OS/VS2 System Programming Library: Debugging Handbook

Volume 1

GC28-0708-1
File No. S370-37

Includes Selectable Units:

Scheduler Improvements	VS2.03.804
Supervisor Performance #1	VS2.03.805
Supervisor Performance #2	VS2.03.807
Data Management	VS2.03.808
IBM 3800 Printing Subsystem	VS2.03.810
TSO/VTAM	VS2.03.813
Scheduler/IOS Support	VS2.03.816
Service Data Improvements	VS2.03.817
MSS Enhancements	5752-824
3838 Vector Processing Subsystem	5752-829
3895 Device Support	5752-830
System Security Support	5752-832
Dumping Improvements	5752-833
Attached Processor Support	5752-847
MVS Processor Support	5752-851
Hardware Recovery Enhancements	5752-855
Interactive Problem Control System	5752-857
TSO/VTAM Level 2	5752-858
Data Management Support	5752-860

Includes Program Product:

MVS/System Extensions	5740-XE1

IBM Corporation, Publications Development, Dept. D58,
Bldg. 706-2, PO Box 390, Poughkeepsie, New York 12602

Second Edition (November, 1978)

This is a major revision of and obsoletes GC28-0708-0 and GC28-0751-0 incorporating changes released in the following Technical Newsletters and System Library Supplements:

Scheduler Improvements	VS2.03.804	SU4	(GN28-2678)
Supervisor Performance #1	VS2.03.805	SU5	(GN28-2680)
Supervisor Performance #2	VS2.03.807	SU7	(GN28-2704)
Data Management	VS2.03.808	SU8	(GN28-2679)
IBM 3800 Printing Subsystem	VS2.03.810	SU10	(GN28-2722)
TSO/VTAM	VS2.03.813	SU13	(GN28-2655)
Service Data Improvements	VS2.03.817	SU11	(GN28-2768)
3838 Vector Processing Subsystem	5752-829	SU29	(GC28-0925-0)
3895 Device Support	5752-830	SU30	(GC28-0800-0)
System Security Support	5752-832	SU32	(GC28-0845-0)
Dumping Improvements	5752-833	SU33	(GC28-0816-0)
MVS Processor Support	5752-851	SU51	(GD23-0025-0)
Hardware Recovery Enhancements	5752-855	SU55	(GC28-0891-0)
Interactive Problem Control System	5752-857	SU57	(GD23-0096-0)
TSO/VTAM Level 2	5752-858	SU58	(GD23-0048-0)
Data Management Support	5752-860	SU60	(GD23-0076-0)
MVS/System Extensions (program product)	5752-XE1	XE1	(SD23-0001-0)

This edition with Technical Newsletter GN28-2967 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*, GA20-0001, for the editions that are applicable and current.

The JES3 information contained in this manual is applicable only if JES3 has been integrated into your system.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's 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.

Technical Newsletter

This Newsletter No. GN28-2984
Date July 16, 1979
Base Publication No. GC28-0708-1
File No. S370-37
Prerequisite Newsletters/ Supplements None

OS/VS2 System Programming Library:
Debugging Handbook Volume 1

This newsletter contains replacement pages for *Debugging Handbook (Vol. 1)* in support of IBM 3203-5.

Before inserting any of the attached pages into *Debugging Handbook (Vol. 1)*, read ***carefully*** the instructions on this cover. They indicate when and how you should insert the pages.

Pages to be Removed	Attached Pages to be Inserted*
xi - xii	xi - xii
4-7 - 4-10	4-7 - 4-10
4-25 - 4-36	4-25 - 4-36

*If you are inserting pages from different Newsletters/Supplements and ***identical*** page numbers are involved, always use the page with the latest date (shown in the slug at the top of the page). The page with the latest date contains the most complete information.

A change to the text or to an illustration is indicated by a vertical line to the left of the change.

Summary of Amendments

This technical newsletter contains new and updated information in support of the IBM 3203-5 device.

Note: *Please file this cover letter at the back of the base publication to provide a record of changes*

IBM Corporation, Publications Development, Department D58, Building 706-2,
PO Box 390, Poughkeepsie, New York 12602

Printed in U.S.A.

Technical Newsletter

This Newsletter No. GN28-2967
Date March 12, 1979
Base Publication No. GC28-0708-1
File No. S370-37
Prerequisite Newsletters/ Supplements None

OS/VS2 System Programming Library:
Debugging Handbook Volume 1

This newsletter contains replacement pages for *Debugging Handbook (Vol. 1)*.

Before inserting any of the attached pages into *Debugging Handbook (Vol. 1)*, read ***carefully*** the instructions on this cover. They indicate when and how you should insert the pages.

Pages to be Removed	Attached Pages to be Inserted*
Cover - Edition Notice	Cover - Edition Notice
5-47 - 5-48	5-47 - 5-48

*If you are inserting pages from different Newsletters/Supplements and ***identical*** page numbers are involved, always use the page with the latest date (shown in the slug at the top of the page). The page with the latest date contains the most complete information.

A change to the text or to an illustration is indicated by a vertical line to the left of the change.

Summary of Amendments

This update modifies SVC 109.

Note: *Please file this cover letter at the back of the base publication to provide a record of changes.*

IBM Corporation, Publications Development, Department D58, Building 706-2, PO Box 390, Poughkeepsie, New York 12602

 Printed in U.S.A.

This handbook provides reference information for use in debugging user or system programs. The user of this publication should have a working knowledge of OS/VS2 functions and logic.

The handbook has been divided into three volumes totaling six sections:

Volume 1 (GC28-0708-1)

- **Section 1. Problem Categories and Analysis** describes an approach to debugging based on identification and analysis of system status indicators.
- **Section 2. Debugging Aids** summarizes major OS/VS2 debugging aids.
- **Section 3. Dump and Trace Formats** describes the output of debugging aids summarized in Section 2.
- **Section 4. Error Indicators** summarizes major system error indicators.
- **Section 5. General Reference** provides general reference information useful for debugging purposes.
- **Section 6. Control Block Chains** illustrates the logical relationships of major system data areas.

Volume 2 (GC28-0709-1)

- **Data Areas A-M** Describes the format of the data areas, and includes data areas frequently used in debugging.

Volume 3 (GC28-0710-0)

- **Data Areas N-Z** Describes the format of the data areas, and includes data areas frequently used in debugging.

The following publications are referenced within this book:

GA22-6966	IBM System/370 Model 155 Operating Procedures
GA22-6969	IBM System/370 Model 165 Operating Procedures
GC27-6974	OS/VS Problem Determination Aids and Messages and Codes for GPS and GSP
GC28-0627	OS/VS2 System Programming Library: Job Management
GC28-0628	OS/VS2 System Programming Library: Supervisor
GC28-0673	OS/VS System Modification Program (SMP)
GC28-0674	OS/VS2 System Programming Library: Services Aids
GC28-0675	OS/VS2 System Programming Library: OLTEP
GC28-0677	OS/VS2 System Programming Library: SYS1.LOGREC Error Recording
GC28-0681	OS/VS2 System Programming Library: Initialization and Tuning Guide
GC28-0683	OS/VS2 Supervisor Services and Macro Instructions
GC28-0692	OS/VS2 JCL
GC28-0725	OS/VS2 System Programming Library: Diagnostic Techniques
GC30-2045	OS/VS TCAM User's Guide
GC34-2006	OS/VS2 MVS Interactive Problem Control System (IPCS) User's Guide and Reference
GC38-0015	IBM System/370 Model 145 Operating Procedures
GC38-0025	IBM System/370 Model 158 Operating Procedures
GC38-0030	IBM System/370 Model 168 Operating Procedures
GC38-0229	Operator's Library: OS/VS2 MVS System Commands
GC38-1000	OS/VS Message Library: Mass Storage System Messages
GC38-1002	OS/VS Message Library: VS2 System Messages
GC38-1007	OS/VS Message Library: Linkage Editor and Loader Messages
GC38-1008	OS/VS Message Library: VS2 System Codes
GC38-1045	OS/VS Message Library: VS2 EREP Messages
GC38-1046	OS/VS Message Library: VS2 TSO Terminal Messages
GY21-0012	OS BSAM Logic for IBM 1419/1275
GY21-0013	OS Data Management Macro Logic for IBM 1285/1287/1288
GY30-3000	IBM 3735 Programmable Buffered Terminal: Form Description Macro Instruction and Form Description Utility: Program Logic Manual (OS, DOS, and VS Systems)
SB0F-8210	OS/VS2 System Logic Library: Volumes 1-7
SYB8-0606	OS/VS2 Data Areas (Microfiche)
SY24-5162	OS/VS IBM 3886 Optical Character Reader Model 1 Logic
SY24-5163	OS/VS IBM 3890 Document Processor Logic
SY24-5167	OS/VS2 IBM 3540 Logic
SY24-6000	OS/VS2 JES2 Logic
SY26-3814	OS/VS Loader Logic
SY26-3815	OS/VS Linkage Editor Logic
SY26-3820	OS/VS2 Checkpoint/Restart Logic
SY26-3823	OS/VS2 I/O Supervisor Logic
SY26-3825	OS/VS2 Virtual Storage Access Method (VSAM) Logic
SY26-3826	OS/VS2 Catalog Management Logic
SY26-3827	OS/VS2 OPEN/CLOSE/EOV Logic

SY26-3828	OS/VS2 DADSM Logic
SY26-3831	OS/VS2 BDAM Logic
SY26-3832	OS/VS2 SAM Logic
SY26-3833	OS/VS2 ISAM Logic
SY26-3834	OS/VS2 VIO Logic
SY27-7241	OS/VS Graphics Problem-Oriented Routines Logic
SY27-7242	OS/VS Graphics Subroutine Package Logic
SY27-7246	OS/VS BTAM Logic
SY27-7256	Introduction to VTAM Logic
SY27-7260	OS/VS2 Graphics Access Method Logic
SY27-7263	VTAM Data Areas
SY27-7269	OS/VS2 MVS VTIOC and TCAS Logic
SY28-0612	OS/VS2 JES3 Logic
SY28-0621	OS/VS2 VTAM Logic
SY28-0623	OS/VS2 System Initialization Logic
SY28-0713	OS/VS2 System Logic Library: Volume 1 of 7
SY28-0714	OS/VS2 System Logic Library: Volume 2 of 7
SY28-0715	OS/VS2 System Logic Library: Volume 3 of 7
SY28-0716	OS/VS2 System Logic Library: Volume 4 of 7
SY28-0717	OS/VS2 System Logic Library: Volume 5 of 7
SY28-0718	OS/VS2 System Logic Library: Volume 6 of 7
SY28-0719	OS/VS2 System Logic Library: Volume 7 of 7
SY28-0643	OS/VS2 Service Aids Logic
SY28-0650	OS/VS2 Terminal Monitor Program and Service Routines
SY28-0651	OS/VS2 TSO Command Processor Logic: Volume I (ACCOUNT)
SY28-0652	OS/VS2 TSO Command Processor Logic: Volume IV
SY28-0676	OS/VS2 OLTEP Logic
SY28-0678	OS/VS2 SYS1.LOGREC Error Recording Logic
SY28-0685	OS/VS System Modification Program (SMP) Logic
SY28-0773	OS/VS Environmental Recording, Editing and Printing (EREP)
SY30-2040	OS/VS2 TCAM Logic
SY33-8041	OS/VS and VM/370 Assembler Logic
SY33-8548	OS/VS2 TSO Command Processor Logic: Volume 2 (EDIT)
SY35-0004	OS/VS2 TSO Command Processor Logic: Volume III (TEST)
SY35-0005	OS/VS Utilities Logic
SY35-0010	OS/VS2 Access Method Services Logic
SY35-0014	Mass Storage Control (MSC) Trace Reports Logic
SY35-0016	Mass Storage Control Table Create Logic

