	Indicates to the SRM the deletion of an address space, allowing the SRM to release resources assigned to that address space.
<b>Circumstances:</b>	Memory Delete is about to free the storage for the ASCB and unassign the ASID.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID. Reg 0, byte 3: Sysevent code.
<b>Outputs:</b>	Reg 1, byte 3: contains X'04' indicating that Memory Delete is to wait until posted by the SRM.

Figure 5-3. Descriptions of SYSEVENT Codes (Part 4 of 24)

### The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent code:</b>	08
<b>Mnemonic:</b>	JOBSELECT
<b>Meaning of Mnemonic:</b>	Job selection.
<b>Purpose:</b>	Indicates that an address space has started using system services on behalf of a new job, START or MOUNT command, or a TSO session.
<b>Circumstances:</b>	Whenever IEFSD161 handles one of the above.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 2: performance group number. Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: pointer to jobname or user ID.
<b>Outputs:</b>	None.

---

<b>Sysevent Code:</b>	09
<b>Mnemonic:</b>	JOBTERM
<b>Meaning of Mnemonic:</b>	Job termination.
<b>Purpose:</b>	Indicates that an address space has completed using system services on behalf of a job, START or MOUNT command, or a TSO session.
<b>Circumstances:</b>	Whenever IEFSD166 handles one of the above.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: pointer to jobname or user ID.
<b>Outputs:</b>	None.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 5 of 24)

### The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	0A
<b>Mnemonic:</b>	INITATT
<b>Meaning of Mnemonic:</b>	Attach by initiator.
<b>Purpose</b>	Indicates an initiator has attached a task. Is related to a JOBSELECT sysevent (code 8).
<b>Circumstances:</b>	Whenever IEFSD263 attaches a task.
<b>Inputs:</b>	<p>Reg 0, bytes 0-1: ASID or zero.            Reg 0, byte 3: Sysevent code.            Reg 1, byte 2: Performance Group Number of attached task, or 0. The presence of a 0 in this byte indicates that the address space will execute in a privileged status reserved primarily for system tasks, in which Service Rate will not be a swap consideration.</p> <p>Reg 1, byte 3: Dispatching priority to which this address space should be set.</p>
<b>Outputs:</b>	None.

---

<b>Sysevent Code:</b>	0B
<b>Mnemonic:</b>	INITDET
<b>Meaning of Mnemonic:</b>	Detach by initiator.
<b>Purpose:</b>	Indicates a task has been detached by an initiator.
<b>Circumstances:</b>	Whenever IEFSD263 detaches a task.
<b>Inputs:</b>	<p>Reg 0, bytes 0-1: ASID or zero.            Reg 0, byte 3: Sysevent code.            Reg 1, byte 3: Dispatching priority to which this address space should be set.</p>
<b>Outputs:</b>	None.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 6 of 24)

## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	0C
<b>Mnemonic:</b>	QSCEST
<b>Meaning of Mnemonic:</b>	Quiesce started.
<b>Purpose:</b>	Permits an initial assessment of whether an address space, suspected of being in long wait, is in fact in long wait. Provides for the reversal of the Quiesce of an address space.
<b>Circumstances:</b>	The SRM has recently posted Quiesce.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code. Reg 1, byte 0: Contains X'00' if the address space is not in long wait; X'80' if all tasks in the address space are in long wait.
<b>Outputs:</b>	Reg 1, byte 3: Contains X'00' if RCT is to continue with Quiesce; contains X'08' if the address space should be restored to its original status.

---

<b>Sysevent Code:</b>	0D
<b>Mnemonic:</b>	QSCECMP
<b>Meaning of Mnemonic:</b>	Quiesce completed.
<b>Purpose:</b>	Permits a final assessment of whether the address space is to be swapped out. If between QSCEST (code 12) and QSCECMP, a USERRDY (code 4) has been received for the address space, Quiesce will be notified that the memory is not in true long wait status.
	<i>Note:</i> The swapped in interval is defined to end with this sysevent.
<b>Circumstances:</b>	RCT has completed quiesce processing for an address space.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code. Reg 1, byte 0: Contains X'00', if address space not in long wait; X'80' if address space is in long wait.
<b>Outputs:</b>	Reg 1, byte 0: contains X'00' if USERRDY (code 4) was just received; unchanged by SRM if no USERRDY received since QSCEST (code 12). Reg 1, byte 2: Contains swap reason code. Reg 1, byte 3: Contains X'00' if the RCT is to schedule swap-out; X'08' if address space is to be restored.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 7 of 24)

## The Use of GTF to Track Sysevents (continued)

<b>Sysevent Code:</b>	0E this code is not used.
<b>Sysevent Code:</b>	0F
<b>Mnemonic:</b>	SWOUTCMP
<b>Meaning of Mnemonic:</b>	Swap-out completed.
<b>Purpose:</b>	Indicates that swap-out processing has completed.
<b>Circumstances:</b>	All I/O needed to swap-out this address space has just completed.
<b>Inputs:</b>	<p>Reg 0, bytes 0-1: ASID or zero.            Reg 0, byte 3: Sysevent code.            Reg 1, bytes 0-3: Address of a parameter list.            The format is as follows:</p> <p>Word 1, bytes 0-1: The number of pages swapped-out.            Word 1, bytes 2-3: The working set size (the number of pages to be swapped-in).            Word 2, bytes 0-2: The number of pages freed by swap-out without performing I/O.            Word 2, byte 3: - Flag byte indicating:</p> <p>Bits 0-6: Reserved            Bit 7: 0 if address space is in long wait;            1 if address space is waiting for an unfinished Real Storage Manager service.</p>
<b>Outputs:</b>	None.
<b>Sysevent Code:</b>	10 (hex).
<b>Mnemonic:</b>	SWINSTAT
<b>Meaning of Mnemonic:</b>	Swap-in status.
<b>Circumstances:</b>	Swap-in processing for an address space that has just started, or just completed.
<b>Inputs:</b>	<p>Reg 0, bytes 0-1: ASID or zero.            Reg 0, byte 3: SYSEVENT code.            Reg 1, byte 3:</p> <p>01: Swap-in is starting.            02: Swap-in is complete.</p>
<b>Outputs:</b>	None.

Figure 5-3. Descriptions of SYSEVENT Codes (Part 8 of 24)

**The Use of GTF to Track Sysevents (continued)**

---

<b>Sysevent Code:</b>	11 (hex)
<b>Mnemonic:</b>	SWINFL
<b>Meaning of Mnemonic:</b>	Swap-in failed
<b>Circumstances:</b>	Swap-in processing failed to obtain or initialize the LSQA and fixed pages for the specified address space.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code
<b>Outputs:</b>	None.

---

**Figure 5-3. Descriptions of SYSEVENT Codes (Part 8.1 of 24)**





## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	12 (hex)
<b>Mnemonic:</b>	QSCEFL
<b>Meaning of Mnemonic:</b>	Quiesce failed.
<b>Purpose:</b>	Notifies the SRM that during an attempt to quiesce an address space the Quiesce function has failed. The address space has been restored when the sysevent is issued.
<b>Circumstances:</b>	Region Control Task failed to complete quiesce processing due to an abnormal situation.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code.
<b>Outputs:</b>	None.

---

<b>Sysevent Code:</b>	13 (hex)
<b>Mnemonic:</b>	RSTORCMP
<b>Meaning of Mnemonic:</b>	Restore completed.
<b>Purpose:</b>	Permits an assessment of whether an address space, suspected of having left long wait status, is in fact ready.
<b>Circumstances:</b>	Region Control Task has completed restore processing for an address space. The circumstances giving rise to the restoring of an address space still in long wait stem from not knowing that the address space is waiting on more than one event.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code. Reg 1, byte 0: Contains X'00' if the address space is ready; contains X'80' if the address space is in long wait.
<b>Outputs:</b>	None.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 9 of 24)

## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	14 (hex)
<b>Mnemonic:</b>	ENQHOLD
<b>Meaning of Mnemonic:</b>	ENQ contention occurred.
<b>Purpose:</b>	Informs the SRM of an address space responsible for contention.
<b>Circumstances:</b>	An address space is now being delayed because a resource is held. Contention between tasks in a single address space is not distinguished from contention between address spaces.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID of address space holding the resource. Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: Address of minor QCB for the resource.
<b>Outputs:</b>	None.

---

<b>Sysevent Code:</b>	15 (hex)
<b>Mnemonic:</b>	ENQRLSE
<b>Meaning of Mnemonic:</b>	ENQ contention reduced.
<b>Purpose:</b>	Informs the SRM of an address space which formerly was responsible for contention.
<b>Circumstances:</b>	Contention has disappeared because a resource has been released.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID of address space holding the resource during contention. Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: Address of minor QCB for resource.
<b>Outputs:</b>	None.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 10 of 24)

## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	16 (hex)
<b>Mnemonic:</b>	RSMCNSTS
<b>Meaning of Mnemonic:</b>	Real Storage Manager constants.
<b>Purpose:</b>	Supplies the SRM with the size of functioning real storage and the number of pages on the available frame queue, when the "available frame queue below limit" sysevent (code X'17') is issued.
<b>Circumstances:</b>	Issued when a VARY Storage command is processed and when the system is initialized.
<b>Inputs:</b>	Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-1: Number of pages of functioning real storage. Reg 1, bytes 2-3: Number of pages that are on the Available Frame Queue when the "available frame queue below limit" sysevent is issued.
<b>Outputs:</b>	None.

---

<b>Sysevent Code:</b>	17 (hex)
<b>Mnemonic:</b>	AVQLOW
<b>Meaning of Mnemonic:</b>	Available frame queue below limit.
<b>Purpose:</b>	Notifies the SRM that the number of real pages on the available frame queue has dropped below predefined limits.
<b>Circumstances:</b>	Issued whenever allocation of a real page frame causes the number left on the available frame queue to drop below one of the predefined limits.
<b>Inputs:</b>	Reg 0, byte 3: Sysevent code. Reg 1, byte 3: X'01' if the number of real pages on the available frame queue has dropped below the limit. X'02' if the number of real pages on the available frame queue has dropped to zero. X'03' if a page fault occurs and there are no pages on the available frame queue. X'04' if the ratio of fixed pages to total pages has increased above the allowable value.
<b>Outputs:</b>	None.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 11 of 24)

## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	18 (hex)
<b>Mnemonic:</b>	AVQOK
<b>Meaning of Mnemonic:</b>	Available frame queue above limit.
<b>Purpose:</b>	Notifies the SRM that the number of real pages on the available frame queue has risen above a predefined limit.
<b>Circumstances:</b>	Is issued whenever unallocation of a real page frame causes the number left on the available frame queue to rise above the predefined limit. This sysevent is issued only when the number of real pages rises above the predefined limit after the "available frame queue below limit" sysevent (code 17) was issued.
<b>Inputs:</b>	Reg 0, byte 3: Sysevent code.
<b>Outputs:</b>	None.

---

<b>Sysevent Code:</b>	19 (hex)
<b>Mnemonic:</b>	SQALOW
<b>Meaning of Mnemonic:</b>	Unallocated SQA below threshold.
<b>Purpose:</b>	Indicates that the amount of unallocated virtual SQA has dropped below one of two predefined thresholds.
<b>Circumstances:</b>	Virtual Storage Manager has just satisfied an SQA allocation request which resulted in the amount of unallocated SQA dropping below one of the two predefined thresholds.
<b>Inputs:</b>	Reg 0, byte 3: Sysevent code. Reg 1, byte 3: Contains X'01' if first (less serious) threshold is passed; X'02' if second threshold is passed.
<b>Outputs:</b>	None

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 12 of 24)

---

**The Use of GTF to Track Sysevents (continued)**

---

<b>Sysevent Code:</b>	1A
<b>Mnemonic:</b>	SQAOK
<b>Meaning of Mnemonic:</b>	Unallocated SQA above threshold.
<b>Purpose:</b>	Indicates that the amount of unallocated SQA has risen above one of two predefined thresholds.
<b>Circumstances:</b>	Virtual Storage Manager has just handled an SQA unallocation request which resulted in the amount of unallocated SQA rising above one of the two predefined thresholds.
<b>Inputs:</b>	Reg 0, byte 3: Sysevent code. Reg 1, byte 3: Contains X'01' if first (less serious) threshold has passed; X'02' if second threshold passed.
<b>Outputs:</b>	None.

---

<b>Sysevent Code:</b>	1B This code is not used.
-----------------------	---------------------------

---

**Figure 5-3. Descriptions of SYSEVENT Codes (Part 13 of 24)**

## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	1C
<b>Mnemonic:</b>	DEVALLOC
<b>Meaning of Mnemonic:</b>	Device allocation request.
<b>Purpose:</b>	Provides SRM with necessary data for making a device allocation decision where two or more candidates exist.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code.  Reg 1, bytes 0-3: Address of a list of three full-word addresses. The first points to a list of candidate UCB addresses. The second points to a list of addresses of UCBs already allocated to the requesting jobstep. The third points to a two-word return area.  The first word in the list of candidate UCBs contains a count of the number of candidates in the list. The first word of the list of addresses of already allocated UCBs contains a count of the number of addresses in the list. All input and output data areas must be fixed.
<b>Outputs:</b>	Reg 1, bytes 0-3: Contains the same address present at input.  Return area 1st word: Contains the address of the candidate list entry which was selected.  Reg 15, byte 3: Contains X'00' if allocation selection was successfully made; X'08' if it was unsuccessful.

---

<b>Sysevent Code:</b>	1D
<b>Mnemonic:</b>	CONFIGCH
<b>Meaning of Mnemonic:</b>	System configuration change.
<b>Purpose:</b>	Indicates that resources are to be removed from or added to the system.
<b>Circumstances:</b>	The system operator has issued a VARY (online or offline) command for a channel, path, or CPU.
<b>Inputs:</b>	Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: Points to the SMF type 22 record that describes the configuration change.
<b>Outputs:</b>	None.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 14 of 24)

## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	1E
<b>Mnemonic:</b>	VERIFYPG
<b>Meaning of Mnemonic:</b>	Verify performance group.
<b>Purpose:</b>	To determine if the input Performance Group Number is currently "known" to the SRM, and to indicate the default value if the input number is not "known".
<b>Circumstances:</b>	LOGON or the Converter/Interpreter has received a Performance Group Number which needs verification.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code. Reg 1, byte 3: Performance Group Number.
<b>Outputs:</b>	Reg 1, byte 2: Contains 0 if the input number is valid. If the input number is not valid, it contains 1 if the ASID belongs to a non-TSO user, or 2 if the ASID belongs to a TSO user.