The handbook specifically omits the following general reference topics, which are covered in the *System/370 Reference Summary* (card), GX20-1850:

Machine instructions
Extended mnemonic instructions
CNOP alignment
Assembler instructions
Summary of constants
EDIT and EDMK pattern characters
Channel commands
EBCDIC translation table
Machine instruction formats
Control registers
CCW
Dynamic address translation
Hexadecimal and decimal conversion

Note: If you use only one order number, you will receive only that volume. To receive all three volumes, you must use the three order numbers or the following form number: GB0F-8211.

A handbook-sized binder, order number S229-4124, may be purchased from IBM. Customers may order it through their marketing representative. IBM personnel should order the binder from Mechanicsburg.

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

RACF - Resource Access Control Facility Program Number 5740-XXH.

MVS/System Extensions Program Number 5740-XE1.

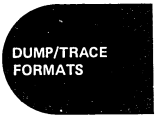
The date for this publication is November 30, 1978. Only supplements and TNLs with dates later than November 30, 1978, apply to this publication.

Contents

Figures


PROBLEM
CATEGORIES


DEBUGGING
AIDS


DUMP/TRACE
FORMATS


ERROR
INDICATORS


GENERAL
REFERENCE


CONTROL
BLOCK CHAINS

Summary of Amendments for GC28-0708-1

General

This edition has been reorganized into a three volume publication. See the Preface and Contents for the basic design and setup.

Specific

- Volumes 1, 2, and 3 incorporate maintenance updates accumulated since the last revision. Also, the following SUs have been integrated into these volumes.

Scheduler Improvements	VS2.03.804
Supervisor Performance #1	VS2.03.805
Supervisor Performance #2	VS2.03.807
Data Management	VS2.03.808
IBM 3800 Printing Subsystem	VS2.03.810
TSO/VTAM	VS2.03.813
Service Data Improvements	VS2.03.817
3838 Vector Processing Subsystem	5752-829
3895 Device Support	5752-830
System Security Support	5752-832
Dumping Improvements	5752-833
MVS Processor Support	5752-851
Hardware Recovery Enhancements	5752-855
Interactive Problem Control System	5752-857
TSO/VTAM Level 2	5752-858
Data Management Support	5752-860

- Volume 1 incorporates program product information for MVS/System Extensions (5740-XE1) and highlights this information where applicable.
- Section 2 of Volume 2 (GC28-0709 or GC28-0752) Control Block Chains has been moved to Volume 1 (GC28-0708-1) as Section 6.
- Section 1 of Volume 2 (GC28-0709 or GC28-0752) - "How to Find Information" has been moved. This information is now contained in the description of the individual data areas. Each Volume 2 and 3 data area greater than 2 pages in length will have a label-displacement list appended to it. This information already exists in OS/VS Data Areas (microfiche) and serves here as a replacement for the " How to Find Information" section.
- The publications summary (Section 6 in GC28-0708 or GC28-0751) has been deleted and replaced by a list of applicable publications in the Preface of Volume 1 (GC28-0708-1). A complete list of MVS publications can be obtained from the MVS Release Guide.

This edition has been reorganized for a three volume publication. See the Preface and Contents for the basic design and setup.

Section 1: Problem Categories and Analysis

This section summarizes problem categories and analysis. It includes the following subsections:

- Diagnostic Approach
- Serially Reusable Resources
- Resource Status Indicators

Program errors fall into three major categories:

- Unending loops
- Unending wait conditions
- Abnormal termination of a task or job step

The publication *OS/VS2 Message Library: VS2 System Codes*, outlines problem determination procedures to be followed for each of these problem categories. Problem determination is the activity required to identify a failing program or hardware unit, and to determine who is responsible for program support or hardware maintenance.

This section provides information that supplements, but does not replace or supercede, formal problem determination procedures.

Diagnostic Approach

The difficult way to analyze a dump is to look at each address space and task individually, looking at its request blocks and resume-PSWs to determine what it was doing (or waiting for) at the time of the dump. The objective of this section is to describe a diagnostic approach that will quickly determine which task or address space is preventing normal system operation.

One way to start debugging is to look at system and address space status indicators. Part 1 of Figure 1 shows the control blocks that contain system and address space indicators in effect during normal system operation. Part 2 of Figure 1 shows the control blocks containing status indicators for the system and address spaces after an abnormal operation. Figure 1 acts as a recall mechanism; it does not represent all the control blocks active in a system (only one TCB on the chain is shown, for example), nor all offsets into the control blocks. Specific values can be located in Volume 2 or Volume 3, or on microfiche in OS/VS2 Data Areas.

Serially Reusable Resources

Many system resources can only be used serially — that is, by only one task at a time. A system loop or wait generally occurs because a task has control of such a resource and never finishes with it or gives it up. Other tasks that require the resource are thus unable to continue processing.

Following are some examples of serially reusable resources.

Logical Section of the Processor: The responsibility for determining which task gets control of the processor belongs to the dispatcher. The dispatcher's decision is based on such things as priority, dispatchability status, processor affinity, and the presence or absence of a wait count. If, due to an error condition, the ASCB, TCB ready queue, or ready TCB pointer is not correct, then those address spaces or tasks beyond the break in the queue will never get control of the processor.

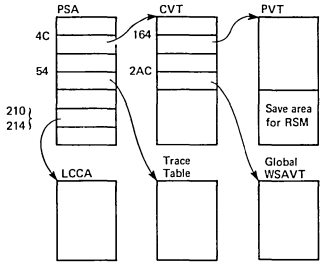
Channels: At any one time, a selector channel can handle one request for data transfer, while a multiplexer channel can handle one request per device. Since requests are asynchronous, the system must have some means to keep track of these requests and to schedule them for execution when the channel or device is available. This is provided by the logical channel queue, which consists of I/O queue elements (IOQs) that are created as the requests are generated.

Storage: There is actually a two-level competition for storage: hardware and software. The channels compete with each other and with the processor; any channel has priority over the processor, and the channels have priority with respect to each other. It is possible for a hardware error in a channel to prevent a channel operation from finishing; if that operation has already disabled the processor, preventing references to storage, the further execution of instructions is impossible. The system light stays on continuously, and the system fails to enter the manual state when the STOP button is pressed. System reset clears this condition, which cannot be accurately detected from a dump alone.

The software competition involves control of both real and virtual storage. OS/VS2 handles this competition in a manner that prevents lockouts and permits the execution of tasks that have control of the storage they require.

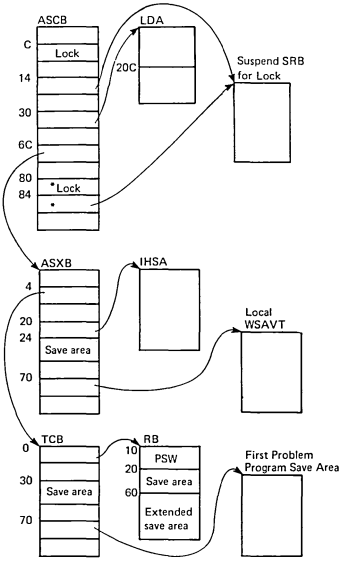
Miscellaneous Resources: Many OS/VS2 functions require that resources be used serially to preserve the integrity of data and to ensure normal operation. Examples of these functions are allocating and unallocating devices, dispatching work using the TCB queue, updating of VTOCs, using the catalog and system data sets; writing to the operator; and so forth. The system provides for these functions in two ways: (1) The ENQ and DEQ macro instructions are used to construct control blocks (QCBs and QELs) that indicate the name of the resource, the task requesting it, and whether the request is for shared or exclusive use of the resource. For exclusive requests, only the first-queued task is allowed to execute. If a task requests exclusive control of several resources, it is not allowed to execute until it has control of all of them. (2) Locks are used to serialize the use of control blocks associated with the dispatching of work. For example, the local lock is used to serialize the use of the TCB.

System Level



- PSA – Contains PSW, interruption codes, registers, and register save area for lock manager and FLIHs.
- LCCA – Registers saved by FLIHs, program check OPSW.
- PVT – Work save area for RSM.

Address Space Level



ASCB – Contains lock indicator.

LDA – Contains 9-64 byte work save area for GETMAIN/FREEMAIN.

ASXB – Contains 72 byte save area.

IHSA – Contains save area for locally locked interrupts, PSW, registers, and FRR stack.

RB – Contains PSW, register save area, extended save area.

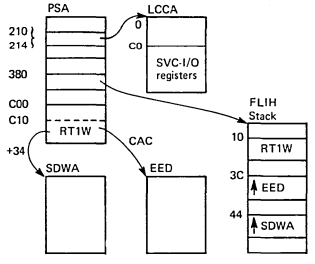
TCB – Contains task mode register save area.

WSAVT – Contains table of save area addresses.

* – Indicates changes that apply to MVS/System Extensions

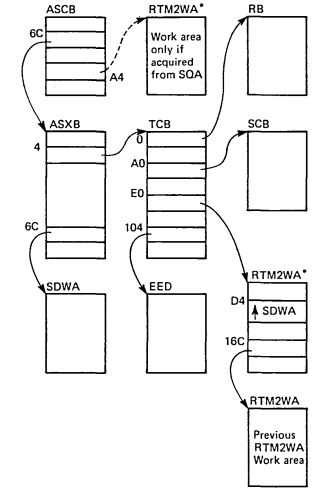
Figure 1. System and Address Space Status Indicator Locations (Normal Status Areas) (Part 1 of 2)

System Level



- LCCA – Contains program check registers and PSW, SVC and I/O registers.
- PSA – Contains PSWs, interrupt codes, super flags, pointer to the current stack, and pointer to the FLIH stack. Location C00 begins normal stack, location C10 begins RT1W. Location 380 points to the current stack.