---

<b>Sysevent Code:</b>	1F
<b>Mnemonic:</b>	RESETPG
<b>Meaning of Mnemonic:</b>	Reset performance group.
<b>Purpose:</b>	Resets the Performance Group Number associated with an ASID.
<b>Circumstances:</b>	The system operator has entered a RESET jobname, PERFORM=nn command.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID. Reg 0, byte 3: Sysevent code. Reg 1, byte 3: New performance group number.
<b>Outputs:</b>	Reg 1, byte 2: Contains X'00' if the RESET request was honored, X'04' if the new Performance Group Number is not valid, or X'08' if the ASID is not currently assigned.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 15 of 24)

## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	20 (hex)
<b>Mnemonic:</b>	NEWIPS
<b>Meaning of Mnemonic:</b>	Set new IPS.
<b>Purpose:</b>	Change the IPS currently used by the SRM.
<b>Circumstances:</b>	The system operator has entered a SET command with the IPS keyword. To synchronize the deletion of the old IPS, the SET command processor waits on an ECB which will be posted by the SRM only after all references to the old IPS have been replaced. The SET command processor is responsible for obtaining the storage for the new IPS description, and for releasing the storage for the old IPS description, when the SRM has indicated that it will no longer be referenced.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: Contains the address of the SRM Workload Manager specification table (WMST) that describes the new IPS.
<b>Outputs:</b>	Reg 1, bytes 0-3: Contains the address of the SRM Workload Manager specification table (WMST) that describes the old IPS.

---

<b>Sysevent Code:</b>	21 (hex)
<b>Mnemonic:</b>	ALTCPREC
<b>Meaning of Mnemonic:</b>	Alternate CPU recovery (ACR)
<b>Purpose:</b>	Notifies the SRM that one CPU has been removed from the configuration.
<b>Circumstances:</b>	As a result of some error, ACR has had to reconfigure one CPU out of the system.
<b>Inputs:</b>	Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: CPU address of failed CPU.
<b>Outputs:</b>	None.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 16 of 24)



### The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	22 (hex)
<b>Mnemonic:</b>	TGETTPUT
<b>Meaning of Mnemonic:</b>	TGET/TPUT satisfied.
<b>Purpose:</b>	Indicates a change in the status of the current TSO transaction.
<b>Circumstances:</b>	TGET or TPUT completed.
<b>Inputs:</b>	<p>Reg 0, bytes 0-1: ASID or zero.          Reg 0, byte 3: Sysevent code.          Reg 1, byte 0: - Flag byte, as follows:</p> <p>bit 0: Contains 0 if TGET was satisfied; contains 1 if TPUT was satisfied.</p> <p>bit 1: (applies to TGET satisfied only.) Contains 0 if all the data in the TSO input message was transferred by the TGET; contains 1 if part of the data in the TSO input message was not yet transferred by this TGET, i.e., if at least one more TGET is required to obtain the rest of the data in the TSO input message.</p> <p>bits 2-7: reserved</p>
<b>Outputs:</b>	None.

---

<b>Sysevent Code:</b>	23 (hex) This Sysevent is not traced by GTF.
<b>Mnemonic:</b>	SYQSCST
<b>Meaning of Mnemonic:</b>	System Quiesce function started.
<b>Purpose:</b>	Indicates that task and I/O activity in the system are being quiesced.
<b>Circumstances:</b>	The system operator has issued the QUIESCE command. Before the SYQSCST sysevent is issued, all CPUs should have been placed under the control of the Quiesce function.
<b>Inputs:</b>	Reg 0, byte 3: Sysevent code.
<b>Outputs:</b>	None.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 17 of 24)

## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	24 (hex) This Sysevent is not traced by GTF.
<b>Mnemonic:</b>	SYQSCCMP
<b>Meaning of Mnemonic:</b>	System Quiesce function completed.
<b>Purpose:</b>	Indicates that task and I/O activity in the system are being resumed.
<b>Circumstances:</b>	When the system has been quiesced, the operator has pushed the START button.
<b>Inputs:</b>	Reg 0, byte 3: Sysevent code.
<b>Outputs:</b>	None.

---

<b>Sysevent Code:</b>	25 (hex)
<b>Mnemonic:</b>	SETDMN
<b>Meaning of Mnemonic:</b>	Set new constraint values for a domain. Changes the MPL constraint values for a specific domain.
<b>Circumstances:</b>	The operator has issued the SETDMN command.
<b>Inputs:</b>	Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: Contains the address of a fixed data area. The data area is as follows: Byte 0: domain number. Byte 1: flags that indicate whether a corresponding value is passed in byte 2, 3 or 4. Byte 2: new minimum constraint value. Byte 3: new maximum constraint value. Byte 4: new domain weight.
<b>Outputs:</b>	Reg 15, byte 3: 00 — Sysevent successful. 04 — Domain is invalid. 08 — Minimum constraint would exceed maximum constraint.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 18 of 24)

## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	26 (hex)
<b>Mnemonic:</b>	REQSERVC
<b>Meaning of Mnemonic:</b>	Request for service data.
<b>Purpose:</b>	Permits service-related data to be obtained for a given address space from the SRM.
<b>Circumstances:</b>	<p>TSO TIME command will also use the REQSERVC sysevent to obtain service data. The REQSERVC sysevent must be issued prior to the JOBTTERM sysevent (code 9), because the address space data is reset upon receipt of JOBTTERM.</p> <p>The output area does not have to be fixed, and the issuer is not required to be authorized.</p>
<b>Inputs:</b>	<p>Reg 0, bytes 0-1: ASID or zero.          Reg 0, byte 3: Sysevent code.          Reg 1, bytes 0-3: Contains address of a 3-word area where the service data is to be stored.</p>
<b>Outputs:</b>	<p>Service data supplied by SRM:</p> <p>In the case of a TSO address space, the 3-word area contains:</p> <p>Word 1 - Total service for the job.          Word 2 - Total transaction active time.          Word 3 - bytes 0-1: Performance group number last assigned to the address space.          Bytes 2-3: Total number of transactions.</p> <p>In the case of a non-TSO address space, the 3-word area contains:</p> <p>Word 1 - Total service for the session.          Word 2 - Total active time for all transactions.          Word 3 - bytes 0-1: Performance group number last assigned to the address space.          Bytes 2-3: Zeros.          Reg 15, byte 3: Contains X'04' if data was lost due to accumulation control block error; X'00' otherwise.</p>

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 19 of 24)

## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	27 (hex)
<b>Mnemonic:</b>	REQPGDAT
<b>Meaning of Mnemonic:</b>	Request by SMF for job paging data.
<b>Purpose:</b>	Permits SMF to obtain paging data for a given address space from the SRM.
	<i>Note:</i> This sysevent is intended only for use by SMF because the related data fields in the OUSB and the OUXB are reset to zero on readouts. If requested by another caller, the data would be unavailable to SMF.
<b>Circumstances:</b>	SMF issues REQPGDAT during step termination. The REQPGDAT sysevent must be issued prior to the JOBTERM sysevent (code 9) because the address space data is reset upon receipt of JOBTERM.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: Contains the address of a 14-word area where the paging data is to be stored.
<b>Outputs:</b>	Paging data supplied by SRM: <ul style="list-style-type: none"> <li>Word 1 - Count of non-VIO page-ins.</li> <li>Word 2 - Count of non-VIO Page-outs.</li> <li>Word 3 - Count of non-VIO reclaims.</li> <li>Word 4 - Count of VIO page-ins.</li> <li>Word 5 - Count of VIO page-outs.</li> <li>Word 6 - Count of VIO reclaims.</li> <li>Word 7 - Count of pages swapped in.</li> <li>Word 8 - Count of pages swapped out.</li> <li>Word 9 - Count of swapouts.</li> <li>Word 10 - Count of common area page-ins.</li> <li>Word 11 - Count of common area reclaims.</li> <li>Word 12 - Count of pages stolen.</li> <li>Word 13, 14 - Count of CPU page-seconds.</li> </ul> <p>Reg 15, byte 3: Indicates whether data was successfully returned (00), or not (04).</p>

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 20 of 24)

**The Use of GTF to Track Sysevents (continued)**

---

<b>Sysevent Code:</b>	28 (hex)
<b>Mnemonic:</b>	COPYDMDT
<b>Meaning of Mnemonic:</b>	Copy domain table information.
<b>Purpose:</b>	Obtain a copy of the SRM's domain table.
<b>Circumstances:</b>	The operator has issued the DISPLAY command with the DMN parameter.
<b>Inputs:</b>	Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: Contains the address of a 2584-byte fixed data area.
<b>Outputs:</b>	Reg 1 contains the same address as input.  The fixed data area on output contains: Bytes 0-1: Number of valid domains. Bytes 2-3: Reserved. Bytes 4-2584: Copy of SRM's domain table.

---

**Figure 5-3. Description of SYSEVENT Codes (Part 20.1 of 24)**



## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	29 (hex)
<b>Mnemonic</b>	DONTSWAP
<b>Meaning of Mnemonic:</b>	Address Space is now not swappable.
<b>Purpose:</b>	Indicates to the SRM that the issuing address space must not be swapped until further notice.
<b>Circumstances:</b>	Application dependent.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID of issuing address space, or zero. Reg 0, byte 3: Sysevent code.
<b>Outputs:</b>	Reg 1, byte 3: Contains X'04' if request is not for the current address space; contains X'00' if the request to mark the address space as non-swappable was honored; contains X'08' if request was not authorized, or if the outstanding count of DONTSWAP requests (code 29) has reached its maximum value.  <i>Note:</i> Nonswappable status is reset at the time of attach by initiator (code 0A) and detach by initiator (code 0B). That is, nonswappable status is not carried across job steps.
<hr/>	
<b>Sysevent Code:</b>	2A
<b>Mnemonic:</b>	OKSWAP
<b>Meaning of Mnemonic:</b>	Memory now swappable.
<b>Purpose:</b>	Indicates to the SRM that the issuing address space may now be swapped.
<b>Circumstances:</b>	Application dependent.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID of issuing address space, or zero. Reg 0, byte 3: Sysevent code.
<b>Outputs:</b>	Reg 1, byte 3: Contains X'04' if request is not for the current address space; contains X'00' if the request to mark the address space as swappable was honored; contains X'08' if request was not authorized.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 21 of 24)

## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	2B
<b>Mnemonic:</b>	REQSWAP
<b>Meaning of Mnemonic:</b>	Request to swap out address space.
<b>Purpose:</b>	A particular address space is required to be swapped out.
<b>Circumstances:</b>	An address space is being requested to release the real storage frames it currently occupies. At the time of the subsequent swap-in, the Real Storage Manager re-allocates real storage frames to the swapped-in address space.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: Address of ECB to be posted.
<b>Outputs:</b>	Reg 1, byte 3: Contains X'00' if the swap-out request was honored; X'04' if the request was ignored because of the non-swappable status of the indicated address space; contains X'08' if address space is being swapped out.

---

<b>Sysevent Code:</b>	2C
<b>Mnemonic:</b>	BRINGIN
<b>Meaning of Mnemonic:</b>	Request to swap in address space.
<b>Purpose:</b>	A particular address space is required to be swapped in.
<b>Circumstances:</b>	The current job in this address space has been canceled. If BRINGIN were not issued, an address space that had been swapped out because of a shortage might be kept out until the shortage had been relieved.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code.
<b>Outputs:</b>	Reg 1, byte 3: Contains X'00' if the swap-in request was honored; X'08' if address space is currently in the process of being swapped.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 22 of 24)



## The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	2D
<b>Mnemonic:</b>	WKLDINIT
<b>Meaning of Mnemonic:</b>	Workload activity measurement initialization.
<b>Purpose:</b>	Permits MF/1 and RMF to inform the SRM to start collecting workload activity data by Performance Group Period (PGP), and provides the buffer used for data collection.
<b>Circumstances:</b>	MF/1 or RMF workload activity measurements are being activated.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: The address of a global data collection buffer.
<b>Outputs:</b>	Reg 1, bytes 0-3: 0 if data collection is successfully initialized. Reg 15, byte 3: X'00' if the request was honored, and no exception conditions were found; X'08' if a request to start workload activity data collection was rejected because of an incorrect buffer size; X'20' if data collection is already active.

---

<b>Sysevent Code:</b>	2E
<b>Mnemonic:</b>	WKLDCOLL
<b>Meaning of Mnemonic:</b>	Workload activity measurement collection.
<b>Purpose:</b>	Permits MF/1 and RMF to retrieve a copy of the data collected since the previous WKLDCOLL or WKLDINIT sysevent (code 2D). If, however, the IPS has changed, MF/1 or RMF will issue the WKLDTERM sysevent (code 2F), followed by WKLDINIT to provide a different data collection area.
<b>Circumstances:</b>	End of an MF/1 or RMF measurement interval.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: The address of a fixed buffer into which the collected workload activity measurements are to be copied.
<b>Outputs:</b>	Reg 1, bytes 0-3: Unchanged Reg 15, byte 3: X'00' if the request was honored, and no exception conditions were found; X'04' if previously started workload activity data collection has been stopped because of an IPS change; X'40' if the data collection buffer has not been established.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 23 of 24)

### The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	2F
<b>Mnemonic:</b>	WKL DTERM
<b>Meaning of Mnemonic:</b>	Workload activity measurement termination.
<b>Purpose:</b>	Permits MF/1 and RMF to inform the SRM to stop collecting workload activity data, and retrieve the buffer used for data collection.
<b>Circumstances:</b>	MF/1 or RMF workload activity measurements are being terminated, or an IPS change has occurred.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code. Reg 1, bytes 0-3: Zero.
<b>Outputs:</b>	Reg 1, bytes 0-3: The address of the global, fixed, workload activity data collection buffer that is no longer being used by the SRM.  Reg 15, byte 3: X'00' if the request was honored, and no exception conditions were found; X'40' if the data collection buffer has not been established.

---

<b>Sysevent Code:</b>	30 (hex)
<b>Mnemonic:</b>	None
<b>Purpose:</b>	Issued by the SRM itself in order to invoke its control routine immediately without waiting for a sysevent issued by another component.
<b>Inputs:</b>	Reg 0, bytes 0-1: ASID or zero. Reg 0, byte 3: Sysevent code. Reg 1, bytes 1-3: Address of system resource block under which this sysevent is issued.
<b>Outputs:</b>	None.

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 24 of 24)

### The Use of GTF to Track Sysevents (continued)

---

<b>Sysevent Code:</b>	31 (hex)
<b>Mnemonic:</b>	REQSVDAT
<b>Meaning of Mnemonic:</b>	Request Service Data.
<b>Purpose:</b>	Permits SMF to obtain service-related data for a given address space.
<b>Circumstances:</b>	SMF issues REQSVDAT during job or session termination.
<b>Inputs:</b>	<p>Reg 0, bytes 0-1: ASID or 0.            Reg 0, byte 3: Sysevent code.            Reg 1, bytes 0-3: The address of a 4-word fixed area where the service data is to be stored.</p>
<b>Outputs:</b>	<p>The 4-word area contains:</p> <p>Word 1 - Total service for the job or session.            Word 2 - Total transaction active time.            Word 3:                Bytes 0-1: Performance group number last assigned to the address space.                Bytes 2-3: For a TSO address space, the total number of transactions. For other address spaces, 0.            Word 4 - Total job or session residence time.</p> <p>Reg 15, byte 3:                00 - Normal completion.                04 - Data was lost due to accumulation control block error.</p>

---

Figure 5-3. Descriptions of SYSEVENT Codes (Part 24.1 of 24)



### Examples of Some Sysevent Data Reductions

Figures 5-4 through 5-8 illustrate some of the data reductions that can be obtained from the sysevent trace records. Figure 5-4 shows the result of sorting the sysevent file by ASID (address space ID), and within each ASID by time-of-day (TOD).

This example displays the sequence of sysevents that are issued for the running of jobs in address space 6, although the complete report could include sysevents for all active address spaces. Note that at relative TOD 24.285 the address space was created (MEMCREAT issued), and that at relative TOD 25.897 the job EDIT was selected. You may also note that ASID 6 was idle for 68.9 seconds before it was used again to run the job SORT51.

REPORT1 LISTING OF SYSEVENTS BY ASID PAGE 4													
ASID	SYSEVENT CODE (HEX)	SYSEVENT NAME	DELTA TOD	TOD	RELATIVE TOD	TOD	JOB NAME	R0-H2	RI-H1	RI-H2	R15 (HEX)	R0 (HEX)	R1 (HEX)
6	06	MEMCREAT	0.000000	24.285146	2498965.793278			6	0	1	01000000	00060006	00000001
6	2A	OKSWAP	0.035247	24.320393	2498965.828525			42	197	-3584	01000000	0006002A	00C5F200
6	1E	VERIFYPG	1.012148	25.332542	2498966.840673			30	0	256	01000000	0006001E	00000100
6	08	JOBSELECT	0.046640	25.379182	2498966.887313		INIT	8	254	-29344	01000000	00060108	00FE8D60
6	1C	DEVALLOC	0.116886	25.496067	2498967.004199			28	197	-13264	01000000	0006001C	00C5CC30
6	0A	INITATT	0.105569	25.601636	2498967.109768			10	0	112	01000000	0006000A	00000070
6	29	DONTSWAP	0.007375	25.609011	2498967.117143			41	255	8448	01000000	00060029	00FE2100
6	2A	OKSWAP	0.002777	25.611788	2498967.119920			42	255	-11776	01000000	0006002A	00FFD200
6	1E	VERIFYPG	0.207558	25.819346	2498967.327478			30	0	256	01000000	0006001E	00000100
6	29	DONTSWAP	0.046022	25.865368	2498967.373500			41	254	-30208	01000000	00060029	00FE8A00
6	2A	OKSWAP	0.002342	25.867710	2498967.375842			42	255	4352	01000000	0006002A	00FF1100
6	08	JOBSELECT	0.029490	25.897200	2498967.405332		EDIT	8	254	-30032	01000000	00060108	00FE8A00
6	1C	DEVALLOC	0.260198	26.157398	2498967.665530			28	196	27176	01000000	0006001C	00C46A28
6	0A	INITATT	0.136110	26.293508	2498967.801640			11	0	368	01000000	0006000A	00000170
6	0B	INITDET	11.004691	37.298199	2498978.806330			11	0	112	01000000	0006000B	00000070
6	27	REQGDAT	0.873976	38.172174	2498979.680306			39	196	24616	01000000	00060027	00C46028
6	31	REQSVDAT	0.052880	38.225054	2498979.733186			38	197	-16016	01000000	00060031	00C5C170
6	09	JOBTERM	0.551330	38.776384	2498980.284516			9	254	-30032	00000000	00060009	00FE8A00
6	29	DONTSWAP	0.004897	38.781281	2498980.289412			41	254	-30208	01000000	00060029	00FE8A00
6	2A	OKSWAP	0.004888	38.786169	2498980.294300			42	255	-11776	01000000	0006002A	00FFD200

Figure 5-4. Sysevents Sorted by Address Space ID and by Time of Day (Part 1 of 2)

REPORT1 LISTING OF SYSEVENTS BY ASID PAGE 5

ASID	CODE (HEX)	SYSEVENT NAME	DELTA	TOD	RELATIVE TOD	TOD	JOB NAME	R0-H2	RL-H1	RL-H2	R15 (HEX)	R0 (HEX)	RL (HEX)
6	0B	INITDET	0.229847	39.016016	2498980.524147			11	0	127	01000000	006000B	0000007F
6	09	JOBTM	1.185431	40.201447	2498981.709579			9	254	-29344	00000000	0060009	00FE8D60
6	29	DONTM	0.267263	40.468710	2498981.976842			41	254	-29440	01000000	0060029	00FE8D60
6	2A	OKSWAP	0.004881	40.473591	2498981.981723			42	255	4352	01000000	006002A	00FF1100
6	07	MEMREAT	0.787074	41.260664	2498982.768796			7	254	-29692	01000000	0060007	00FE8C04
6	0A	MEMREAT	68.936987	110.197651	2499051.705783			6	0	1	01000000	0060006	00000001
6	2A	OKSWAP	0.038446	110.236097	2499051.744229			42	197	-3584	01000000	006002A	00C5F200
6	1E	VERIFYPG	1.033204	111.269301	2499052.774733			30	0	256	01000000	006001E	00000100
6	08	JOBSELC	0.044454	111.313755	2499052.821887		INIT	8	254	-29448	01000000	0060108	00FE8CF8
6	1C	DEVALLOC	0.116931	111.430686	2499052.938818			28	197	-13264	01000000	006001C	00C5CC30
6	0A	INITATT	0.105535	111.536221	2499053.044353			10	0	112	01000000	006000A	00000070
6	29	DONTM	0.007381	111.543602	2499053.051734			41	255	8448	01000000	0060029	00FF2100
6	2A	OKSWAP	0.003306	111.546908	2499053.055040			42	255	-11776	01000000	006002A	00FFD200
6	1E	VERIFYPG	0.208044	111.754952	2499053.263084			30	0	256	01000000	006001E	00000100
6	29	DONTM	0.327362	112.082314	2499053.590446			41	254	-30464	01000000	0060029	00FE8900
6	2A	OKSWAP	0.002291	112.084605	2499053.592737			42	255	4352	01000000	006002A	00FF1100
6	08	JOBSELC	0.014430	112.099035	2499053.607167		SORT51	8	254	30256	01000000	0060108	00FE89D0
6	1C	DEVALLOC	0.554243	112.653278	2499054.161409			28	196	26648	01000000	006001C	00C46818
6	1C	DEVALLOC	0.374354	113.027631	2499054.535763			28	196	26648	01000000	006001C	00C46818
6	1C	DEVALLOC	0.196342	113.223973	2499054.732105			28	196	26648	01000000	006001C	00C46818
6	1C	DEVALLOC	0.256034	113.480007	2499054.988139			28	196	26648	01000000	006001C	00C46818
6	1C	DEVALLOC	0.203693	113.683700	2499055.191832			28	196	26648	01000000	006001C	00C46818
6	1C	DEVALLOC	0.169903	113.853603	2499055.361735			28	196	26648	01000000	006001C	00C46818
6	0A	INITATT	25.951629	139.955766	2499055.463898			10	0	368	01000000	006000A	00000170
6	0B	INITDET	3.094599	143.050365	2499059.584997			11	0	112	01000000	006000B	00000070
6	27	REQGDAT	3.096372	146.356737	2499087.864869			39	196	28000	01000000	0060027	00C46D60
6	1C	DEVALLOC	0.377081	158.021174	2499100.075646			28	196	27944	01000000	006001C	00C46D28
6	1C	DEVALLOC	0.377081	158.021174	2499100.075646			28	196	27944	01000000	006001C	00C46D28
6	1C	DEVALLOC	0.319020	146.675757	2499088.183889			28	196	27944	01000000	006001C	00C46D28
6	1C	DEVALLOC	0.706039	147.381796	2499088.899928			28	196	27944	01000000	006001C	00C46D28
6	0A	INITATT	2.475548	147.629344	2499089.137475			10	0	368	01000000	006000A	00000170
6	0B	INITDET	9.218179	156.847522	2499098.355654			11	0	112	01000000	006000B	00000070
6	27	REQGDAT	0.796571	157.644093	2499099.152225			39	196	24960	01000000	0060027	00C46180
6	1C	DEVALLOC	0.546341	158.567514	2499100.075646			28	196	24944	01000000	006001C	00C46170
6	1C	DEVALLOC	0.377081	158.567514	2499100.075646			28	196	24944	01000000	006001C	00C46170
6	0A	INITATT	0.285603	158.853117	2499100.361249			10	0	368	01000000	006000A	00000170
6	0C	QSCCEM	19.132159	177.985276	2499119.493408			12	-32768	0	00000000	006000C	80000300
6	0D	QSCCEM	0.066583	178.051859	2499119.559991			13	255	-32372	01000000	006000F	00FE818C
6	0F	SWOUTCMT	0.373022	178.424881	2499119.933013			15	255	-32372	01000000	006000F	00FE818C
6	10	SWPINST	0.023114	178.447995	2499119.956127			16	0	1	01000000	0060010	00000001
6	10	SWPINST	0.168238	178.616733	2499120.124365			16	0	1	01000000	0060010	00000002
6	13	RSTORCMT	0.669550	179.285784	2499120.793915			19	0	0	00000000	0060013	00000000
6	0B	INITDET	5.067999	184.353783	2499125.861914			11	0	112	01000000	006000B	00000070
6	27	REQGDAT	0.525064	184.878846	2499126.386978			39	196	25088	01000000	0060027	00C46200
6	1C	DEVALLOC	0.734655	185.613501	2499127.121633			28	196	25848	01000000	006001C	00C464F8
6	1C	DEVALLOC	0.183022	185.796523	2499127.304655			28	196	25848	01000000	006001C	00C464F8
6	1C	DEVALLOC	0.188660	185.985183	2499127.493314			28	196	25848	01000000	006001C	00C464F8
6	1C	DEVALLOC	0.186157	186.171340	2499127.679471			28	196	25848	01000000	006001C	00C464F8
6	1C	DEVALLOC	0.190462	186.361802	2499127.869933			28	196	25848	01000000	006001C	00C464F8
6	1C	DEVALLOC	0.172484	186.534285	2499128.042417			28	196	25848	01000000	006001C	00C464F8
6	0A	INITATT	0.161511	186.695796	2499128.203928			12	0	368	01000000	006000A	00000170
6	0C	QSCCEM	20.740748	207.436545	2499148.944676			10	-32768	0	00000000	006000C	80000300
6	0D	QSCCEM	0.077732	207.514277	2499149.022408			13	0	0	00000000	006000D	00000300
6	0F	SWOUTCMT	0.381616	207.895892	2499149.404024			15	253	30632	01000000	006000F	00FD77A8

Figure 5-4. Sysevents Sorted by Address Space ID and by Time of Day (Part 2 of 2)

Figure 5-5 shows the number of times a particular sysevent was issued. It gives both the mean time and the time distribution between successive issuances. Consider sysevent code 1C (DEVALLOC). This sysevent was issued 282 times (there are 281 time intervals between successive issuances). The mean time between such events was 4.26 seconds. The distribution also shows that 61.1% of the intervals (60.1% + 0.3% + 0.7%) were within 0.5 seconds.

TIME BETWEEN SYSEVENTS OF SAME TYPE															
CODE	CODE NAME	COUNT REAL DELTA	M (REAL DELTA) DELTA TOD	PERCENTS DELTA TOD (REAL DELTA)											
				<= 0.001	<= 0.050	<= 0.100	<= 0.500	<= 1.000	<= 5.000	<= 10.000	<= 50.000	<= 100.000	>		
01	TIMEREXP	126	9.723068	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
03	NIOWAIT	18	65.380090	0.0	0.0	0.0	0.0	0.0	0.0	11.1	11.1	38.8	5.5	33.3	0.0
04	USERRDY	16	67.759794	0.0	0.0	0.0	0.0	0.0	0.0	6.2	12.5	43.7	0.0	37.5	0.0
06	MEMCREAT	2	45.301654	0.0	0.0	0.0	0.0	0.0	0.0	50.0	0.0	0.0	50.0	0.0	0.0
07	MEMDEL	0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
08	JOBSELC	23	51.193591	0.0	0.0	0.0	4.3	8.6	13.0	0.0	0.0	34.7	30.4	8.6	0.0
09	JOBTERM	20	59.802141	0.0	0.0	0.0	0.0	4.9	9.9	0.0	0.0	44.9	29.9	9.9	0.0
0A	INITAFT	93	12.884177	0.0	0.0	0.0	1.0	7.5	13.9	35.4	40.8	1.0	0.0	0.0	0.0
0B	INITDET	91	13.186660	0.0	0.0	0.0	2.1	5.4	12.0	35.1	43.9	0.0	1.0	0.0	0.0
0C	QCEST	36	28.502030	0.0	16.6	5.5	0.0	0.0	0.0	2.7	8.3	52.7	11.1	2.7	0.0
0D	QCECMP	42	28.666021	0.0	9.5	9.5	0.0	2.3	4.7	7.1	57.1	4.7	4.7	4.7	0.0
0F	SWOUTCMP	41	29.362830	0.0	0.0	0.0	0.0	19.5	2.4	2.4	7.3	56.0	7.3	4.8	0.0
10	SWINSTAT	82	28.643126	0.0	0.0	13.6	37.4	20.4	8.6	4.0	0.0	0.0	0.0	0.0	0.0
13	RSTORCMP	41	25.037034	0.0	9.7	0.0	12.1	4.8	4.8	14.6	41.4	9.7	2.4	2.4	0.0
14	ENHOLD	9	76.807824	0.0	0.0	0.0	0.0	0.0	0.0	11.1	22.2	55.5	0.0	11.1	0.0
15	ENORLSE	9	76.792479	0.0	0.0	0.0	0.0	0.0	0.0	11.1	22.2	55.5	0.0	11.1	0.0
17	AVQLOW	5	216.231253	0.0	0.0	0.0	0.0	0.0	0.0	0.0	19.9	39.9	19.9	19.9	0.0
18	AVQOK	5	216.232278	0.0	0.0	0.0	0.0	0.0	0.0	0.0	19.9	39.9	19.9	19.9	0.0
1C	DEVALLOC	281	4.263045	0.0	0.7	0.3	60.1	7.4	6.4	11.7	12.8	0.3	0.0	0.0	0.0
1E	VERIFYPG	23	51.175377	0.0	0.0	0.0	13.0	4.3	8.6	0.0	34.7	30.4	8.6	8.6	0.0
31	REQSVDAT	20	60.007462	0.0	0.0	0.0	0.0	4.9	9.9	0.0	44.9	29.9	9.9	9.9	0.0
27	REQPGDAT	90	13.332241	0.0	0.0	0.0	3.3	3.3	14.4	33.3	44.4	0.0	0.0	1.1	0.0
29	DONTSWAP	47	25.759186	0.0	0.0	0.0	14.8	21.2	23.4	0.0	21.2	14.8	4.2	4.2	0.0
2A	OKSWAP	50	24.213887	0.0	5.9	0.0	15.9	17.9	21.9	0.0	19.9	13.9	3.9	3.9	0.0
30	SRB	114	10.826754	0.0	0.8	0.0	8.7	7.0	17.5	9.6	56.1	0.0	0.0	0.0	0.0