Address Space Level



- RB – Contains flags, PSW, and registers.
- RTM2WA – Contains error data.
- SDWA – Contains indicators for errors in locally locked routines.
- TCB – Contains completion code, flags, and registers.

*RTM2WA normally chained off of the TCB but also chained off the ASCB if no LSQA is available.

Figure 1. System and Address Space Status Indicator Locations (Error Status Area) (Part 2 of 2)

Resource Status Indicators

Figure 2 defines a simple but logical approach to determining the current use of serially reusable resources. The following text is a list of the more important resources, and associated status indicators. (For a list of ENQ/DEQ names, associated resources, and requesting modules, refer to the serialization summary in Section 5.)

Processor Resources

1. Current ASID (address space identifier)
   - The PSAANEW/PSAAOLD field of the PSA is a doubleword new/old ASCB pointer.
   - The ASCBASID field of the ASCB is the ASID (2 bytes).
2. Current TCB (task control block)
   - The PSATNEW/PSATOLD field of the PSA is a doubleword new/old TCB pointer. (If the old TCB pointer, PSATOLD, is zero, an SRB has been dispatched.)
   - If the TCBRBP field of the TCB points to itself (instead of an RB), then the TCB is the pseudo-wait TCB and is not chained to any other TCB.
3. TCB Chain (by priority for each address space)
   - Location 4C hex points to the CVT.
   - The CVTASVT field of the CVT points to the ASVT.
   - The ASVTENTY field of the ASVT begins a series of one word entries that point to ASCBs (one for each active ASID).
   - The ASCBTNEW field of the ASCB points to the first ready TCB on the ready TCB queue. When no ready tasks exist, ASCBTNEW points to the last TCB. This pointer is never zero (applies to MVS/System Extensions).
   - The ASCBASXB field of the ASCB points to the ASXB.
   - The ASCBTCBS field of the ASCB in the number of ready TCBs not requiring the local lock (applies to MVS/System Extensions).
   - The ASCBTCBL field of the ASCB is the number of ready TCBs requiring the local lock (applies to MVS/System Extensions).
   - The ASXBFTCB field of the ASXB points to the first TCB in the TCB ready queue.
   - The ASXBLTCB field of the ASXB points to the last TCB.
   - The TCBTCB field of the TCB points to next TCB.
   - The TCBBACK field of the TCB points to the previous TCB.
   - The last TCB in the chain has a fullword of zeros at TCBTCB; the first TCB has a fullword of zeros at TCBBACK.
4. Subtask Chains (end of chain is always zero)
   - The TCBOTC field of the TCB points to the mother TCB of this TCB.
   - The TCBLTC field of the TCB points to the daughter TCB most recently attached.
   - The TCBNTC field of the TCB points to the sister TCB attached by the common mother.
   - The error task TCB is the only TCB without a mother TCB. In an address space created by a START, MOUNT, or LOGON request, the error task is the region control task (RCT).
5. Dispatching
   - Task dispatchability flags are in the TCBFLGS4 and TCBFLGS5 fields of the TCB. If any bit in these two bytes is set to one, the TCB is nondispatchable. (If bit 7 of TCBFLGS5 is set to one, the reason for nondispatchability is indicated by a flag bit set to one in the TCBNDSP1, TCBNDSP2, or TCBNDSP3 field of the TCB.
   - Tasks are dispatched based on the priority of the address space and then on the priority of the tasks within the address space.
   - Dispatching is in the following four levels.
     – Global SRB dispatcher (first).

- Local SRB dispatcher (second).
- Local supervisor dispatcher (third).
- Task (TCB) dispatcher (fourth).

Channel Resources

1. Channel restart queue (LCHWDTB - logical channel word entry)
   - The CVTILCH field of the CVT points to the first logical channel word entry.
   - Each entry is 32 bytes long.
   - The first word in the logical channel entry points to the first IOQE on the LCH (logical channel queue table). Is FFFFFFFF hex if empty.
   - The second word in the logical channel entry points to the last IOQE on the LCH (logical channel queue table).
   - The associated UCB is located by:
     - The IOQIOSB field of the IOQ points to the IOSB.
     - The IOSUCB field of the IOSB points to the UCB.

   Note: The IOSB has many flag bytes that contain information about the status of the I/O operation. For a complete description see the IOS logic manual, the diagnostic aids section, the topic called-informative IOSB fields.
   - The UCBFLA field of the UCB contains one byte of device status:
     - 80 hex is device busy.
     - 40 hex is device not ready.
     - A2 hex is device active-executing a channel program.
     - 08 hex is control unit busy.
     - 10 hex is sense pending.
   - Mount pending is indicated by 80 hex in the UCBDMCT field of the UCB.

Memory Resources — ENQ/DEQ control blocks

1. Major QCB.
   - The CVTFQCB field of the CVT points to first major QCB.
   - The MAJNMAJ field of the QCB points to next major QCB.
   - The MAJPMAJ field of the QCB points to the previous major QCB.
   - The MAJFMIN field of the QCB points to the first minor QCB.
   - The MAJLMIN field of the QCB points to the last minor QCB.
   - The MAJNAME field of the QCB is the major name (qname; 8 bytes).
2. Minor QCB.
   - The MINNMIN field of the QCB points to the next minor QCB.
   - The MINPMIN field of the QCB points to the previous minor QCB.
   - The MINFQEL field of the QCB points to the first QEL.
   - The MINLQEL field of the QCB points to the last QEL.
   - MINNAMEL field of the QCB is:
     - Length of minor name (rname; 1 byte).
     - Flags (1 byte - MINFLGS field of the QCB).

       80 hex is Scope = SYSTEM.

       40 hex is Scope = SYSTEMS.

       20 hex is Scope = STEP.

       10 hex is No ENQ Allowed (set by FRR).
     - ASID (for Scope = STEP only; 2 bytes - MINASID field of the QCB).
   - The MINNAME field of the QCB is the minor name (rname; 1-256 bytes).

3. QEL.
   - The QELNQEL field of the QEL points to the next QEL.
   - The QELPQEL field of the QEL points to the previous QEL.
   - The QELTCB field of the QEL points as follows:
     - If bit 0 = 0 QELTCB points to the TCB.
     - If bit 0 = 1 QELTCB points to a LIST QEL.
   - The QELQFLGS field of the QEL is:
     - QFLAGS (1 byte).
       80 hex is a shared QEL.
       40 hex is a "must complete" QEL.
       20 hex is a LIST QEL.
       10 hex is a RESERVE QEL
       00 hex is an exclusive QEL.
     - LFLAGS (1 byte).
       80 hex indicates that the ECB or RB has been posted.
       40 hex indicates an ECB request.
     - ASID (2 bytes - QELASID field of the QEL).

   LIST QEL only (QFLAGS = 20 hex):
   - QELSVRB field of the QEL points as follows:
     - If bit 0 = 0 points to the SVRB.
     - If bit 0 = 1 points to the ECB.
   - QELLCNT field of the QEL is:
     - Number of QELs for this request (2 bytes).
     - Number of QELs "waiting" (2 bytes - QELWCNT field of the QEL).
   - QEL + 18 hex points to the UCB (if QFLAGS contains 10 hex, indicating a RESERVE QEL).

4. WTO Buffers and WTOR reply queue elements.
   - WQE (write queue element).
     - The CVTCUCB field of the CVT points to the UCM.
     - UCM + 18 hex points to the first WQE (or zero).
     - UCM + 3C hex points to the last WQE (or zero).
     - UCM + 1C hex points to the first ORE (or zero).
     - WQE + 1 (3 bytes) points to the next WQE (or zero).
     - ORE + 1 (3 bytes) points to the next ORE (or zero).
     - UCM + 2D hex (1 byte) is maximum number of RQEs.
     - UCM + 2E hex (2 bytes) is maximum number of OREs.
     - UCM + 38 hex (2 bytes) is number of outstanding OREs.
     - UCM + 3A hex (2 bytes) is number of active WQEs.

5. SRB (service request block).
   - Global SRB. (enqueued on global service priority list)
     - The CVTGSPL field of the CVT points to GSPL.
     - The PSASVT field of the PSA points to the SVT.
     - The SVTGSPL field of the SVT points to the GSPL.
   - Local SRB. (execution based on an address space priority)
     - The ASCBSPL field of the ASCB points to the service priority list (SPL).
     - The ASCBLSPL field of the ASCB points to the LSPL.

6. SVC Table.
   - The CVTABEND field of the CVT points to the SCVT.
   - The SCVTSCVT field of the SCVT points to the SVC table.
   - Each entry in the SVC table is 2 words:
     - Bytes 0-3 contain the SVC entry point address.
     - Byte 4 (type and authorization):
       000. 0... is type 1 (unauthorized).
       000. 1... is type 1 (APF authorized).
       100. 0... is type 2 (unauthorized).
       100. 1... is type 2 (APF authorized).
       110. 0... is type 3 or 4 (unauthorized).
       110. 1... is type 3 or 4 (APF authorized).
       001. 0... is type 6 (unauthorized).
       001. 1... is type 6 (APF authorized).
       .... ..1. is a non-preemptive SVC.
     - Byte 6 (locks) to be obtained by the SVC first level interruption handler):
       80 hex is LOCAL lock.
       40 hex is CMS lock.
       20 hex is SRM lock.
       10 hex is SALLOC lock.
       08 hex is DISP lock.