Figure 5-5. Time Distribution Between Sysevents of Same Type

The report in Figure 5-6 shows the distribution of swapout time. It was produced by selecting records from the sysevent trace data only for code 0D (QUIESCE COMPLETED) and code 0F (SWAPOUT COMPLETED). There were 36 code-0D records and all 36 resulted in completed swapouts. The mean time to complete a swapout (the time between QUIESCE COMPLETED and SWAPOUT COMPLETED) was 0.27 seconds. Note that 66.6% of the swapouts completed within 0.30 seconds. Note also that 11% of the swapouts took longer than 0.5 second.

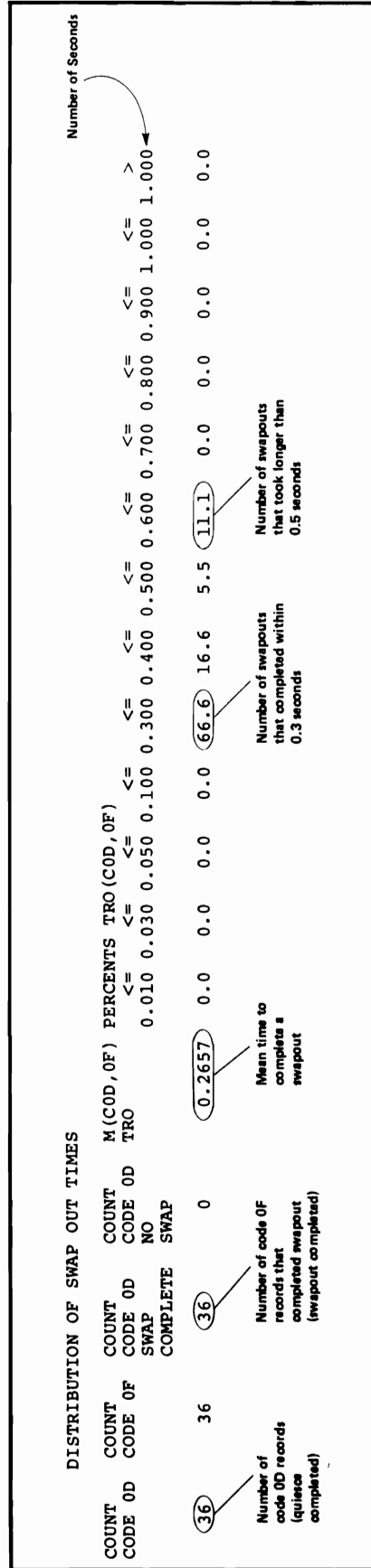


Figure 5-6. Distribution of Time Needed to Complete a Swapout



Figure 5-7 depicts a listing that shows real frame shortages, as indicated by occurrences of AVQLOW (code X'17') and AVQOK (code X'18') records sorted by relative time of day. The coded numeral '1' in the last column indicates an AVQLOW non-critical threshold, and the numeral '2' indicates an AVQLOW critical threshold. Note that at relative TOD 0.0 a non-critical AVQLOW occurred, followed by five critical AVQLOWs. Finally at relative time 0.65 an AVQOK occurred, signifying a temporary end to the real frame shortage.

LIST OF SYSTEM RECORDS				
SYSEVENT CODE (HEX)	CODE NAME	DELTA TOD	RELATIVE TOD	R1,B3 (indicator)
17	AVQLOW	0.000000	0.000000	1
17	AVQLOW	0.139316	0.139316	2
17	AVQLOW	0.045019	0.184335	2
17	AVQLOW	0.041238	0.225573	2
17	AVQLOW	0.013628	0.239201	2
17	AVQLOW	0.402581	0.641782	2
18	AVQOK	0.010584	0.652366	
17	AVQLOW	0.660104	1.312469	1
17	AVQLOW	0.460672	1.773141	2
18	AVQOK	0.007536	1.780677	
17	AVQLOW	5.513137	7.293814	1
17	AVQLOW	0.262811	7.556625	2
18	AVQOK	0.005968	7.562593	
17	AVQLOW	0.275185	7.837777	1
18	AVQOK	0.927716	8.765494	

Figure 5-7. Instances of Real Frame Shortages, Distributed by Time of Day

A reasonable question to be asked is: How long did the real frame shortage last? This question can be answered by means of pencil, paper, and the report in Figure 5-7, or by use of the reduction program to produce the report shown in Figure 5-8. In Figure 5-8 four non-critical AVQLOWs were issued and seven critical AVQLOWs were issued. The *average* amount of time spent at AVQLOW was approximately 0.58 seconds, with all durations being less than 1 second. During this analysis period the *total* time spent at AVQLOW was 2.3 seconds.

DISTRIBUTION OF AVQLOW TIMES												
COUNT C17	COUNT C17C	COUNT C18	M (C17-C18) AVQTIME (AVERAGE)	PERCENTS <= 0.1	<= 0.3	<= 0.5	<= 0.7	<= 1	<= 300	> 300	MAX RELATIVE TOD	T (C17-C18) AVQTIME (TOTAL)
4	7	4	0.5793	0.0	25.0	25.0	25.0	25.0	0.0	0.0	8.765494	2.3171

Figure 5-8. Distribution of Real Frame Shortages

## TCAM Tuning Considerations

This section describes special considerations for page fixing TCAM in areas containing TCAM CSECTs, modules, and control blocks.

It covers three topics related to page fixing and page faults:

- Packaging the MCP to Minimize Page Fixes and Page Faults
- Coding INTRO Operands to Minimize Fixed Pages
- Ordering of OPENs to Minimize Fixed Pages for LCBs and SCBs.

For additional performance guidelines on TCAM, refer to the *OS/VS2 TCAM Programmer's Guide* and the *OS/VS TCAM User's Guide (GC30-2025)* which give guidelines on efficient section ordering of the Message Control Program (MCP) and explanations of space utilization caused by selecting certain TCAM options. These guidelines and explanations should help minimize the number of fixed pages required in the TCAM system, and also minimize the number of page faults.

### Packaging the MCP to Minimize Page Fixes and Page Faults

The parmlib member, IEAPAK00, can be used to group TCAM modules in the pageable link pack area (PLPA). The user can reduce page faults by grouping modules that refer to each other during execution. Suggested groupings are:

- TCAM open modules
- TCAM close modules
- TCAM error recovery modules.

In addition, TCAM operator control modules can be grouped in the PLPA according to those that are most often used. For further information on IEAPAK00, see "Part 2: System Initialization – PARMLIB Members" in this manual. For information on the PLPA, see the performance topic, "The Pageable Link Pack Area: Its Advantages and Uses" in Part 5. The following control blocks and modules will be fixed by TCAM and should be grouped by the linkage editor ORDER control statement as described below:

- Control blocks:

Assembled in the MCP –	<u>CSECT</u>
Address Vector Table (AVT)	MCP
Data Control Blocks (DCBs)	MCP
Device Characteristics Table	IEDQSTCS
Invitation Lists	MCP
Option Tables	IEDQOPC, IEDQOPT
Queue Control Blocks (QCBs)	IEDQQCBC
Station Control Blocks (SCBs)	IEDQSCBC
Terminal Entries	IEDQTRMC
Terminal Name Table (TNT)	IEDQTNT

## Packaging the MCP to Minimize Page Fixes and Page Faults (continued)

Dynamically Acquired (via GETMAIN):

- Buffer Units
- Channel Program Blocks (CPBs)
- Dial SCBs
- Line Control Blocks (LCBs)

- Loaded modules:

- Attention Handler Routine
- Channel End Appendage
- I/O Trace Table
- PCI Appendage
- Special Characters Table
- Start I/O Appendage

The module names for the TCAM routines listed above can be found in *OS/VS2 TCAM Logic*, SY30-2040).

The linkage editor ORDER control statement can be used to cause the TCAM MCP to be loaded on a page boundary and to group CSECTs of the MCP that are fixed. The ORDER statement can also be used to group modules in the MCP according to their use. Section 4 of *OS/VS2 TCAM Logic* correlates TCAM modules to TCAM function in the MCP. The user can determine the order in which his MCP calls on TCAM modules and then use the ORDER statements to cause TCAM to order the MCP accordingly, thus reducing the number of page faults between execution of different modules. The following example shows how the ORDER statement can be used to group CSECTs and modules so that the number of page fixes is minimized:

```
ORDER MCP (P)      } (P) requests page boundary alignment by the linkage
                   } editor.
ORDER IEDQQCBC
ORDER IEDQTNT
ORDER IEDQTRMC
ORDER IEDQOPT      } IEDQOPT and IEDQOPC are in the MCP only if the
ORDER IEDQOPC      } OPTION facility is used.
ORDER IEDQSTCS
ORDER IEDQSCBC
ORDER IEDAYZ        } IEDAYZ and IEDQKA02 are page fixed only if TSO
ORDER IEDQKA02     } is included in a TCAM system running with an IBM
                   } 2701, 2702, or 2703 control unit.
ENTRY MCP
```

## Coding INTRO Operands to Minimize Fixed Pages

By carefully selecting operands of the INTRO macro instruction, the user can minimize the number of fixed pages required for the following if real storage size is critical:

- Station Control Blocks (SCBs).
- I/O Trace Table.
- Buffer Units.
- Channel Program Blocks (CPBs).

The pages for these control blocks are obtained and fixed during TCAM initialization.

The following operands of the INTRO macro instruction determine the number of pages that are fixed:

USEREG=  
UNITSZ=, KEYLEN= (See note below)  
LNUNITS=  
MSUNITS=  
TRACE=  
CPB=

### Station Control Blocks

The USEREG= operand of INTRO specifies the number of registers that can be saved in a station control block (SCB). SCBs that are assembled in the MCP are fixed in the IEDQSCBC CSECT. The size of an SCB is eighty-four bytes plus four bytes for each register saved. The size of IEDQSCBC can thus be unnecessarily increased if the user creates register save areas in each SCB when he does not need them.

---

*Note:* KEYLEN= and UNITSZ= are mutually exclusive. UNITSZ= is the more preferred term and will be used for the remainder of this topic.

## I/O Trace Table, Buffer Units, and Channel Program Blocks

Space is allocated for the I/O trace table, the buffer units, and the CPBs by one GETMAIN macro. The GETMAIN is for the number of pages (4096 bytes per page) needed to contain all of the control blocks. The user can determine the number of pages required as follows:

1. Determine the number of pages needed to contain the I/O trace table. Specify this storage by the TRACE= operand. Each trace entry is sixteen bytes long. In addition, thirty-two bytes of control information are required for the entire table. Therefore:

$$\text{Trace table size} = 16 (\text{number of trace entries}) + 32$$

2. If any space remains in the last page of the I/O trace table, use it for buffer units as long as each unit fits completely within the page. When the space in the page is insufficient, specify additional pages as described below. In each additional page, the last buffer unit must fit completely within the page; that is, it must not cross a page boundary.

The amount of storage for the buffer units is specified by the UNITSZ=, LNUNITS=, and MSUNITS= operands. The total number of units is equal to the sum of LNUNITS and MSUNITS. The size of each unit is found by adding twelve (or twenty if TCAM is in a VTAM system) to the UNITSZ=value. The extra bytes are used by TCAM for internal control information. If the buffer unit size is not a multiple of a double word, the value will be rounded up to the next double word.

3. If any space remains in the last page needed by the buffer units, use it for the CPBs. Additional pages may be needed to contain all the CPBs. As with buffer units, CPBs cannot cross a page boundary.

The amount of storage required for CPBs is determined by the CPB= and UNITSZ= operands. The CPB= operand gives the number of CPBs. The size of each CPB is found by adding seventy-two (72) to the buffer unit size.

When the number of pages required has been determined, then the GETMAIN for the storage is issued. The space within the pages is allocated to the control blocks and initialized. Any space within a page that is not used because a buffer unit or CPB would not fit completely is freed. The address of the storage and its length is saved for the page fix which will be issued during the TCAM OPEN macro execution.

The following example shows how the proper choice of INTRO operand values can help minimize the number of pages that are fixed by TCAM. This example is for TCAM in a non-VTAM environment.

Initial Case		
LNUNITS=	50	MSUNITS= 21
TRACE=	200	UNITSZ= 116
CPB=	20	
Page	Allocation	Bytes
1	200 trace entries	3232
	6 buffer units	768
	unused	96
2	32 buffer units	4096
3	32 buffer units	4096
4	1 buffer unit	128
	19 CPBs	3800
	unused	168
5	1 CPB	200
	unused	3896
Total Pages Fixed:		5
Total Unused Bytes:		4160

Change 1		
LNUNITS=	50	MSUNITS= 21
TRACE=	198	UNITSZ= 116
CPB=	20	
Page	Allocation	Bytes
1	198 trace entries	3200
	7 buffer units	896
2	32 buffer units	4096
3	32 buffer units	4096
4	20 CPBs	4000
	unused	96
Total Pages Fixed:		4
Total Unused Bytes:		96

Change 2		
LNUNITS=	49	MSUNITS= 21
TRACE=	200	UNITSZ= 116
CPB=	20	
Page	Allocation	Bytes
1	200 trace entries	3232
	6 buffer units	768
	unused	96
2	32 buffer units	4096
3	32 buffer units	4096
4	20 CPBs	4000
	unused	96
Total Pages Fixed:		4
Total Unused Bytes:		192

Change 3		
LNUNITS=	50	MSUNITS= 21
TRACE=	200	UNITSZ= 116
CPB=	19	
Page	Allocation	Bytes
1	200 trace entries	3232
	6 buffer units	768
	unused	96
2	32 buffer units	4096
3	32 buffer units	4096
4	1 buffer units	128
	19 CPBs	3800
	unused	168
Total Pages Fixed:		4
Total Unused Bytes:		264

In the initial case, five pages are needed to contain the control blocks. However, 3896 bytes of the last page are not used even though the entire page is fixed. Changes 1, 2, and 3 show that by altering the INTRO operands TRACE=, LNUNITS=, and CPB= respectively, the fifth page is not needed.

Figure 5-9 shows the number and size of both the buffer units and CPBs available per page in relation to the UNITSZ=operand for a non-VTAM environment. It also shows the number of units CPBs that can be fixed on a page, along with the remaining unused portion, depending on the various values of UNITSZ. For further information on the SCB I/O trace table, buffer units, and channel program blocks, see *OS/VS2 TCAM Logic*, SY30-2059.

UNITSZ=	Buffer Units			CPBs		
	Length	Number Per Page	Unused	Length	Number Per Page	Unused
36	48	85	16	120	34	16
44	56	73	8	128	32	0
52	64	64	0	136	30	16
60	72	56	64	144	28	64
68	80	51	16	152	26	144
76	88	46	48	160	25	96
84	96	42	64	168	24	64
92	104	39	40	176	23	48
100	112	36	64	184	22	48
108	120	34	16	192	21	64
116	128	32	0	200	20	96
124	136	30	16	208	19	144
132	144	28	64	216	18	208
140	152	26	144	224	18	64
148	160	25	96	232	17	152
156	168	24	64	240	17	16
164	176	23	48	248	16	128
172	184	22	48	256	16	0
180	192	21	64	264	15	136
188	200	20	96	272	15	16
196	208	19	144	280	14	176
204	216	18	208	288	14	64
212	224	18	64	296	13	248
220	232	17	152	304	13	144
228	240	17	16	312	13	40
236	248	16	128	320	12	256
244	256	16	0	328	12	160
252	264	15	136	336	12	64

Figure 5-9. Available Buffer Units and CPBs According to UNITSZ Operand (for a non-VTAM environment)

### Ordering OPEN Macros to Minimize Fixed Pages for LCBs and SCBs

The user can significantly reduce the number of fixed pages required for Line Control Blocks (LCBs) and Station Control Blocks (SCBs) through efficient ordering of the TCAM OPEN macro instructions. These control blocks are obtained by TCAM during OPEN execution of line group DCBs.

- One LCB is obtained for each line in a line group.
- One SCB is obtained for each line in a dial line group.

The table in Figure 5-10 indicates the size of an LCB for each terminal type and the maximum number of lines in a group that will fit into two pages of storage.

Terminal Type	LCB Size	Maximum Number of LCBs in 2 Pages
2260 Local	200	39
3270 Local	200	39
7770	216	36
1050	216	36
1030	216	36
115A	216	36
2740	216	36
2741	216	36
115A-WTTA	224	35
5041	216	36
TWX	224	35
WTT	224	35
2740 Dial	232	34
1050 Dial	240	33
2740 Contention	240	33
Autopoll: start/stop	248	32
BSC	248	32

**Note:** The LCBs for a line group are allocated contiguously in storage. Contiguous to the allocated storage for LCBs are eight bytes of storage for each of the LCBs in the line group. This storage is used to contain the threshold values for each line. (See the THRESH= operand of the INTRO macro.)

**Figure 5-10. LCB Sizes By Terminal Type**

The following restrictions and techniques will affect the allocation of LCBs and SCBs.

### Restrictions

The Channel Program Area (CPA) of the LCB must be in fixed storage and cannot cross a page boundary. The CPA begins at offset 144 (X'90') in the LCB and extends to the end of the LCB. For all LCB sizes, this restriction is not a problem when the LCBs for a group can fit into one or two pages. (The LCBs for a line group must be contiguous in storage.) If more than two pages are needed, TCAM imposes the following restrictions (except for 2260L and 3270L line groups):

- The LCB size is changed to 248.
- Only 132 lines are allowed in the line group.
- The first LCB for the group is allocated at an offset of 24 in the first page. This offset value, called the alignment, ensures that up to 132 LCBs can be allocated without a CPA crossing a page boundary.



## Techniques

TCAM issues GETMAIN macros in page increments for LCBs and SCBs. The storage in a page that is not allocated for control blocks is referred to as “available storage”. It is chained with other blocks of available storage. The first word of each available storage block points to the next available storage block, or to zero if it is the last block in the chain. The second word contains the length of the available storage block. The pointer to the first available storage block is kept in the TCAM CVT extension control block (TCX). If the LCBs for a line group will fit into an available storage block, then the block is used. What is not used remains in the chain of available storage. If all of the available storage in the page is used, then that page is removed from the chain. If no available storage block is large enough to contain the requested space, then a GETMAIN for the number of pages needed is issued. If after allocating space in the acquired page or pages there is storage left over, it is added to the “available storage” chain.

Dial line groups require additional processing. Preceding the space for the line group are eight bytes of control information. If the line group requires more than two pages for LCBs, an alignment value of 16 will be issued. The 16 byte alignment, plus the eight bytes of control information, will assure that the CPAs do not cross page boundaries for up to 132 lines. Also, an SCB will be obtained for each line in the group. The length of an SCB, in bytes, is:

length of SCB= 84+ 4(USEREG), where USEREG is the value of the USEREG= operand of the INTRO macro.

The SCBs for a line group are obtained as a contiguous block of storage and will be allocated in the LCB available storage, if possible. To ensure that a minimal number of pages are fixed for LCBs and SCBs, open the largest line groups first (that is, the ones with the most lines).

**This page leaf represents page 302 which was deleted by Supervisor Performance \*2.**

## Miscellaneous Performance Guidelines

The following tuning guidelines are reprinted from an Installation Newsletter.

- *Data Set Placement:* Locate the VTOC near the center of the pack, and place heavily used data sets close to the VTOC. Note that the relative importance (frequency of use) of system data sets has changed from that in OS/MVT. For example, SYS1.SVCLIB and SYS1.LINKLIB had high frequency usage in OS/MVT. In MVS, the modules previously located on these data sets now reside in SYS1.LPALIB. The SYSRES data set placement used for benchmark measurement runs was:

Contents	Cylinder
SYS1.LPALIB	(1-36)
SYS1.PARMLIB	(37-46)
SYS1.DUMPO0	(47-52)
SYS1.DUMPO1	(53-56)
SYS1.DSSVM	(57-59)
SYS1.NUCLEUS	(66-80)
SYS1.DCMLIB	(81-82)
SYS1.LOGON	(136-137)
SYS1.SVCLIB	(138-142)
SYS1.CMDLIB	(143-152)
SYS1.MACLIB	(153-188)
SYS1.PROCLIB	(189-194)
SYS1.UADS	(195-198)
SYS1.BROADCAST	(199-199)
VTOC	(200-200)
CATALOG	(201-225)
SYS1.STGINDEX	(226-231)
SYS1.LINKLIB	(232-271)
SYS1.MANX	(272-281)
SYS1.MANY	(282-291)
SYS1.HELP	(388-388)
SYS1.TELCMLIB	(391-398)
SYS1.IMAGELIB	(399-399)
SYS1.LOGREC	(400-401)

- *Program Product Region Sizes:* Specify a region size in the JCL for all Program Products which use GETMAIN to obtain all available storage. This will prevent the program from getting eight megabytes of virtual storage.
- *Use of Initiators:* Do not run in a highly overinitiated environment. Start at a conservative initiator level for example, and work up to the number of initiators that gives the balance of throughput and distribution of service that is desired.

### Miscellaneous Performance Guidelines (continued)

- *Journaling*: Measurements have shown up to 34% (a significant) increase in overhead because of journaling when running VIO and a workload containing many job steps.

- \$\$Inn command 118 (VS2.03.807)
- ABEND dump parameters
  - as used with SYSABEND data set 48
  - as used with SYSUDUMP data set 58
  - need for care in changing them dynamically via the CHNGDUMP command 48-49
- absorption rate 175 (VS2.03.807)
- access method services
  - formatting data sets 102.2, 116 (VS2.03.807)
  - page space shortage 118 (VS2.03.807)
  - swap data set formatting 125.1 (VS2.03.807)
- active time of a transaction 174 (VS2.03.807)
- adding more paging space 158.11 (VS2.03.807)
- address space
  - importance 176, 177 (VS2.03.807)
  - relation to transaction 173 (VS2.03.807)
  - swapping 163 (VS2.03.807)
- address spaces, maximum number of concurrent 110
- algorithm
  - auxiliary storage shortage prevention 158.11 (VS2.03.807)
  - swap operations 158.2 (VS2.03.807)
- allocation, device, performance of (*see* device allocation performance)
- ALTCPREC sysevent 280
- APF parameter in IEASYSxx 92
- APG parameter in IEAIPSxx 202.19 (VS2.03.807) (*see also* automatic priority group) (VS2.03.807)
- APG parameter in IEASYSxx 93
- appendages, EXCP 53
- arm position information 158.2 (VS2.03.807)
- ASM
  - device usage 158.9 (VS2.03.807)
  - I/O load balancing 158.11 (VS2.03.807)
  - paging device contention 158.8 (VS2.03.807)
  - performance recommendations 158.8-158.10 (VS2.03.807)
  - questions and answers 158.11-158.12 (VS2.03.807)
  - slot selections 158.12 (VS2.03.807)
- ASMCNSTS sysevent 277
- ASSIGN request 158.4 (VS2.03.807)
- authorized program facility (APF) parmlib member 52
- automatic priority group (APG) 162, 164 (VS2.03.807)
- automatic priority group (APG) parameter in IEASYSxx 93
- auxiliary storage management initialization 158.1-158.12 (VS2.03.807)
- auxiliary storage shortage prevention 168, 158.11 (VS2.03.807)
- Auxiliary Storage User Pool, MF/1 report 158.10 (VS2.03.807)
- average number of ready users 201 (VS2.03.807)
- AVQLOW sysevent 275
- AVQOK sysevent 276
- batch, defining requirements for 189 (VS2.03.807)
- BLDL or BLDLF parameter in IEASYSxx 94
- BLDL list (IEABLDxx) 56
  - IBM-supplied default list 57
- bottleneck
  - service definition coefficients 202.15 (VS2.03.807)
  - SRM throughput control 162 (VS2.03.807)
- BUF parameter in SMFPRMxx 149
- buffer for permanent I/O errors 158.7 (VS2.03.807)
- BUFFERS TSO parameter 133
- BUFFERSPACE parameter, use of 257
- BUFSIZE TSO parameter 133
- catalog, VSAM
  - performance factor guidelines 257
  - questions and answers 260
- CCW strings 158.11 (VS2.03.807)
- channel activity report, MF/1
  - description of 212
  - how to use it 225
- channel command word strings 158.11 (VS2.03.807)
- channel contention 158.2 (VS2.03.807)
- CHNGDUMP command, precautions and example 48-49
- circular queue 158.9 (VS2.03.807)
- CLPA IPL duplexing 158.2 (VS2.03.807)
- CLPA parameter in IEASYSxx, meaning and use 95
- clumping of allocations 167 (VS2.03.807)
- CMD parameter in IEASYSxx (*see also* COMMNDxx) 97
- CNSTR parameter 202.19 (VS2.03.807)
- COMMNDxx
  - detailed description 41
  - synopsis 27
- common area pages 115 (VS2.03.807)
- competition for resources
  - dispatching priorities 162 (VS2.03.807)
  - exchange swaps 163 (VS2.03.807)
  - general discussion 162 (VS2.03.807)
  - multiple performance periods 186 (VS2.03.807)
  - performance objectives 178-181 (VS2.03.807)
  - performance objectives and ISVs 181-183 (VS2.03.807)
- concatenating libraries to LINKLIB 139
- concurrent paging 158.10 (VS2.03.807)
- CONFIGCH sysevent 278
- configuration, system, determining changes to 262
- contention
  - channel and device 158.2 (VS2.03.807)
  - for paging devices 158.8 (VS2.03.807)
- contention factors, system 166 (VS2.03.807)
- contention for storage, excessive 165 (VS2.03.807)
- contention index
  - definition 172 (VS2.03.807)
  - how to calculate 201 (VS2.03.807)
  - selecting domain weights 201 (VS2.03.807)
  - use of by SRM 172, 173 (VS2.03.807)
- contention, SRM throughput control 162 (VS2.03.807)
- CPU activity report
  - description of 210
  - how to use it 224