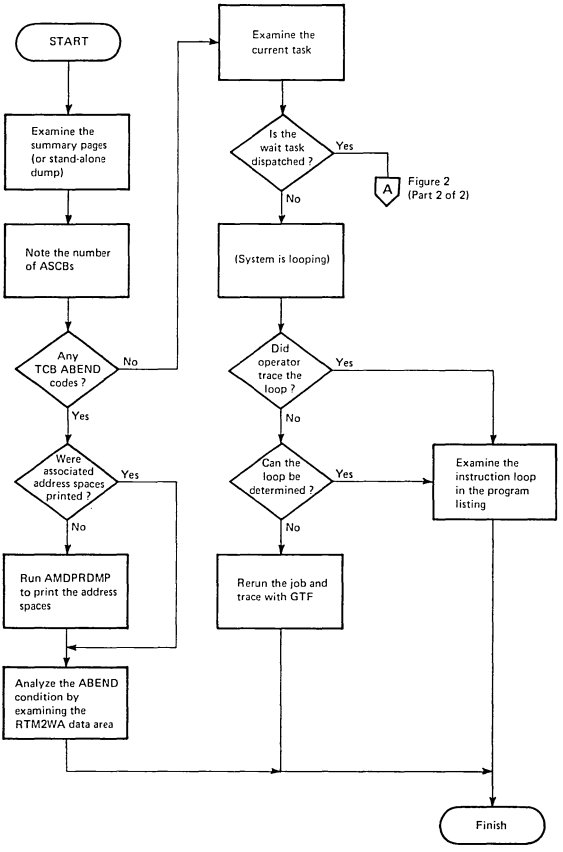


Figure 2. Diagnostic Approach (Part 1 of 2)

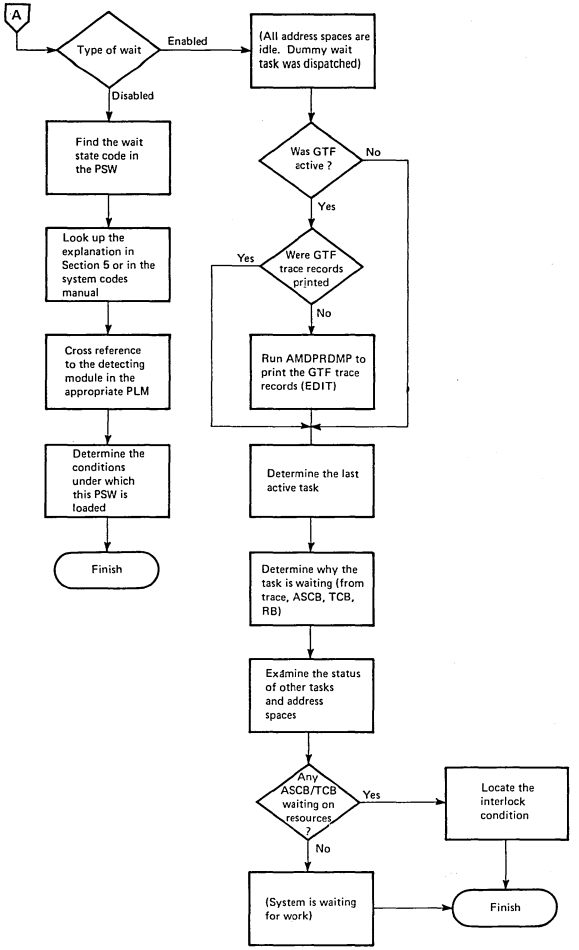


Figure 2. Diagnostic Approach (Part 2 of 2)

DEBUGGING AIDS

This section summarizes major OS/VS2 debugging aids. It includes the following subsections:

- Function Summary
  lists major debugging aids, briefly describes their functions, and indicates appropriate reference publications.
- Debugging Aids Summary (Figure 3)
- Service Aids Reference Summary
  is a reprint of OS/VS2 information from the service aids reference card.

Interactive Problem Control System - IPCS

The OS/VS2 MVS Interactive Problem Control System (IPCS) Selectable Unit (SU57) provides MVS installations with expanded capabilities for diagnosing software failures and facilities for managing problem information and status.

IPCS includes facilities for:

- Online examination of storage dumps.
- Analysis of key MVS system components and control blocks.
- Online management of a directory of software problems that have occurred in the user's system.
- Online management of a directory of problem-related data, such as dumps or the output of service aids.

IPCS runs as a command processor under TSO, allowing the user to make use of existing TSO facilities from IPCS, including the ability to create and execute command procedures (CLISTs) containing the IPCS command and its subcommands.

IPCS supports three forms of MVS storage dumps:

- High-speed stand-alone dumps produced by AMDSADMP.
- Virtual dumps produced by MVS SDUMP on SYS1.DUMPxx data sets.
- Virtual dumps produced by MVS SDUMP on data sets specified by the SYSMDUMP DD statement, when the Dumping Improvements Selectable Unit (SU33) is installed.

For information about IPCS, refer to the *OS/VS2 MVS Interactive Problem Control System (IPCS) User's Guide and Reference*.

Function Summary

Figure 3 lists debugging and debugging-related facilities by type and function. For each function, the text describes the role of each facility listed in the figure. The figure identifies publications where each facility is described in detail.

Dumping

Dumping functions are of five types:

- SNAP dumps
- ABEND dumps
- SVC dumps
- Stand-alone dumps
- SPZAP dumps

Snap Dumps

The SNAP macro instruction dumps a specified task to a specified data set.

The SNAP macro instruction can be used with various operands that govern the content of the dump.

For JES3, RJPSNPS spools the channel-end data to the JES3 queue volumes and prints this data at termination. CBPRNT snap dumps control blocks of JES3 and MVS. DEBUG snap dumps selected JES3 control blocks to the DEBUG data set.

Abend Dumps

Functions are provided by:

- SYSABEND, SYSMDUMP, SYSUDUMP, and JESSADMP (DD statements)
- IEAABD00, IEADMP00, and IEADMR00 (SYS1.PARMLIB members)
- AMDPRDMP (service aid)
- CHNGDUMP and CANCEL (operator commands)
- ABEND, CALLRTM, and SETRP (macro instructions)

SYSABEND, JESSADMP, SYSMDUMP or SYSUDUMP (DD statement):

- Defines a data set to contain an ABEND dump.

IEAABD00 (SYS1.PARMLIB member):

- Defines content options for SYSABEND dumps. In effect, these options are added to options specified in ABEND, CALLRTM, and SETRP macro instructions. Unless changed by the installation, IEAABD00 defines dump options as SDATA=(LSQA,ENQ,CB,TRT,DM,IO,ERR), PDATA=(ALLPDATA). A dump defined by these options alone would include the local system queue area (LSQA, SUBPOOL 229 and 230), ENQ control blocks (ENQ) and other control blocks (CB) for the terminated task, the GTF or system trace table (TRT), data management control blocks (DM), I/O control blocks (IO), recovery/termination error control blocks (ERR), the PSW, registers, save area trace, pack and link pack areas, and virtual storage subpools 0-127, 252 for the terminated task (ALLPDATA).

IEADMP00 (SYS1.PARMLIB member):

- Defines content options for SYSUDUMP dumps. In effect, these options are added to options specified in ABEND, CALLRTM, and SETRP macro instructions. Unless changed by the installation, IEADMP00 specifies the same options as IEAABD00, with these exceptions: LSQA and IO are omitted.

IEADMR00 (SYS1.PARMLIB member):

- Defines SVC dump options for SYSMDUMP dumps. SYSMDUMP dumps are unformatted ABEND dumps that can be formatted by AMDPRDMP service aid. In effect, these options are added to options specified in ABEND, CALLRTM, and SETRP macro instructions. Unless changed by the installation, IEADMR00 specifies SDATA=(NUC,SQA,LSQA,SWA,TRT,RGN).

CHNGDUMP (operator command):

- Changes the dump content options originally specified in SYS1.PARMLIB, in the system option list. The options override or are added to the options specified by the ABEND, CALLRTM, SETRP macros, and the associated parmlib member. The changes remain in effect until a re-IPL, or until changed by a subsequent CHNGDUMP command.

CANCEL (operator command):

- Cancels a job or TSO terminal session.
- Abnormally terminates the current job step.
- Optionally requests an ABEND dump.

*CANCEL (JES3 operator command):

- Cancels a JES3 dynamic support program.
- Abnormally terminates the job step.
- Provides an optional ABEND dump.

*FAIL (JES3 operator command):

- Terminates a JES3 dynamic support program.
- Used when a program fails to respond to *CANCEL.
- Provides an optional ABEND dump.

*DUMP (JES3 operator command):

- Forces a dump of the global processor.

AMDPRDMP (service aid - MVS problem program):

- Formats and prints the unformatted ABEND dump contained in the data set specified by the SYSMDUMP DD statement.

ABEND (macro instruction):

- Abnormally terminates the current task and its subtasks or (optionally) all the tasks for the current job step.
- Optionally requests an ABEND dump for all terminated tasks.
- Optionally specifies dump content options.

CALLRTM (macro instruction):

- Invokes the recovery termination manager for a task or an address space.
- Optionally requests an ABEND dump for all terminated tasks.
- Optionally specifies dump content options.

SETRP (macro instruction):

- Sets return parameters for return to the recovery termination manager by the FRR or ESTAE/ESTAI routine.
- Optionally requests an ABEND dump for all terminated tasks or for the current task if a retry is to be performed.
- Optionally specifies dump content options.

SVC Dumps

Functions are provided by:

- DUMP system parameter
- SYS1.DUMPnn data sets
- SDUMP macro instruction
- CHNGDUMP, DUMP, and SLIP operator commands
- AMDPRDMP service aid

DUMP (system parameter):

- Specifies virtual address spaces to be dumped to specified data sets.
- Specifies whether SYS1.DUMPnn data sets are available to the system.
- Specifies whether SYS1.DUMPnn data sets are on tape, on direct access storage, or both.

SYS1.DUMPnn (00<nn<09; system data set):

- Receives output of SVC dumps requested by means of the operator DUMP command.
- Receives output of all SVC dumps.
- Receives output of an SVC dump requested by an SDUMP macro instruction (unless the macro instruction specifies some other data set to receive the output).
- Receives output of an SVC dump taken as a result of matching a SLIP trap.

SDUMP (macro instruction):

- Dumps specified virtual address spaces to specified data set(s) (or to SYS1.DUMPnn data set(s)).
- Defines dump content options.
- Quiesces the system until any SUMDUMP (from branch entry to SVC dump) has been dumped.
- Optionally quiesces the system until the SQA and CSA have been dumped.

CHNGDUMP (operator command):

- Changes the dump-content and system-quiesce options in the system options list. These options override or are added to the options specified by SDUMP macro instructions. The changes remain in effect until re-IPL or until changed by a subsequent CHNGDUMP command.

DUMP (operator command):

- Requests an SVC dump of a specified address space. The system prompts the operator for the address space specification and dump content options. Output is directed to a SYS1.DUMPnn data set.

SLIP (operator command)

- SLIP allows an SVC dump to be scheduled when the conditions specified in the SLIP trap are met. The SVC dump may be tailored to the users needs by using the SDATA, LIST, SUMLIST, and ASIDLST keywords of the SLIP command.

AMDPRDMP (service aid; OS/VS2 problem program):

- Formats and prints an SVC dump contained in SYS1.DUMPnn or other data set specified in an SDUMP macro instruction.

Stand-Alone Dump

Functions are provided by:

- AMDSADMP macro instruction
- STORE STATUS operating procedure
- Stand-alone dump program (SADMP)
- AMDPRDMP service aid

AMDSADMP (service aid; macro instruction):

- Generates a stand-alone dump (SADMP) program tailored to the user's needs. This program produces either low-speed (formatted) output to tape or printer, or high-speed (unformatted) output to tape.

STORE STATUS (operating procedure):

- Stores the processor timer, clock comparator, current PSW, prefix reqisters, general registers, floating point registers, and control registers into permanently assigned storage locations (absolute address range 0-4K). This operation permits the stored data to be properly recorded in a stand-alone dump.