- CPU load adjusting
  - discussion 165 (VS2.03.807)
  - factor in swap decisions 183 (VS2.03.807)
- CPU
  - monitoring of usage 166 (VS2.03.807)
  - overutilized 166 (VS2.03.807)
  - task execution time 198 (VS2.03.807)
  - underutilized 166 (VS2.03.807)
- CPU parameter
  - resource factor coefficient 202.22 (VS2.03.807)
  - service definition coefficient 202.19 (VS2.03.807)
- CPU service, selecting coefficient value 198 (VS2.03.807)
- CPU service units 174 (VS2.03.807)
- crash, system, how to warm-start after 22
- critical storage shortages 168 (VS2.03.807)
- CSA
  - excluded from common areas 90, 158.1 (VS2.03.807)
  - paging data set size estimation 158.10 (VS2.03.807)
- CSA parameter in IEASYSxx 98
- CTRLPROG macro instruction 119 (VS2.03.807)
- current MPL 200 (VS2.03.807)
- cut-off workload level
  - definition 177 (VS2.03.807)
  - effect of 182 (VS2.03.807)
  - limit on load adjusting values 187 (VS2.03.807)
  - use of 202.1, 202.13 (VS2.03.807)
- CVIO
  - parameter 22 (VS2.03.807)
  - specification 158.9 (VS2.03.807)
- CVIO parameter in IEASYSxx 99
- CVOL usage (*see Using OS Catalog Management with the Master Catalog: CVOL Processor and VSAM Programmer's Guide*)
- DATASET macro instruction
  - DUPLEX parameter specification 102.2 (VS2.03.807)
  - for primary PLPA copy 158.8 (VS2.03.807)
  - PAGE parameter specification 115-116 (VS2.03.807)
  - SWAP parameter specification 125.1 (VS2.03.807)
- DATASET sysgen macro
  - rules and syntax example for specifying EXCP appendages 53
- DEFINE PAGESPACE command 40 (VS2.03.807)
- DEFINE PAGESPACE processor
  - general use 116 (VS2.03.807)
  - page space shortage 118 (VS2.03.807)
- defining page data sets 40 (VS2.03.807)
- detected wait swap count 202.12 (VS2.03.807)
- detection of wait states 164 (VS2.03.807)
- DEVALLOC sysevent 278
- device activity report (MF/1)
  - description of 215
  - how to use it 226
- device allocation performance 252
  - guidelines for improving allocation response 253
  - order in which requests are serviced 252
  - questions and answers 256
- device allocation, SRM 167 (VS2.03.807)
- device class selection 208
- device contention
  - ASM 158.8 (VS2.03.807)
  - service bursting 158.2 (VS2.03.807)
- device preference table, use of 253
- device speed tradeoffs for paging 158.9 (VS2.03.807)
- device type parameter in VATLSTxx member 157
- device usage by ASM 158.9 (VS2.03.807)
- Direct Access Device Activity report 158.9 (VS2.03.807)
- dispatchable job mix 166 (VS2.03.807)
- dispatching priorities
  - assigning of 202.16 (VS2.03.807)
  - automatic priority group (APG) 164, 165 (VS2.03.807)
  - evaluation and adjustment 202.8, 202.10 (VS2.03.807)
  - guidelines for selecting 202 (VS2.03.807)
  - performance characteristics 183 (VS2.03.807)
  - queue, use of 164 (VS2.03.807)
  - SRM control via 162, 164 (VS2.03.807)
- DMN parameter 202.19 (VS2.03.807)
- domain
  - constraints (*see domain constraints*) (VS2.03.807)
  - definition 170 (VS2.03.807)
  - delay 202.9 (VS2.03.807)
  - examples 171 (VS2.03.807)
  - guidelines for defining 200 (VS2.03.807)
  - MPL control over 170 (VS2.03.807)
  - purpose of 160, 161 (VS2.03.807)
  - special purpose 202 (VS2.03.807)
  - specification 202.18 (VS2.03.807)
  - SRM control via 161 (VS2.03.807)
  - switching of 186 (VS2.03.807)
  - weight (*see domain weight*) (VS2.03.807)
- domain constraints
  - definition 170 (VS2.03.807)
  - evaluation and adjustment 202.10, 202.14, 202.15 (VS2.03.807)
  - guidelines for selecting 200 (VS2.03.807)
  - multiprogramming level 170 (VS2.03.807)
  - specification 202.19 (VS2.03.807)
  - weight 172 (VS2.03.807)
- domain weight
  - definition 172 (VS2.03.807)
  - evaluation and adjustment 202.10 (VS2.03.807)
  - guidelines for selecting 200 (VS2.03.807)
  - specification 202.19 (VS2.03.807)
  - use of 201, 202 (VS2.03.807)
- DONTSWAP sysevent 285
- DPRTY parameter 202.16 (VS2.03.807)
- DSV parameter in SMFPRMxx 149
- dump
  - ABEND dump parameter
    - IEAABD00 48
    - IEADMP00 58
  - SVC dump 100-102
- DUMP parameter in IEASYSxx 100
- duplex page data set specification 102.1 (VS2.03.807)
- DUPLEX parameter
  - description 102.1 (VS2.03.807)
  - overview 87 (VS2.03.807)
  - sysgen macro – initialization parameter equivalence 85 (VS2.03.807)

## DUPLEXDS, NAME

keyword combination 22 (VS2.03.807)  
 parameter 85 (VS2.03.807)  
 duplexing, CLPA IPL 158.2 (VS2.03.807)  
 DUR, performance period duration 202.20 (VS2.03.807)  
 duration of performance periods  
   classes and subclasses 202.1 (VS2.03.807)  
   in real time units 202.9 (VS2.03.807)  
   selecting values 202.1 (VS2.03.807)  
   (*see also* performance period) (VS2.03.807)  
 dynamic allocation, invoking from a batch job 254  
 dynamic allocation of paging space 158.12 (VS2.03.807)

elapsed time of a transaction 174, 175 (VS2.03.807)  
 ENQ exchange swap count 202.14 (VS2.03.807)  
 ENQHOLD sysevent 274  
 ENQRLSE sysevent 274  
 enqueue bottleneck 167 (VS2.03.807)  
 enqueue contention 202.14 (VS2.03.807)  
 enqueue delay minimization 166 (VS2.03.807)  
 enqueue processing, SRM 162 (VS2.03.807)  
 enqueue residence value (ERV)  
   definition 166 (VS2.03.807)  
   ENQ swap out count 202.14 (VS2.03.807)  
   specification 202.22 (VS2.03.807)  
 ERV (*see* enqueue residence value) (VS2.03.807)  
 example  
   AMASPZAP service aid 158.4 (VS2.03.807)  
   calculating reserve address space 158.4 (VS2.03.807)  
   common page data set sizing 158.6 (VS2.03.807)  
   duplex page data set sizing 158.7 (VS2.03.807)  
   HMASMP service aid 158.5 (VS2.03.807)  
   local page data set sizing 158.7 (VS2.03.807)  
   PAGE parameter syntax 117 (VS2.03.807)  
   PLPA page data set sizing 158.6 (VS2.03.807)  
   sizing swap data sets 158.8 (VS2.03.807)  
   space calculation 158.5 (VS2.03.807)  
   syntax of PAGNUM parameter 119 (VS2.03.807)  
 exchange on recommendation value swap  
   count 202.12 (VS2.03.807)  
 exchange swap  
   definition 163 (VS2.03.807)  
   factors contributing to decision 183 (VS2.03.807)  
   workload level recommendation 177 (VS2.03.807)  
 exhaustion of paging space 158.11 (VS2.03.807)  
 EXCP appendage, how to specify 53  
 EXCP count maintained by SRM 198 (VS2.03.807)  
 EXCP relationship to I/O service unit 198 (VS2.03.807)  
 execution characteristics  
   discussion 161 (VS2.03.807)  
   batch vs. TSO 200 (VS2.03.807)  
 EXT parameter in SMFPRMxx 150

FIX parameter in IEASYSxx 103  
 fixed-head devices 158.1, 158.9 (VS2.03.807)  
 fixed impact of allocations 167 (VS2.03.807)  
 fixed link pack area list 61  
 fixed priorities 165 (VS2.03.807)  
 fixed requirements  
   discussion 190 (VS2.03.807)  
   not being met 202.8 (VS2.03.807)

formatting a data set for VSAM 102.2 (VS2.03.807)  
 formula for estimating paging data set sizes 158.10 (VS2.03.807)  
 FREE=CLOSE, use of 254

GMT (Greenwich mean time) 144

## GTF

member of parmlib 43  
 parameters 45  
 starting GTF (general) 39  
 starting via the IBM-supplied procedure 43, 45  
 used to trace sysevents 263

## GTFPARM

detailed description 43  
 synopsis 27

guaranteed access to real storage 201 (VS2.03.807)

IARDCPY parameter in IEASYSxx 104

relationship with HARDCOPY parameter of SCHEDULR  
 macro at sysgen (*see* "Default Value" 105)

IMASMP service aid 158.5 (VS2.03.807)

low page data sets are specified 116 (VS2.03.807)

IEAABD00 (*see also* IEADMPO0)

detailed description 47  
 synopsis 27

IEAAPFxx

detailed description 52  
 synopsis 28

IEAAPPO0

detailed description 53  
 synopsis 28

IEABLDxx

default member, IBM-supplied 57  
 detailed description 56

synopsis 28

IEADMPO0 (*see also* IEAABD00)

detailed description 58  
 synopsis 29

IEAFIXxx

detailed description 61  
 synopsis 29

IEAIPSxx (*see* IPS) (VS2.03.807)

IEALIMIT value 158.7 (VS2.03.807)

IEALOD00 (*see also* pageable link pack area)

detailed description 72  
 synopsis 30

IEALPAXx (*see also* MLPA and link pack area, pageable)

detailed description 73  
 synopsis 30

IEAOPTxx

evaluation and adjustment 202.8 (VS2.03.807)  
 guidelines for preparing 202.3 (VS2.03.807)  
 parameter concepts 187 (VS2.03.807)  
 parameter specifications 202.22 (VS2.03.807)  
 relation to SRM 160 (VS2.03.807)

IEAPAKBA (batch default pack list), description of 81

IEAPAK00 (*see also* link pack area, pageable)

default lists, IBM-supplied 79  
 detailed description 78

ISAM pack list, suggested 241  
 synopsis 31

- IEAPAKTS (TSO and batch default pack list), description of 80
- IEASYSxx  
 containing primary PLPA copy 158.8 (VS2.03.807)  
 source for PAGE parameter 116 (VS2.03.807)  
 source for PAGNUM parameter 119 (VS2.03.807)  
 SWAP parameter source 125.1 (VS2.30.807)
- IEASYSxx (*see also* individual parameters)  
 default contents 91  
 detailed description 85  
 incompatibilities with MVT and VS2 Release 1 parameters 88  
 MVT parameters, unsupported 88  
 overview of parameters 86  
 related sysgen macros 81  
 synopsis 32  
 syntax rules 91  
 VS2 Release 1 parameters, unsupported 90-91
- IEASYS00  
 containing primary PLPA copy 158.8 (VS2.03.807)  
 default contents 91  
 how overridden by IEASYSxx and operator 85  
 source of PAGE parameter 116 (VS2.03.807)  
 source of PAGNUM parameter 119 (VS2.03.807)  
 source of SWAP parameter 125.1 (VS2.03.807)  
 sysgen parameters copied to IEASYS00 85
- IEA890I error message 158.4 (VS2.03.807)
- IEBUPDTE, example of statements 35
- IKJPRM00  
 detailed description 131  
 synopsis 32
- ILRSLOT  
 constant 158.11 (VS2.03.807)  
 value  
 changing 158.4 (VS2.03.807)  
 description 158.3-158.4 (VS2.03.807)  
 example 158.4-158.5 (VS2.03.807)
- ILRSLOTV  
 constant 158.11 (VS2.03.807)  
 value  
 changing 158.4 (VS2.03.807)  
 description 158.4 (VS2.03.807)  
 example 158.4-158.5 (VS2.03.807)
- ILR009I message 125.2 (VS2.03.807)
- importance  
 of a domain 172 (VS2.03.807)  
 of an address space 176, 177 (VS2.03.807)  
 of types of work 161 (VS2.03.807)
- IMS dispatching priorities 202 (VS2.03.807)
- INITATT sysevent 270
- INITDET sysevent 270
- initial IPS and OPT, preparation of 191 (VS2.03.807)
- initialization, ASM 158.1-158.12 (VS2.03.807)
- initialization, system overview 20
- initialization, use of parmlib 24
- initiator classes and period durations 202.1 (VS2.03.807)
- initiator dispatching priority 202 (VS2.03.807)
- initiators, use of 303
- INLOCKHI (TSO parameter) 133
- INLOCKLO (TSO parameter) 133
- input terminal wait swap count 202.12 (VS2.03.807)
- installation performance specification (IPS)  
 (*see also* IPS) (VS2.03.807)  
 default 202.4 (VS2.03.807)  
 evaluation and adjustment 202.8 (VS2.03.807)  
 examples 202.6 (VS2.03.807)  
 guidelines for preparing 191 (VS2.03.807)  
 parameter concepts 169-186 (VS2.03.807)  
 parameter specifications 202.17-202.21 (VS2.03.807)  
 relation to SRM 160 (VS2.03.807)
- installation requirements, guidelines 189 (VS2.03.807)
- interval service value (ISV)  
 definition 181 (VS2.03.807)  
 competition for resources controlled by 181-183 (VS2.03.807)  
 evaluation and adjustment 202.13 (VS2.03.807)  
 guidelines for selecting 202.1 (VS2.03.807)  
 preemption during ISV 182, 186, 202.13 (VS2.03.807)  
 specification 202.20 (VS2.03.807)
- I/O activity, comparison between problem program and the system 262
- I/O device activity report (MF/1)  
 description of 215  
 how to use it 226
- I/O errors  
 paging 158.11 (VS2.03.807)  
 permanent 158.7 (VS2.03.807)
- I/O load adjusting  
 device allocation 167 (VS2.03.807)  
 factor in swap decisions 183 (VS2.03.807)
- I/O load balancing, by ASM 158.11 (VS2.03.807)
- I/O service, selecting coefficient value 198 (VS2.03.807)
- I/O service units 174 (VS2.03.807)
- I/O supervisor queues, ASM 158.10 (VS2.03.807)
- IOC parameter  
 resource factor coefficient 202.22 (VS2.03.807)  
 service definition coefficient 202.20 (VS2.03.807)
- IPL  
 adding page space during life 118 (VS2.03.807)  
 CLPA 102.1 (VS2.03.807)  
 non-CLPA 102.1-102.2 (VS2.03.807)  
 page data set number requirement 115 (VS2.03.807)  
 SWAP parameter specification 125.1 (VS2.03.807)  
 termination 117 (VS2.03.807)
- IPL, types of 22
- IPS  
 (*see also* installation performance specification) (VS2.03.807)  
 changing the IPS between IPLs via SET IPS 38  
 member description 63  
 parameter in IEASYSxx 106  
 parameter on SET command 202.16 (VS2.03.807)
- ISV parameter 202.20 (VS2.03.807)  
 (*see also* interval service value) (VS2.03.807)
- job classing, adjustment of 202.11 (VS2.03.807)
- JOB CAT DD statement, use of to speed catalog search 258
- JOBSELECT sysevent 269
- JOBTERM sysevent 269
- job wait limit (JWL parameter in SMFPRMxx) 147
- JWT (parameter in SMFPRMxx) 147
- L keyword 115 (VS2.03.807)