Stand-Alone Dump Program (SADMP):

- Low-speed output: Writes a formatted dump to tape or printer. This dump includes real storage and control registers.
- High-speed output: Writes an unformatted dump to tape. The dump includes general, floating-point, control, and prefix registers for each processor; the contents of real storage; LSQA, SWA, subpool 229, and subpool 230 for each virtual address space.

AMDPRDMP (service aid; OS/VS2 problem program):

- Prints the information on a tape containing the preformatted SADMP low-speed output. (This function can also be performed by other programs, such as IEBPTPCH.)
- Formats and prints the SADMP high-speed output.

SPZAP Dumps

The AMASPZAP service aid program dumps

- a data set.
- a specific member of a partitioned data set.
- any portion of a data set residing on a direct access device provided that the record length is less than 32k bytes.

Tracing

Functions are provided by:

- System trace routine
- TRACE command
- GTF (Generalized Trace Facility)
- GTF cataloged procedure
- COMMNDxx and GTFPARM (members of SYS1.PARMLIB)
- GTRACE macro instruction
- AMDPRDMP service aid
- Instruction stepping
- JES3 macro (IATXTRC)

System Trace Routine:

- Records system events in the system trace table. Events include SIO operations; I/O, SVC, program, and external interruptions; TCB and SRB dispatching; SVC returns. The system trace is started automatically during system initialization.

TRACE (OS/VS2 operator command):

- Causes the system trace to remain active or to terminate after master scheduler initialization. (If no TRACE command is issued before master scheduler initialization is completed, the system trace routine is terminated automatically).

GTF (Generalized Trace Facility; started task):

- Suspends the system trace routine (if active) until GTF is stopped.
- Records system events in the GTF private address space or in an external GTF trace data set.
- Records problem program events signalled by GTRACE macro instructions.

GTF cataloged procedure (SYS1.PROCLIB member):

- Defines GTF as a task that can be invoked by a START command.
- Refers to SYS1.PARMLIB member GTFPARM for trace options, unless a different member is specified in the START command.
- Specifies GTF parameters MODE=EXT,DEBUG=NO,TIME=NO (unless changed by the installation). These parameters specify recording in an external data set (MODE=EXT); no termination in the event of a recoverable error in GTF (DEBUG=NO); time-stamping of each block of logical trace records, but no time-stamping of individual records (TIME=NO). These options can be overridden by the START command.

COMMNDxx (SYS1.PARMLIB member, selected by system parameter (CMD=xx)):

- Contains commands to be issued automatically on completion of master scheduler initialization.
- Provides a timely means of issuing the TRACE or START GTF command.

GTFPARM (SYS1.PARMLIB member):

- Specifies GTF trace options, which can be revised by the operator. Unless changed by the installation, GTFPARM defines options as TRACE=SYSM,DSP,PCI,SRM,TRC,USR. These options request GTF to record the following events: entry to recovery routines, SIO operations, and I/O, SVC, program, and external interruptions (SYSM); TCB and SRB dispatching (DSP); PCI interruptions (PCI); entries to the system resources manager (SRM); events related to GTF (TRC); user events signalled by the GTRACE macro instruction (USR). The option SYSM specifies minimal recording for all events except user events (which generate records of up to 256 bytes).

GTRACE (macro instruction):

- Records system or problem program data in GTF trace buffers.

AMDPRDMP (service aid; OS/VS2 problem program):

- Formats and prints selected records from the GTF trace data set, or from GTF buffers in an SVC dump, or provides a high-speed stand-alone dump.

Instruction stepping (System/370 operating procedure):

- Executes one machine instruction at a time, allowing the operator to record pertinent data (PSW, CAW, etc.) from the system control panel. (This is a last-resort method of tracing an unending loop; it effectively suspends all productive use of the system, including both batch and realtime operations.)

JES3 macro-IATXTRC:

- Provides tracing information for JES3 failures. Information appears on a JES3 ABEND dump or on the operator console.

Trapping (Intercepting System Error Information)

Functions are provided by:

- AMASPZAP service aid
- COMPARE STOP operator control
- SLIP operator command

AMASPZAP (service aid; OS/VS2 problem program):

- Overlays instructions in a load module to cause invalid instructions or to allow for the validity testing of data in a control block.

COMPARE STOP (System/370 operator control):

- Stops the processor when a specified real or virtual address is selected by a processor or a channel.

SLIP (operator command):

- A SLIP trap allows the specification of an error circumstance and the action to be taken should that circumstance arise. Use the SLIP command to set, modify and delete SLIP traps.

Patching

Functions are provided by:

- AMASPZAP service aid
- ALTER/STORE operator command (System/370)
- SMP

AMASPZAP (service aid; OS/VS2 problem program):

- Modifies instructions and data in any load module that is a member of a partitioned data set.
- Updates the system status index (SSI) in the directory entry for a load module.
- Modifies specified records of any direct access data set.

ALTER/STORE (System/370 operator command):

- Alters contents of real or virtual storage, general or floating-point registers, or the instruction address portion of the current PSW. (For this procedure, the system must be placed in the stopped state, effectively suspending both batch and realtime operations.)

System Modification Program (SMP)

- Listing PTFs is provided by SMP via the HMASMP control statement "LIST PTF" in order to provide a listing of all module, macro, and superzap changes made to the system.

ICR/PTF Application

Functions are provided by:

AMAPTFLE (service aid; OS/VS2 problem program):

- Generates control statements and JCL needed to apply PTFs; the application function also invokes the linkage editor.
- Generates control statements and JCL needed to apply independent component releases (ICRs).

HMASMP (SMP; OS/VS2 problem program):

- Applies PTFs on a permanent or trial basis.
- Checks new PTFs for applicability to the user's system and maintains records of system status and contents.

Mapping, Formatting, and Printing

Functions are provided by AMBLIST service aid

See also:

- AMDPRDMP described under "Dumping" (for SVC and stand-alone dumps) and "Tracing."
- IFCEREP1 described under "SYS1.LOGREC Error Recording."

AMBLIST (service aid; OS/VS2 problem program):

- Produces formatted listings of object modules and load modules.
- Creates load module maps and cross-reference listings.
- Creates a load module summary, including entry point, APF access code, module attributes, and contents of module's system status index (SSI).
- Lists data stored in load module CSECT identification records (IDRs).
- Creates a map of the system nucleus or link pack area.

Displays

Functions are provided by:

- DISPLAY key (System/370 control panel)
- DISPLAY, MONITOR, and TRACK operator commands (OS/VS2)
- $D and $L operator commands (JES2)
- *X,DC and *X,DISPDJC operator commands (JES3)

DISPLAY key (System/370 control panel):

- Displays real storage, general-purpose registers, and floating-point registers on the system control panel. (For this operation, the system must be placed in the stopped state, effectively suspending both batch and realtime operations.)

DISPLAY (OS/VS2 operator command):

- Displays any of the following items on the operator's console:
  - -- Console configuration.
  - -- Summary of CONTROL command operands and functions.
  - -- System configuration (status of the processor, channels, devices, and storage).
  - -- Device allocation and status.
  - -- Current system status:
    - Active batch jobs, TSO users, MOUNT commands, and started tasks.
    - All batch jobs, MOUNT commands, started tasks, and active initiators.
    - Job names and active initiators, region boundaries.
  - -- System requests:
    - Identification number and (optionally) a list of all system requests waiting for replies.
    - Units waiting for mount requests or operator intervention.
  - -- Local time and date.
  - -- Terminal activity: number of active TSO users and (optionally) their user identification numbers.
  - -- Domain descriptor table.
  - -- SLIP traps.
  - -- SYS1.DUMPxx data set status and titles.
  - -- The current system dump options for SYSABEND, SYSMDUMP SYSUDUMP, and SVC dumps.

MONITOR (OS/VS2 operator command):

- Requests a continuous display of data set status (in mount/demount messages):
  - -- Names of nontemporary data sets.
  - -- Available space on a direct access device.
  - -- Names of data sets, and volume serial numbers of data sets with dispositions of KEEP, CATLG, and UNCATLG.
- Requests a continuous display of job status:
  - -- Names of jobs initiated and terminated.
  - -- Time of job initiation or termination.

TRACK (OS/VS2 operator command):

- Periodically displays system status on a graphic console:
  - --Active batch jobs, MOUNT commands, started tasks, and active initiators.
  - --Active TSO users.
  - --Job names and V=R region boundaries.

$D (JES2 operator command):

- Displays any of the following items on the operator's console:
  - --Status of an initiator and associated job classes.
  - --Status of JES2-controlled non-direct access I/O devices.
  - --Status and other information about currently active jobs (including batch jobs, system tasks, TSO users, and jobs being processed on unit record devices).
  - --Job queue information:
    - Job number, name, status, class, and priority.
    - Number of jobs in a particular queue.
    - Percentage of spool disk utilization.

--Queued output requirements (forms, carriage, train; route code; number of jobs queued for each output class).

--Unsatisfied JES2 requests.

--Information for a specific job (job number, name, status, class, and priority).

- Sends a message to a remote terminal.

$L (JES2 operator command):

- Lists the number of output elements waiting to be processed in each queue (released or held).

*X,DC (JES3 operator command)

- Displays selected areas of JES3 storage on the operator console or prints output on the SYSOUT data set.

*X, DISPDJC (JES3 operator command)

- Displays dependent job control information.

SYS1.LOGREC Error Recording

Functions are provided by:

- SYS1.LOGREC data set
- IFCDIP00 and IFCEREP1 service aids
- AMDPRDMP LOGDATA control statement
- MODE operator command

SYS1.LOGREC (system data set):

- Receives records of machine and program failure and recovery activity.

IFCDIP00 (service aid; OS/VS2 problem program):

- Initializes SYS1.LOGREC during system generation.
- Reinitializes SYS1.LOGREC if destroyed.
- Reallocates SYS1.LOGREC to increase or decrease space allocation.

IFCEREP1 (service aid; OS/VS2 problem program):

- Selects SYS1.LOGREC records by date, record type, device type, or device address.
- Accumulates records in a history data set.
- Formats and prints records, record summaries, and special reports.
- Prints the error id provided by RTM and allows the correlation of software and related hardware records with an associated SVC dump and/or console log entry.

AMDPRDMP (service aid - MVS problem program):

- Formats and prints in-storage LOGREC entries for SYSMDUMPs, SVC dumps and stand-alone dumps when the LOGDATA control statement is supplied and the LOGREC buffer is available in the dump.

  Note: The LOGDATA control statement requires the EREP Selectable Unit (5752-827) be on the system.