- libraries that require authorization, how to specify 52
- LINKLIB
  - concatenating libraries with 139
- link pack area
  - default pack lists 79
  - extension list (IEALPAxx) 73
  - fixed, use of 242
  - loading of 243-245
  - pack list (IEAPAK00) 78
  - pageable: advantages, uses, recommendations 240
- list capability for IPL and parmlib parameters
  - description of operator responses needed to list IPL parameters (*see Operator's Library: OS/VS2 Reference (JES2)*)
  - which parmlib members and IEASYSxx parameters can be listed 27
- LNK (parameter in IEASYSxx) 107
- LNKLSTxx
  - detailed description 139
  - synopsis 33
- load adjusting swap recommendations, use of 202.15 (VS2.03.807)
- load adjusting values
  - cut-off workload level 187 (VS2.03.807)
  - effect on swap analysis 187, 188 (VS2.03.807)
- load list (member IEALOD00)
  - detailed description 72
  - synopsis 30
- log (*see* LOGCLS and LOGLMT)
- LOGCLS (parameter in IEASYSxx) 108
- logical channel utilization 166, 167 (VS2.03.807)
- logical groups handled by Auxiliary Storage Manager, limit on the number of 110
- LOGLMT parameter in IEASYSxx) 109
- LOGONs 118 (VS2.03.807)
- long wait swap count 202.12 (VS2.03.807)
- LPA pages 115 (VS2.03.807)
- LSQA
  - for swap sets 158.5 (VS2.03.807)
  - pages for swap data set 125.2 (VS2.03.807)
  - space for swap data sets 125.1 (VS2.03.807)
- main storage service, selecting coefficient
  - value 198 (VS2.03.807)
- MAN (parameter in SMFPRMxx) 147
- Mass Storage System
  - MVIKEY00 (in SYS1.PARMLIB) 141-143
  - PURGE parameter 120
- master catalog 116 (VS2.03.807)
- MAXUSER (parameter in IEASYSxx) 110
- mean-time-to-wait priority 165, 202 (VS2.03.807)
- measurement facility (*see* MF/1)
- measurement facilities 158.10 (VS2.03.807)
- measurement tools
  - MF/1 203
    - SMF used to supplement MF/1 261
    - sysevent trace (GTF function) 263
- MEMCREAT sysevent 268
- MEMDEL sysevent 268
- MF/1
  - how to use it (introduction) 203
  - messages 135-136
  - operator start command, example of 205
  - parameters
    - conflicts between 207
    - multiple sources of 205
  - procedure, IBM-supplied 204
  - reports and SMF records
    - channel activity 212, 225
    - CPU activity 210, 224
    - device activity 215, 226
    - how to use the reports 224
    - paging activity 216, 227
    - types of reports and SMF records 209
    - workload activity 220-228
  - used to determine ASM initialization requirements 158.7-158.8, 158.10 (VS2.03.807)
  - minimum paging space 117 (VS2.03.807)
- MLPA
  - duplexed for recoverability 98 (VS2.03.807)
  - exclusion from common areas 90, 158.1 (VS2.03.807)
  - paging data set size estimation 158.10 (VS2.03.807)
  - parameter in IEASYSxx 111
- modified link pack area list (IEALPAxx) 73
- mount attribute (in VATLSTxx member)
  - definition 158
  - parameter position, meaning, and value range 157
- MOUNT commands 118 (VS2.03.807)
- mount message suppression (parameter in VATLSTxx member) 158
- movable-head devices 158.8-158.9 (VS2.03.807)
- MPL (*see* multiprogramming level) (VS2.03.807)
- MSO, service definition coefficient 202.20 (VS2.03.807)
- MSS (*see* Mass Storage System)
- multiprogramming level (MPL)
  - adjusting function 165 (VS2.03.807)
  - definition 170 (VS2.03.807)
  - evaluation and adjustment 202.10, 202.11 (VS2.03.807)
  - guidelines for selecting 200 (VS2.03.807)
  - target MPL 170 (VS2.03.807)
  - use of 170-172 (VS2.03.807)
- MVIKEY00
  - detailed description 141-143
  - synopsis 33
- NEWIPS sysevent 280
- NIOWAIT sysevent 267
- NIP, writing PLPA to PLPA page data set 116 (VS2.03.807)
- nucleus, map of (*see* NUCMAP)
- NUCMAP (parameter in IEASYSxx) 112
- OBJ, performance objective specification 202.20 (VS2.03.807)
- OLTEP, real region size needed 121
- operations, page/swap 158.1 (VS2.03.807)
- operator commands related to the SRM 202.16 (VS2.03.807)
- operator commands (system) used to tailor the system 37
- operator entry of parameters 23
- operator intervention (*see* OPI, and operator entry of parameters)

- operator responses to SPECIFY SYSTEM PARAMETERS message (*see Operator's Library: OS/VS2 Reference (JES2)*)
- OPI
  - parameter in IEASYSxx 113
  - parameter in SMFPRMxx 150
- OPT
  - (*see also IEAOPTxx*) (VS2.03.807)
  - parameter in IEASYSxx 114
  - parameter in SMFPRMxx 146, 166
- optimizing use of control mechanisms 202.12 (VS2.03.807)
- order for specifying page data sets 115 (VS2.03.807)
- output terminal wait swap count 202.12 (VS2.03.807)
- overriding
  - page data sets 116 (VS2.03.807)
  - swap data sets 125.1 (VS2.03.807)
- overspecifying space 158.9 (VS2.03.807)
- OWAITHI (TSO parameter) 133
- OWAITLO (TSO parameter) 134
  
- pack list (IEAPAK00)
  - default pack lists 79
  - description and use 78
  - ISAM pack list, suggested 241
- page data set values 158.5 (VS2.03.807)
  - definition 40 (VS2.03.807)
  - page-out selection 158.12 (VS2.03.807)
  - performance aspects 158.3 (VS2.03.807)
  - specification order 115 (VS2.03.807)
  - specification sources 116 (VS2.03.807)
- PAGE parameter
  - contrasted with SWAP parameter
    - specification 125.1 (VS2.03.807)
  - description 115-119 (VS2.03.807)
  - for primary PLPA copy 158.8 (VS2.03.807)
  - sysgen macro equivalence 85 (VS2.03.807)
- page seconds 199 (VS2.03.807)
- pageable frame shortage 168 (VS2.03.807)
- pageable frame stealing 169 (VS2.03.807)
- pageable link pack area (*see also* IEAPAK00, IEALOD00, and IEALPAXX)
  - advantages, uses, and recommendations 240
  - ISAM pack list, recommended 241
  - loading of 243
  - module search sequence 243
- PAGEADD command
  - overview 88 (VS2.03.807)
  - page space shortage 118 (VS2.03.807)
  - swap data set 125.2 (VS2.03.807)
- PAGEDSN keyword
  - DATASET macro instruction 115, 116 (VS2.03.807)
  - for primary PLPA copy 158.8 (VS2.03.807)
- PAGEDSN parameter 85 (VS2.03.807)
- page-out selection, page data set 158.12 (VS2.03.807)
- page space shortage 118 (VS2.03.807)
- page/swap operations 158.1 (VS2.03.807)
- paging activity report (MF/1)
  - description of 216
  - how to use it 227
- Paging Activity Report 158.10 (VS2.03.807)
- paging data sets, size estimating 158.10 (VS2.03.807)
- paging devices 158.9 (VS2.03.807)
- paging I/O errors 158.11 (VS2.03.807)
- paging overhead, excessive 202.14 (VS2.03.807)
- paging rate
  - address space vs. Pageable System Area 202.10 (VS2.03.807)
  - monitoring by SRM 165 (VS2.03.807)
- paging rates, comparison between problem program and system 261
- paging space
  - adding 158.11 (VS2.03.807)
  - dynamic allocation 158.12 (VS2.03.807)
  - exhaustion 158.11 (VS2.03.807)
  - minimum 117 (VS2.03.807)
  - TSO requirements 158.7 (VS2.03.807)
- paging subsystem, monitoring by SRM 165 (VS2.03.807)
- PAGNUM parameter
  - default value 119 (VS2.03.807)
  - overview 88 (VS2.03.807)
  - page space shortage 118 (VS2.03.807)
  - syntax example 119 (VS2.03.807)
  - sysgen macro equivalence 85 (VS2.03.807)
  - value range 119 (VS2.03.807)
- parameters
  - implicit 39-40
  - operator entry of 23
  - parmlib (*see individual member names*)
- parmlib
  - characteristics of each member, tabulated 27
  - how members are created 25
  - how to control it 35
  - member descriptions, detailed 41
  - relationships to IPL parameters and sysgen parameters 25
  - syntax rules, general 36
- PARMTZ
  - detailed description 144
  - synopsis 33
- path, physical channel 166 (VS2.03.807)
- performance characteristics
  - control over by SRM 162 (VS2.03.807)
  - dispatching priority 162, 183 (VS2.03.807)
  - performance period 184 (VS2.03.807)
  - slope of objective 178, 180 (VS2.03.807)
- performance degradation
  - duplexing 118 (VS2.03.807)
  - paging space spillage 117 (VS2.03.807)
- performance factors
  - catalog 257
  - device allocation 252
  - miscellaneous 303
  - pageable link pack area 240
  - TCAM 294
  - VIO 246
- performance group
  - definition 185 (VS2.03.807)
  - guidelines for defining 202.4 (VS2.03.807)
  - specification 202.18 (VS2.03.807)
- performance group number (PGN)
  - assigning of 202.15 (VS2.03.807)
  - default 202.16 (VS2.03.807)
  - transaction definition 173 (VS2.03.807)
- performance group period (VS2.03.807)
  - (*see performance period*) (VS2.03.807)

- performance objective
  - competition for resources controlled
    - by 178-181 (VS2.03.807)
  - cut-off workload level 202.2, 202.3, 177 (VS2.03.807)
  - definition 176 (VS2.03.807)
  - selecting service rates 202.2 (VS2.03.807)
  - specification 202.17 (VS2.03.807)
  - use of slope 202.2, 177 (VS2.03.807)
- performance period
  - definition 184 (VS2.03.807)
  - selecting durations 202.1 (VS2.03.807)
- performance recommendations,
  - ASM 158.8-158.10 (VS2.03.807)
- PERFORM parameter, performance group 202.15 (VS2.03.807)
- period, performance group (VS2.03.807)
  - (see performance period) (VS2.03.807)
- permanent I/O errors 158.7 (VS2.03.807)
- PGN parameter 202.20 (VS2.03.807)
  - (see also performance group number) (VS2.03.807)
- placement of common areas 90 (VS2.03.807)
- PLPA
  - (see also IEAPAK00, IEALOD00, and IEALPAXX)
  - advantages, uses and guidelines 240
  - classification 158.1 (VS2.03.807)
  - duplexed for recoverability 102.1 (VS2.03.807)
  - loading 243
  - location 90 (VS2.03.807)
  - minimum paging space 117 (VS2.03.807)
  - module search sequence 243
  - page data set 115 (VS2.03.807)
  - paging data set size estimation 158.10 (VS2.03.807)
- preemption 180, 182 (VS2.03.807)
  - (see also interval service value) (VS2.03.807)
- prevention, auxiliary storage shortage 158.11 (VS2.03.807)
- prevention of storage shortages, SRM 167 (VS2.03.807)
- PURGE (parameter in IEASYSxx) 120
- purging VIO data sets 22 (VS2.03.807)
  
- QSECCMP sysevent 271
- QSCEFL sysevent 273
- QSEST sysevent 271
- questions and answers, ASM 158.11-158.12 (VS2.03.807)
- quick starts
  - DUPLEX parameter 87 (VS2.03.807)
  - prerequisite conditions for 116 (VS2.03.807)
  - TPARTBLE 116 (VS2.03.807)
  
- readback check 158.11 (VS2.03.807)
- ready jobs, average number of 189 (VS2.03.807)
- REAL (parameter in IEASYSxx; see also VRREGN) 121
- real storage
  - default region size for an ADDRSPC=REAL job 128
- real storage, overcommitment of 202.14 (VS2.03.807)
- real time duration for performance period 202.9 (VS2.03.807)
- REC (parameter in SMFPRMxx) 149
- reclaim rate, definition of 216
- reconfigurable storage units 122
- RECONLIM (TSO parameter) 134
  
- reduction of data
  - SGP program to reduce SMF data 261
  - sysevent trace reduction
    - rationale 264-265
    - samples 289-293
- REGION parameter 158.7 (VS2.03.807)
- repeatability of service
  - storage component 199 (VS2.03.807)
  - CPU component 198 (VS2.03.807)
- replacing swap data sets 22 (VS2.03.807)
- reports, MF/1 (see also names of individual reports)
  - descriptions of 209
  - how to use them 224
- REQGDAT sysevent 284
- REQSERVC sysevent 283
- REQSWAP sysevent 286
- request swap 164 (VS2.03.807)
- requirements, response and throughput 189, 190 (VS2.03.807)
- RESET command 202.16 (VS2.03.807)
- RESETPG sysevent 279
- resident time of a transaction 174 (VS2.03.807)
- resource factor coefficients
  - definition 187 (VS2.03.807)
  - guidelines for selecting 202.3 (VS2.03.807)
  - specification 202.22 (VS2.03.807)
- resource manager constant 188 (VS2.03.807)
- resource shortages, prevention of 167 (VS2.03.807)
- resource use functions 165 (VS2.03.807)
- response-throughput bias (RTB)
  - bottleneck elimination 202.15 (VS2.03.807)
  - definition 183 (VS2.03.807)
  - guidelines for use 202.3 (VS2.03.807)
  - specification 202.20 (VS2.03.807)
- response time
  - (see also TSO response time) (VS2.03.807)
  - ISV 182 (VS2.03.807)
  - MPL and domain weight 201 (VS2.03.807)
  - performance objective 180 (VS2.03.807)
  - performance period 186 (VS2.03.807)
- RESVBUF (TSO parameter) 134
- RFC parameter 202.22 (VS2.03.807)
- RMC parameter 202.22 (VS2.03.807)
- RMF 158.7-158.8, 158.10 (VS2.03.807)
- RMF reports 158.3 (VS2.03.807)
- Rotate dispatching priority 165, 202 (VS2.03.807)
- rotational position, slot 158.12 (VS2.03.807)
- RSMCNSTS sysevent 275
- RSTORCMP sysevent 273
- RSU (parameter in IEASYSxx) 122
- RTB parameter 202.20 (VS2.03.807)
  - (see also response-throughput bias) (VS2.03.807)
  