MODE (OS/VS2 operator command):

- Sets recording mode (RECORD or QUIET) for each processor.
- Sets number of machine-check interruptions to be recorded before the operator is notified.
- Displays the machine-check event counters and recording modes for each processor.

Restart and Recovery

Functions are provided by:

- RESTART key (System/370 control panel)
- ATTACH, DETACH, STAE, ESTAE, SETFRR, SETRP, and SPIE macro instructions

RESTART key (System/370 control panel):

- Stores the current PSW in FLCROPSW (PSA + 8), and loads a new PSW from FLCRNPSW (PSA + 0). Processing resumes (without a system reset) under control of the new PSW. If the QUIESCE command has been issued or a restartable wait state has been loaded, processing continues as normal. Otherwise, recovery termination is invoked.

ATTACH (macro instruction):

- Creates a subtask.
- Optionally establishes a STAI or ESTAI routine for the current task; this routine is entered if the subtask (or any of its subtasks) is scheduled for abnormal termination. A STAI or ESTAI routine can establish a retry address, or permit termination to continue (with or without execution of any additional STAI or ESTAI routines). Multiple STAI or ESTAI routines are executed in LIFO order, after execution of any applicable STAE or ESTAE routines.

DETACH:

- Removes a completed subtask from the system.
- Abnormally terminates a non-completed subtask and determines whether the subtask's STAE, ESTAE, STAI, or ESTAI routine (if any) should be executed.

STAE (macro instruction):

- Establishes a routine to be entered if the current task is scheduled for abnormal termination. A STAE routine can establish a retry address or can permit termination to continue. In the case of multiple STAE routines, only the most recently established routine is executed.

ESTAE routines. (macro instruction):

- Establishes a routine to be entered if the current task is scheduled for abnormal termination. An ESTAE routine can establish a retry address or permit termination to continue (with execution of any additional recovery routines). Multiple STAE or ESTAE routines (and associated retry routines, if any) are executed in LIFO order.

SETFRR (macro instruction):

- Establishes a functional recovery routine (FRR) for a program that operates disabled in supervisor state key 0, owns a lock, or executes under an SRB.

SETRP (macro instruction):

- Sets return parameters for return to the recovery termination manager by FRR, ESTAE, or ESTAI routine.

SPIE (macro instruction):

- Establishes a routine to be entered if the current task encounters a program check interruption.

Debugging Aids Summary

Figure 3 is intended to be a quick guide to the major or most-used debugging aids. For complete information about these debugging aids see the applicable manual indicated.

Facilities	Operator Facilities	Service Aids	Macro Instructions	Other Facilities
Dumping: SNAP Dump			SNAP[7]	CBPRNT[11] RJPSNP[5] DEBUG[11]
ABEND Dump	CHNGDUMP[1] CANCEL[1] *CANCEL[11] *FAIL[11] *DUMP[11]	AMDPRDMP[3]	ABEND[7] CALLRTM[12] SETRP[12]	SYSABEND DD[9] SYSMDUMP DD[9] SYSUDUMP DD[9] JESSADMP[11] IEAABD00[10] IEADMP00[10] IEADMR00[10]
SVC Dump	CHNGDUMP[1] DUMP[1]	AMDPRDMP[3]	SDUMP[12]	DUMP param[10] SYS1.DUMPnn[10]
Stand-Alone Dump	STORE STATUS[2]	Stand-Alone Dump Program[3] AMDPRDMP[3]	AMDSADMP[3]	
Listing PTFs		HMASMP[4]		
SPZAP Dump		AMASPZAP[3]		
Tracing	TRACE[1] INSTRUCTION STEP[2]	GTF[3] AMDPRDMP[3]	GTRACE[3,7] IATXTRC[11]	System Trace Routine GTF catproc[3] GTFPARM[10] COMMNDxx[10]
Trapping	COMPARE STOP[2] SLIP[1,12]	AMASPZAP[3]		
Patching	ALTER/ STORE[2]	AMASPZAP[3]		
ICR/PTF Application		AMAPTFLE[3] HMASMP[4]		
Mapping, Formatting, and Printing		AMBLIST[3] [AMDPRDMP]* [IFCEREP1]*		
Displays	DISPLAY Key[2] Command[1] MONITOR[1] TRACK[1] SD[1] SL[1] *X,DC[11] *X,DISPDJC[11]			
SYS1.LOGREC Error Recording	MODE[1]	IFCDIP00[5] IFCEREP1[5]		SYS1.LOGREC[5]
Restart/Recovery	RESTART[2]		ATTACH[7,12] DETACH[7] STAE[12] ESTAE[7,12] SETFRR[12] SETRP[7,12] SPIE[7]	

*For text description, see other functions for which this facility is indicated in the figure.

Reference Manuals

[1]GC38-0229
[2]GC38-0015 (for System/370 Model 145)
GA22-6966 (for System/370 Model 155)
GC38-0025 (for System/370 Model 158)
GA22-6969 (for System/370 Model 165)
GC38-0030 (for System/370 Model 168)
[3]GC28-0674
[4]GC28-0673
[5]GC28-0677
[7]GC28-0683
[9]GC28-0692
[10]GC28-0681
[11]GC28-0627
[12]GC28-0628

Figure 3. Debugging Aids Summary

Service Aids Reference Summary

How to Use This Summary

Symptom Table

The numbers in this table refer to the explanatory notes on the accompanying sheets. For each symptom, read from left to right to find out which functions of these service aids should be used to diagnose and fix the problem. For complete information about IFCDIP00 and IFCEREP1, see *OS/VS2 System Programming Library: SYS1.LOGREC Error Recording*, and *OS/VS Environmental Recording, Editing and Printing (EREP) program logic*, respectively. For complete information about other service aids, see *OS/VS2 System Programming Library: Service Aids*.

SYMPTOM	INFORMATION GATHERING		PATCHING			MAPPING, FORMATTING AND PRINTING		
	SADMP	GTF	SPZAP	DIP00	HMASMP	PRDMP	LIST	EREP0
Warm Start Failure	0	–	5	–	–	10c-e	–	–
Scheduler ABEND	–	1	5	–	–	11	12,13	–
Writer ABEND	–	1	5	–	–	11	–	–
Problem Program ABEND	–	3	5	–	–	11	13	–
Recursive ABEND	0	1	5	–	–	10a,c-d,11	13,16	–
Disabled Loop	0	1	–	–	–	10c-e,11	–	–
Problem Program Loop	–	3	–	–	–	–	13	–
Large Loop with I/O	0	1	5	–	–	10a,c-e,11b-d	16	–
DAR Loop (VS1 Only)	0	1	–	8	–	10c&e,11	13	17
Hard Wait	0	1	–	8	–	10c-e	13	17
Enabled Wait	0	1	–	8	–	10b,11	13	17
Reader/Interpreter Failure	–	–	5	–	–	–	–	–
I/O Failure e.g. console)	0	2	5,7	–	–	10a-e,11b-d	16	17,18
Allocation Failure	0	–	5	–	–	10b-d	13	–
Enqueued Job Lost	–	2	–	–	–	10a&c-e,11	–	–
Chain Scheduling Problem	0	2	–	–	–	–	16	–
Access Method Failure	–	2	5	–	–	11	–	18
Data Management Prgm Check	–	1,3	–	–	–	11	13	–
Module Level Unknown	–	–	6	–	–	–	14	–
User Modification Unknown	–	–	6	–	–	–	15	–
Applying PTF	–	–	4	–	9	–	–	–
Applying ICR	–	–	–	–	–	–	–	–
Applying Local Fix	–	–	4	–	–	–	–	–
APAR Documentation	0	1,3	6	–	–	10a&c-e,11	14,16	–
Print SYS1.DUMP	–	–	–	–	–	10b-d,11	–	–
Capturing System before RE-IPL	0	–	–	–	–	10a-e,11	16	–

Information Gathering

SADMP

0. Dumps the contents of real or selected portions of virtual storage to a tape, which can be formatted and printed using PRDMP. SADMP can also produce a formatted dump directed to tape or the printer.

GTF

1. Traces all system events.
2. Traces selected events, such as I/O interruptions, SIO operations, etc.
3. Traces user programs with GTRACE macro instruction.

Patching

SPZAP (Super Zap)

4. Modifies data in a load module.
5. Sets traps by inserting invalid instructions or user-written SVCs.
6. Dumps load modules by CSECT to allow examination of the text.
7. Dumps selected data to verify the count, key and contents of the data.

DIP00

8. Reinitializes the SYS1.LOGREC data set if destroyed.

HMASMP (System Modification Program)

9. Applies PTFs to system and DLIB packs and lists the PTFs applied.

Mapping, Formatting, and Printing

PRDMP

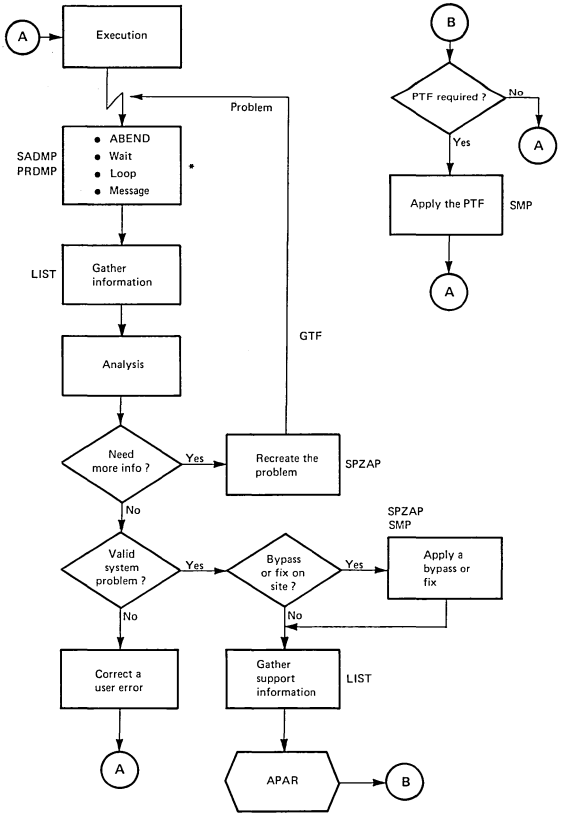
10. Formats and prints the following from SADMP unformatted (high-speed) dumps and SVC dumps:
    a. Link pack area.
    b. Queue control block trace.
    c. Major system data areas.
    d. Selected areas of storage by virtual or real address.
    e. Operating system nucleus.
11. Formats and prints selected records from the GTF trace data set or from trace buffers in a SYS1.DUMPxx or SADMP output data set. Records are selected by keywords such as:
    a. JOBNAME
    b. I/O
    c. SVC
    d. SIO

LIST

12. Lists specific object modules, load modules or load modules in a data set.
13. Maps control sections and overlay structure and lists cross-references within a load module.
14. Lists CSECT identification records for specific load modules.
15. Lists translation data, linkage editor modification data, or SPZAP modifications to control sections in a load module.
16. Maps link pack area.

EREP1

17. Selects, formats and prints records, record summaries and special reports from the SYS1.LOGREC data set, by record type.
18. Selects records by device type, date, or device address.



*For suspected hardware errors, use IFCEREP1 to gather information. If the SYS1.LOGREC data set is destroyed, reinitialize it using IFCDIP00.

Figure 4: Using Service Aids to Process System Problems

DUMP/TRACE
FORMATS

Section 3: Dump and Trace Formats

This section describes the dump and trace output of debugging aids described in Section 2. There are four subsections:

- ABDUMP Output
  (ABEND and SNAP Dumps)
- SADMP Output
  (Low-Speed Stand-Alone Dump)
- AMDPRDMP Output
  (SYSMDUMP, SVC Dump, and High-Speed Stand-Alone Dump)
- GTF Trace Records

Note:

1. ABDUMP output includes the output of the system trace routine. For a description of SPZAP dumps, refer to *OS/VS2 Service Aids*.
2. Throughout this section, variable data in dump and trace output is represented as follows:
   - h – represents one hexadecimal digit.
   - d – represents one decimal digit.
   - c – represents one EBCDIC character.
   - x – represents data whose format is unpredictable.
3. The reader should be aware that examples of dumps presented here may not (field-for-field) match the content of his/her dump.
4. The MVS/System Extensions program product modified and/or added information to the dump output. In the examples of dump output MVS/System Extensions information is highlighted by an "asterisk" to the left of the information.

```
(1) JOB SU50          STEP ST1          TIME 144548   DATE 77041    ID = 000                    PAGE 0001

(2) COMPLETION CODE       SYSTEM = 806

    PSW AT ENTRY TO ABEND  070C1000 00038146          ILC 2    INTC 000D

(4a) ASCB 00FED5C8
          +0    ASCB   C1E2C3C2   FWDP   00000000   BWDP   00FEC838   CMSF   00000000   SVRB   00ABE7A8   SYNC   0000010A
          +18   IOSP   00000000   TNEW   00FED698   CPUS   00000001   IOSQ   00030003   IODP   00070070   STOR   0F265C00
          +30   LDA    00ABF548   RSM    00FED4A8   CSCB   00FED680   TSB    00000000   EJST   00000001   XJST   8E490000
          +48   EWST   8A533935   XWST   9D992000   JSTL   000007D3   ECB    80ABEE58   UBET   00000000   TLCH   00000000
          +60   DUMP   00ABE3A0   FW1    FFFF0000   TMCH   00000000   ASXB   00A9F300   SWDP   00110000   SRBS   00000000
          +78   VSC    00000000   RCTP   00000001   LOCK   00000000   LSQH   00000000   QECB   00000000   MECB   40000000
          +90   OUCB   00FEC240   OUXB   00FEC2D0   FMCT   00230000   XMPQ   00000000   IQEA   00000000   RTWA   00000000
          +A8   MCC    03000000   JBNI   00FED400   JBNS   00FED6C0   SRQ    00000000   VGTT   00000000   PCTT   00000000
          +C0   SMCT   00000000   SWTL   0000023C   SRBT   00000000   ATME   19026000   *LSMQ  00000000   *LSPL  00000000
          +D8   *TCBS  00000000   *TCBL  00000000   *HPRB  00000000   *SRDP  00FEC918   *RESV  00FED3F8   *RESV  04090000

(4b) TCB  ABC080
          +0    RBP    00ABE6E0   PIE    00000000   DEB    00A91798   TIO  00A99020   CMP    80806000   TRN    00000000
          +18   MSS    00A8C3D0   PK-FLG 80010000   FLG    0000FFFF   LLS  00A8CF28   JLB    00000000   JPQ    00000000
          +70   FSA    01084FB0   TCB    00000000   TME    00000000   JST  00A8C080   NTC    00000000   OTC    00A8C920
          +88   LTC    00000000   IOE    00000000   ECB    00A8C204   TSF  20000000   D-PQE  00ABF548   AQE    00AA1108
          +A0   STAB   00A8D078   TCT    80A8C730   USER   00000000   SDF  00000000   MDID   00000000   JSCB   00A8C6F4
          +B8   RESV   00000000   IOBRC  00000000   XCPD   00000000   EXT  00000000   BITS   00000000   DAR    00000000
          +D0   EXT2   00A8C1A8   AECB   00000000   TIRB   00000000   BAK  00A8C920   RTMWA  00AA1628   IOTM   00000000
          +E8   TMSAV  00000000   A8CR   00000000   XSCT   80000040   FQE  00000000   SWA    00ABC320   STAW   00000000
          +100  BID    E3C3C240   RTM1   00000000   ESTA   00000000   UKY  00A8C440   *SQAF  0000FFFF   BYT1   08040000
          +118  RPT    00000000   DBTB   00000000   SWAS   00A92F48   SCB  00000000   RESV   00000000   RESV   00000000
          EXT2  GTF    00000000   SVAB   00000000   EVNT   00000000   *TQE 00000000   RESV   00000000   RESV   00000000
```

Figure 5. ABDUMP Output (Example) (Part 1 of 14)

ABDUMP Output (ABEND and SNAP Dumps)

ABDUMP output varies in accordance with dump content options, which are specified as follows:

- For ABEND dumps, content options are specified in the IEAABD00 and IEADMP00 members of SYS1.PARMLIB; in the ABEND, CALLRTM, and SETRP macro instructions; and in the CHNGDUMP operator command. (These facilities are described under "ABEND Dumps" in Section 2.)
- For SNAP dumps, content options are specified in the SNAP macro instruction.

Figure 5 is an example of ABDUMP output. Contents and related options are described below.

(1) Dump heading.

JOB cccccccc -- Job name.
STEP cccccccc -- Step name.
TIME ddddd -- Hour (dd), minute (dd), and second (dd) when ABDUMP began processing.
DATE ddddd -- Year (dd) and day of year (ddd).
ID=ddd -- Dump identification specified in SNAP macro instruction, or provided as follows by ABEND:
ID=000 -- Dump represents the terminating task.
ID=001 -- Dump represents a subtask of the terminating task.
ID=002 -- Dump represents the terminating task's originating task.
PAGE dddd -- ABDUMP output page number.

(2) COMPLETION CODE -- System or user completion code (for ABEND dumps only).

SYSTEM=hhh -- System completion code. (See "Abnormal Completion Codes" in Section 4.)

USER=dddd -- User completion code (from the ABEND macro instruction).

(3) This field is optional. The program status word requested by the PDATA=PSW option. An example of this information is not shown in the corresponding figure.

PSW AT ENTRY TO ABEND or SNAP hhhhhhhh hhhhhhhh -- PSW for the program that had control when the ABEND or SNAP macro instruction was executed.

ILC hh -- Instruction length code.

INTC hhhh -- Interruption code or SVC number. (See "Program Interruption Codes" in Section 4, or "SVC Summary" in Section 5.)

(4) Control block information requested by the SDATA=CB option.

(4a) ASCB hhhhhhhh -- Address space control block (ASCB).[1]

(4b) TCB hhhhhh -- Task control block (TCB).[1]

[1]Data area address followed by formatted fields. For a description of each field, refer to the appropriate data area description in Volume 2 or Volume 3. (Note that all request blocks are described under the heading "RB".)

```
ACTIVE RBS (4c)

PRB  BF12B8   FLG1 00000000    WC-L-IC 00020033   RESV   00000000   APSW   00000000   SZ-STAB 00110082   FL-CDE 00BF1408
              PSW 078D0000     0008602A           Q/TTR  00000000   WT-LNK 00BDAEA8
              RG 0-7     00BF15D8    00084FEE    00000040    00001464    00001440    000000F8    00BA7018    FD000000
              RG 8-15    00BDA120    80BF1708    00000000    00BF15E0    70F0F22A    00084F90    40F0F532    00BDA150
SVRB 00DDD8   FLG1 20000000    WC-L-IC 00020033   RESV   00000000   APSW   00000000   SZ-STAB 0019D022   FL-CDE 00000000
              PSW 070C1000     00E9F99A           Q/TTR  00000000   WT-LNK 00BF12B8
              RG 0-7     000863EE    8008600C    000865C0    000865C8    00000000    00000001    000865C8    00000064
              RG 8-15    00086590    0000000A    00000064    '00086CEE   40085CEE    00086248    40085FB4    92D61AB0
              EXTSA      000863EE    8008600C    00025160    00BDAEA8    000DDDD8    00FEEAE0    00031364    00FDB000
                         00BDA518    FF000368    4000CB01    80047000
              SCB        00000000    00000000    00000000    00000000    00000000   RESV    00000000
SVRB 00D888   FLG1 02000000    WC-L-IC 00020000   RESV   00000000   APSW   00000000   SZ-STAB 0019D022   FL-CDE 00000000
              PSW 070C2000     00E600BA           Q/TTR  00000000   WT-LNK 000DDDD8
              RG 0-7     000000E0    01BDAC18    00FF2020    0080E600    00BDAEA8    00BDAC18    50E9F564    00BDA818
              RG 8-15    00BB9FA0    00000000    00BB9FF8    00BB9FF8    00FDAB38    00BDAC8C    50EA0A3A    00000000
              EXTSA      00000000    00EDA4C0    0000021C    000D08EC    20FD0000    00000000    00000000    00000000
                         00000000    00000000    00000000    00000000
              SCB        00000000    00000000    00000000    00000000    00000000   RESV    00000000

(4d)
LOAD LIST
        NE 00BF1378   RSP-CDE 00FE9F70   CNT 00010001            NE 00BF1388   RSP-CDE 00FE5CD0   CNT 00010001
        NE 00BF1398   RSP-CDE 00FEB858   CNT 00010001            NE 00BF13C8   RSP-CDE 00FEB8C8   CNT 00010001
```

Figure 5. ABDUMP Output (Example) (Part 2 of 14)

(4c) ACTIVE RBS -- Request blocks (RBs) queued to the TCB.[1]
- IRB hhhhhh -- Interruption request block.
- PRB hhhhhh -- Program request block.
- SVRB hhhhhh -- Supervisor request block.

(4d) LOAD LIST -- Load list elements (LLEs) queued to the TCB. Each line of print represents the contents of two LLEs; these contents are described under "LLE" in Volume 2.

[1]Data area address followed by formatted fields. For a description of each field, refer to the appropriate data area description in Volume 2 or Volume 3. (Note that all request blocks are described under the heading "RB".)