- selecting ASM slots 158.12 (VS2.03.807)
- selection of candidates for permanent data
  - sets 167 (VS2.03.807)
- service, comparison between problem program and system 262
- service
  - computation 174 (VS2.03.807)
  - definition 173 (VS2.03.807)
  - establishing requirements 189 (VS2.03.807)
  - maximum available, example 202.2 (VS2.03.807)

## VS2.03.807

- service bursting 158.2 (VS2.03.807)
- service burst length 158.8 (VS2.03.807)
- service definition coefficients
  - defaults 197 (VS2.03.807)
  - definition 174 (VS2.03.807)
  - evaluation and adjustment 202.15 (VS2.03.807)
  - guidelines for selecting 197 (VS2.03.807)
  - specification 202.17 (VS2.03.807)
- service rate
  - calculation 175, 202.2 (VS2.03.807)
  - definition 174 (VS2.03.807)
  - part of performance objective 176 (VS2.03.807)
  - specified vs. measured 177 (VS2.03.807)
- service units 174 (VS2.03.807)
- SET command 38, 202.16 (VS2.03.807)
- SETDMN command 202.16 (VS2.03.807)
- shortage
  - auxiliary storage, prevention 158.11, 167 (VS2.03.807)
  - page space 118 (VS2.03.807)
  - pageable frame, prevention 168 (VS2.03.807)
  - SQA, prevention 168 (VS2.03.807)
- slope of a performance objective 177, 202.2 (VS2.03.807)
- slot
  - rotational position 158.12 (VS2.03.807)
  - selection, ASM 158.12 (VS2.03.807)
  - usage snapshot 158.10 (VS2.03.807)
- slots, shortage of 168 (VS2.03.807)
- SMF
  - as a measurement tool to supplement MF/1 261
  - data reduction program (SGP) 261
  - parameter in IEASY500 123
  - parameters in SMFPRMxx parmlib member 146
- SMFPRMxx
  - detailed description 146
  - overview 34
- snapshot of slot usage 158.10 (VS2.03.807)
- sources of page data set specifications 116 (VS2.03.807)
- space overspecification 158.9 (VS2.03.807)
- special purpose domains 202 (VS2.03.807)
- specification order for page data sets 115 (VS2.03.807)
- specific unit address, avoidance guideline 253
- SQA duplex page data set sizing
  - consideration 158.7 (VS2.03.807)
- SQA
  - parameter in IEASYSxx 124
  - space shortage 124, 168
- SQALOW sysevent 276
- SQAOK sysevent 277
- SRM (see System Resources Manager)
- SRM page space shortage warnings 118 (VS2.03.807)
- SRM trace record, format and use 263
- SRV parameter 202.20 (VS2.03.807)
- start initiator commands 118 (VS2.03.807)
- stealing, pageable frame 169 (VS2.03.807)
- STEPCAT DD statement, use of to speed catalog search 258
- storage service units 174, 198 (VS2.03.807)
- subclass of work, period duration 202.1 (VS2.03.807)
- subsystem
  - defining requirements for 189 (VS2.03.807)
  - domains for 200 (VS2.03.807)
  - nonswappable 200 (VS2.03.807)
- support, ASM for page data sets 118 (VS2.03.807)
- SVC DUMP data set 100-102
- swap analysis 187 (VS2.03.807)
- swap counts, problem analysis 202.12 (VS2.03.807)
- swap data set
  - replacement 22 (VS2.03.807)
  - size performance aspect 158.3 (VS2.03.807)
  - sizing example 158.8 (VS2.03.807)
  - values 158.6 (VS2.03.807)
- SWAP DATA SET BAD message 125.2 (VS2.03.807)
- SWAPDSN parameter 85 (VS2.03.807)
- SWAP parameter
  - description 125.1-125.2 (VS2.03.807)
  - how specified 125.1 (VS2.03.807)
  - minimum swap space 125.2 (VS2.03.807)
  - overview 88 (VS2.03.807)
  - swap space shortage 125.2 (VS2.03.807)
  - syntax description 125.1 (VS2.03.807)
  - syntax examples 125.2 (VS2.03.807)
  - sysgen macro equivalence 85 (VS2.03.807)
- swap recommendation value
  - calculation 187 (VS2.03.807)
  - exchange swap 177 (VS2.03.807)
  - workload level 177 (VS2.03.807)
- swapping
  - analysis 187 (VS2.03.807)
  - control of (see swapping frequency) (VS2.03.807)
  - to control competition for resources 178-181 (VS2.03.807)
  - types 163, 164 (VS2.03.807)
- swapping frequency
  - cut-off level 202.1 (VS2.03.807)
  - ISV 181, 188 (VS2.03.807)
  - performance objective slope 181 (VS2.03.807)
  - resource factor coefficients 188 (VS2.03.807)
- swapping overhead, excessive 202.12 (VS2.03.807)
- SWINFL sysevent 272
- SWOUTCMP sysevent 272
- syntax example for the DUPLEX parameter 102.2 (VS2.03.807)
- syntax rules for IPS and OPT parameters 202.17 (VS2.03.807)
- SYQSCCMP sysevent 282
- SYQSCST sysevent 281
- sysevents
  - meaning of the various sysevent codes 265-288
  - sample data reductions 289
  - tracing and data reduction 263
- sysgen
  - CTRLPROG macro instruction 119 (VS2.03.807)
  - DATASET macro instruction 125.1 (VS2.03.807)
  - parameters that are copies to IEASY500 85
  - relationships between sysgen parameters and parmlib members 25
  - specifying page data sets 116 (VS2.03.807)
  - SWAP parameter 125.1 (VS2.03.807)
- SYSLOG (operand of HARDCPY parameter in IEASYSxx) 104
- SYSP (parameter specified at IPL by the operator) 126
- System Activity Measurement Facility (see also MF/1)
  - how to use it 203
  - parameters, description of 135
- system contention factors 166 (VS2.03.807)
- system initialization (see initialization)
- system master catalog 102.2 (VS2.03.807)

- system parameter list (IEASYSxx)
  - detailed description 85
  - overview of parameters 86
  - synopsis 32
- system performance factors 231
- system resources, access to via domains 161 (VS2.03.807)
- system resources manager (SRM)
  - constants 202.23 (VS2.03.807)
  - default IPS 202.4 (VS2.03.807)
  - description 159 (VS2.03.807)
  - functions 163 (VS2.03.807)
  - guidelines for parameter selection 188 (VS2.03.807)
  - installation management controls 202.15 (VS2.03.807)
  - objectives 160 (VS2.03.807)
  - parameter concepts 169 (VS2.03.807)
  - parameter evaluation and adjustment 202.8 (VS2.03.807)
  - parameter values and syntax 202.17 (VS2.03.807)
  - types of control 161 (VS2.03.807)
- system throughput via mean-time-to-wait
  - priority 165 (VS2.03.807)
- system tuning and the SRM 159 (VS2.03.807)
- SYSP operator command
  - PAGE parameter 115 (VS2.03.807)
  - PAGNUM parameter 119 (VS2.03.807)
- SYS1.DSSVM data set 152 (VS2.03.807)
- SYS1.DUMPnn data set 100-102
- SYS1.DUMPxx data set 152 (VS2.03.807)
- SYS1.PARMLIB (*see also* names of individual members)
  - description of members 41
  - synopsis 27
  - use of (overview) 24
- SYS1.STGINDEX data set
  - concatenation of 152 (VS2.03.807)
  - in master catalog 260 (VS2.03.807)
- SYS1.VTAMLST 38
- target MPL 170 (VS2.03.807)
- task execution time 198 (VS2.03.807)
- TCAM
  - changing parameters at START command 38
  - tuning considerations 294
- telemetry paging data sets 158.9 (VS2.03.807)
- temporary page activity reference table 116 (VS2.03.807)
- Terminal I/O Coordinator parameters (*see* TIOC parameters)
- TERMWAIT sysevent 266
- TGETTPUT sysevent 281
- throughput control, SRM 162 (VS2.03.807)
- time interval for service rate 174, 175 (VS2.03.807)
- time sharing
  - how to start under MVS 38
  - maximum number of logged-on users (USERMAX parameter) 134
  - parameters (IKJPRM00) 131
- time sharing paging data sets 158.9 (VS2.03.807)
- time zone constant 144
- TIMEREXP sysevent 266
- TIOC parameters 131
- TOD (keyword in COMMNDxx)
  - syntax and meaning 42
  - use of 144
- TPARTBLE 116 (VS2.03.807)
- track overflow
  - duplex page data set 102.2 (VS2.03.807)
  - page data set 116 (VS2.03.807)
  - swap data set 125.1 (VS2.03.807)
- transaction
  - definition 173 (VS2.03.807)
  - duration of batch and TSO 202.1 (VS2.03.807)
  - life span 184 (VS2.03.807)
  - performance group 185 (VS2.03.807)
  - performance period 184 (VS2.03.807)
  - service used by 175 (VS2.03.807)
  - time intervals defined for 175 (VS2.03.807)
- TSEVENT00 sysevent 265
- TSO
  - allocation suggestions 255
  - IPS service modifications 65
  - parameters 131
- TSO, defining requirements for 189 (VS2.03.807)
- TSO domains, defining of 200 (VS2.03.807)
- TSO paging space requirements 158.7 (VS2.03.807)
- TSO response time
  - (*see also* response time) (VS2.03.807)
  - erratic 202.8 (VS2.03.807)
  - impacted by paging overhead 202.10 (VS2.03.807)
  - too good 202.9, 202.11 (VS2.03.807)
  - poor 202.10 (VS2.03.807)
- TSO transactions, dispatching priorities 202 (VS2.03.807)
- turnaround time
  - due to performance objective 180 (VS2.03.807)
  - not good enough 202.11 (VS2.03.807)
- types of work, grouped into domains 161 (VS2.03.807)
- UIC (*see* unreferenced interval count) (VS2.03.807)
- under initiation, definition of 161
- unilateral swap count 202.12 (VS2.03.807)
- unilateral swap in 163 (VS2.03.807)
- unilateral swap out 163, 202.12 (VS2.03.807)
- UNITNAME sysgen macro, use of
  - defining separate esoteric subgroups 253
  - specifying VIO data sets 249
- UNIT parameter 102.2 (VS2.03.807)
- UNIT specification
  - DUPLEX parameter 102.2 (VS2.03.807)
  - PAGE parameter 117 (VS2.03.807)
  - SWAP parameter 125.2 (VS2.03.807)
- unreferenced interval count (UIC)
  - definition 169 (VS2.03.807)
  - analysis of overhead 202.14 (VS2.03.807)
- UNT, unit specification for the DUR
  - parameter 202.21 (VS2.03.807)
- use attribute in VATLSTxx member
  - definition 152
  - position of parameter, meaning, and value range 156
  - use characteristics, for domain creation 161 (VS2.03.807)
- USERMAX TSO parameter 134
- USERRDY sysevent 267
- utilization of resources
  - adjusting functions 166 (VS2.03.807)
  - evaluation 202.14, 202.15 (VS2.03.807)
  - monitoring by SRM 165 (VS2.03.807)

- V=R default region size 128
- VAL parameter in IEASYSxx 127
- VATLSTxx
  - detailed description 152-153
  - synopsis 136
  - used for non-demountable devices 151
- value, error buffer 158.6
- values
  - page data set 158.5 (VS2.03.807)
  - page number, obtaining 119
  - swap data set 158.6 (VS2.03.807)
- VERIFYPG sysevent 279
- VIO
  - advantages 246
  - disadvantages 248
  - how to specify VIO data sets 249
  - performance considerations 248
  - questions and answers 250
  - window, definition and size 249
- VIO data set
  - paging data set size estimation 158.10 (VS2.03.807)
  - purging 22 (VS2.03.807)
- VOLSER parameter 102.2 (VS2.03.807)
- volume attribute list (VATLSTxx member) 151-153  
(*see also* VATLSTxx member)
- volume serial parameter in VATLSTxx member 157
- VRREGN parameter in IEASYSxx 128  
(*see also* REAL)
- VSAM
  - format 102.2, 116 (VS2.03.807)
  - swap data set format 125.1 (VS2.03.807)
- VSAM catalog
  - performance factor guidelines 257
  - questions and answers 260
- VTAM
  - default pack lists 79-80
  - SYS1.VTAMLST 37
- warm starts
  - DUPLEX parameter 87 (VS2.03.807)
  - prerequisite conditions for 116 (VS2.03.807)
  - TPARTBLE 116 (VS2.03.807)
- warning messages, page space shortage 118 (VS2.03.807)
- weight
  - of domains (*see* domain weight) (VS2.03.807)
  - of load adjusting recommendations 187 (VS2.03.807)
  - of service components 174 (VS2.03.807)
- WKL, workload level specification 202.21 (VS2.03.807)
- WKLDLDCOLL sysevent 287
- WKLDINIT sysevent 287
- WKLDTERM sysevent 288
- workload activity report (MF/1)
  - description of 220 (VS2.03.807)
  - how to use it 228 (VS2.03.807)
- workload control, SRM 162 (VS2.03.807)
- workload levels
  - competition for resources 178-181 (VS2.03.807)
  - cut-off level 177 (VS2.03.807)
  - definition 176 (VS2.03.807)
  - relationship to performance objective 176 (VS2.03.807)
  - specification 202.17 (VS2.03.807)
  - swap recommendation value 177 (VS2.03.807)
- work sheets 191 (VS2.03.807)
- write-between-read logic 158.2 (VS2.03.807)
- WTL macro, maximum number allowed 109
- WTO buffers, number of 128
- WTOBFRS parameter in IEASYSxx 128
- WTORPLY 130
- OE1 completion code 158.4 (VS2.03.807)
- 03C wait state 158.4 (VS2.03.807)

*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 your name and address 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 System Programming Library: Initialization and Tuning (S370-34) Printed in U.S.A. GC28-0755-0



**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)**



*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 your name and address 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.

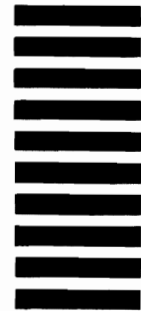
Fold

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 System Programming Library: Initialization and Tuning (S370-34)

Printed in U.S.A.

GC28-0755-0



**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)