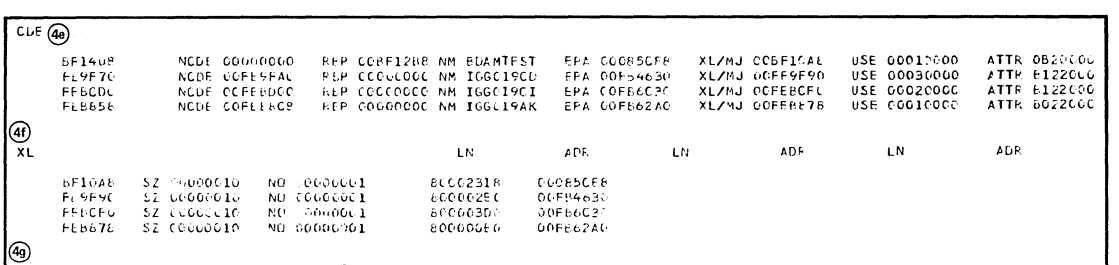


Figure 5. ABDUMP Output (Example) (Part 3 of 14)

(4e) CDE -- Contents directory entries (CDEs) queued to RBs and LLEs. Each line of print represents one CDE.[1]

(4f) XL -- Extent list entries (XTLSTs) queued to CDEs. Each line of print represents one XTLST.[1]

(4g) TIOT -- Task I/O table (TIOT) queued to the TCB. Each succeeding line of print represents one DD entry in the table. Contents are described under "TIOT" in Volume 3.

(4h) IQE -- Interruption queue elements (IQEs). Each line of print represents one IQE.[1] No example of this information is shown in the corresponding figure.

(4i) VSM -- Virtual storage management subpool queues for the task(s) that are dumped. The display includes:

SPQE -- Subpool queue element (SPQE).[1] Note that the subpool ID field (SPID) is printed in decimal format.

DQE -- Descriptor queue element (DQE). Fields are described under "DQE" in Volume 2.

FQE -- Free queue element (FQE). Fields are described under "FQE" in Volume 2.

[1]Data area address followed by formatted fields. For a description of each field, refer to the appropriate data area description in Volume 2 or Volume 3.

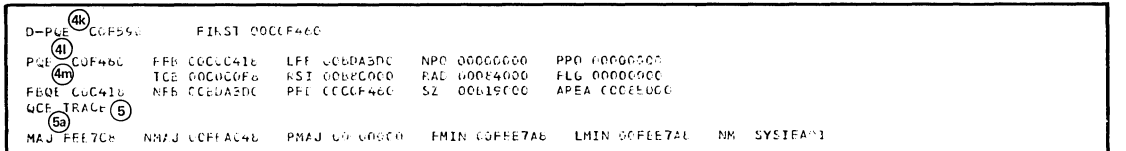


Figure 5. ABDUMP Output (Example) (Part 4 of 14)

(4k) D-PQE hhhhhh -- Dummy partition queue element (address minus eight bytes, followed by formatted fields).
FIRST hhhhhh -- Address of first PQE.

(4l) PQE hhhhhh -- Partition queue element (PQE) for storage in the address space.[1]

(4m) FBQE hhhhhh -- Free block queue element (FBQE).[1]

(5) QCB TRACE -- Trace requested by the SDATA=ENQ option. (Trace of queue control blocks and queue elements associated with the job step.)

(5a) MAJ hhhhhh -- Major queue control block (QCB).[1]

(5b) MIN hhhhhh -- Minor QCB queued to major QCB. Each succeeding line of print represents a queue element (QEL) queued to the minor QCB; fields are described under "QEL" in Volume 3.

(5c) TCB SUMMARY -- task control block summary.

[1]Data area address followed by formatted fields. For a description of each field, refer to the appropriate data area description in Volume 2 or Volume 3.

```
(5d)
-DEB----------------------------------------AT LOCATION  00AAE49C
------

-24     DEBAVT   00F31348 0004099F DEBPCIA  0004099F DEBCEA   0004099F DEBXCEA  0004099F
-10     PREFIX   00000000 00000000 EXTNSION 00AAE5A8 LENGTH   11       AMTYPE   20       TBLOF    0002
+0      TCBADR   06ACC180 NEXTDEB  10AAE524 IRBADR   C8000000 PATB     17000100
+10     USRPG    01000000 RRQ      FF000000 DCBADR   8F085664 APPADR   04AAE478
+20     18AAF1A0 00000000 00010008 000B00A3
+30     00010001 00000000 00000000 00000000
+40     C2C2C2C1 C3C8C3C3 C3C4C2C3 00000000
+50     00000000 00000000 00000000 00000000
+60     00000000

(5e)
-DCB-     (BSAM)                              AT LOCATION 00085664
-----

+10     0029336D 00000001 00004000 00000001 04000001 C0000000 00683474 00AAE49C
+30     92FA6DF0 00FAA770 10085600 00000000 58045070 00084EA8 01C2DBF0 00C2D5F0
+50     00000000 00C17138

(5f)
-IOB-     (SAM,BPAM-NORMAL SCHEDULING)     AT LOCATION  00084EB0
-----

-8      01084EA8 7F000000
+0      FLAG     0200     SENS     0000     ECBPT    00084EAC CSW      00000000 00000000
+10     START    00084ED8 DCBPT    00085664 RESTR    00000000 VARIABLE 00000000
```

Figure 5. ABDUMP Output (Example) (Part 5 of 14)

(5d) DEB -- Data extent block prefix and data extent block. Each line of print includes:

- The virtual address of the first printed word of stroage.
- Eight words of storage in hexadecimal format.
- The same eight words in EBCDIC format, with periods representing characters other than blank, A-Z, and 0-9.

If the dump is directed to a 3800 printer and the JCL specifies CHARS=DUMP on the SYSUDUMP, SYSABEND or SNAP-related DD statement, each line of print includes:

- The virtual address of the first printed word of storage.
- Sixteen words of storage in hexadecimal format, printed in groups of four words with the first character of each word underlined.
- The same 16 words in EBCDIC format, with periods representing characters other than blank, A-Z, and 0-9. This will appear as two 32-character groups separated by one blank.

Fields are described under "DEB" in Volume 2. (Note for JES2 and JES3, the DEB prefix is meaningless.)

(5e) DCB -- data control block pointed to by the preceding DEB. This information is formatted only for the DM option.

(5f) IOB -- I/O control block pointed to by the preceding DCB. This information is only formatted for the IO option.

```
(5i)  -RTM2WA----------------------------------AT LOCATION  00AA1AC8
       ------

      +0       D9E3D4F2 00AA1AC8 FF070000 00039048 00AA1FB8 00ABF5F0 00FEL658 0000C000
      +20      0000C000 00000000 00000000 00000000 00AA1FB8 00ADE5F0 00FE7060

      -EEL TYPE1 REGS AND PSW
      +3C      0000C000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      +5C      0000C000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      +7C      0000C000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      -----

      +8C      00000000 00000000 00000000

      -EEL TYPE3 MACHINE CHECK
      +98      00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      -----

      +B4      00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      +D4      00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      +F4      00000000 00000000 00000000

      -SNFPARMS
      +108     00000000 00000000 00000000 00000000 00000000 00000000

      -DUMP STORAGE RANGES
      +114     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      -----

      +134     00000000 00000000 00000000 00000000 00000000 00AA1C1C 00AA1C20 00AA1C24 00AA1C28
      +154     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

      -SUBFUNCTION REGISTER SAVE AREA
      +174     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      +194     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      +1B4     00000000 00000000
      -----

      +1EC     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      +1DC     00000000 00000000 00000000 00000000 00FF77FC 00000000

      -RECURSION REGISTERS
      +1F4     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
      +214     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

(5g) XDBA -- IOS EXCP debugging area (EXCPD) pointed to by the TCB. This information is formatted only for all DEBs under the task for the IO option. No example of this information is shown in the corresponding figure.

(5h) UCB -- unit control block pointed to by the DEB. This information is only formatted for all DEBs under the task for the IO option. No example of this information is shown in the corresponding figure.

(5i) RTM2WA -- RTM work area queued to the TCB. This information is only formatted if available, for the ERR option.

Figure 5. ABDUMP Output (Example) (Part 6 of 14)

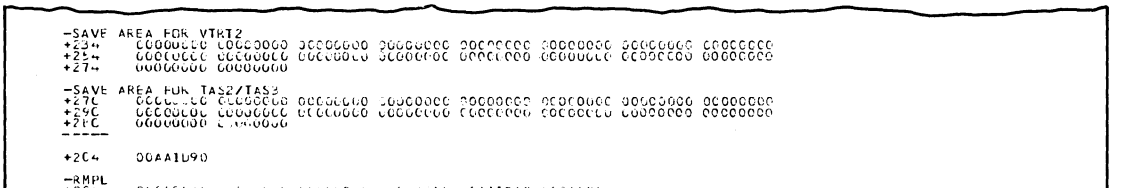


Figure 5. ABDUMP Output (Example) (Part 7 of 14)

```
                                RTM2WA SUMMARY

+10     COMPLETION CODE              00DC1000
+2C     ABENDING PROGRAM NAME
+54     ABENDING PROGRAM ADDR        00000000

+3C     REGS AT TIME OF ERROR        00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   (0-7)
+5C                                  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   (8-F)
+7C     EC PSW AT TIME OF ERROR      00000000 00000000 00000000 00000000
+8C     SDWACOMP                     00000000

+E8     RETURN CODE FROM RECOVERY ROUTINE-00,CONTINUE WITH TERMINATION-IMPLIES PERCOLATION
+EC     RETRY ADDR RETURNED FROM RECOVERY EXIT   00000000
+E4     RB ADDR FOR RETRY                        00000000

+0      CVT  ADDR                    00039048
+38     RTCT ADDR                    00FF7060
+08     SCB  ADDR                    00001001
+0A     SDWA ADDR                    00000000
+14     SVRB ADDR                    00ABE5F0
+160    PREV RTM2WA FOR THE TASK     00000000
+174    PREV RTM2WA FOR RECURSION    00000000

+BE     ASID OF ERROR IF CROSS MEMORY ABTERM        0000
+300    ERROR ASID                   0000

+370    CURRENT TRACE ENTRY FOR SAVED TRACE TABLE   00000000
+380    FIRST TRACE ENTRY FOR SAVED TRACE TABLE     00000000
+384    LAST TRACE ENTRY FOR SAVED TRACE TABLE      00000000

                                RTM2WA BIT FLAG SUMMARY

+2C8    TASK TERMINATION

                                ESA BIT FLAG SUMMARY (RTM1 RELATED INFORMATION) FROM SVRB AT 00ABE5F0

+89     RECURSION DURING INITIALIZATION
```

Figure 5. ABDUMP Output (Example) (Part 8 of 14)

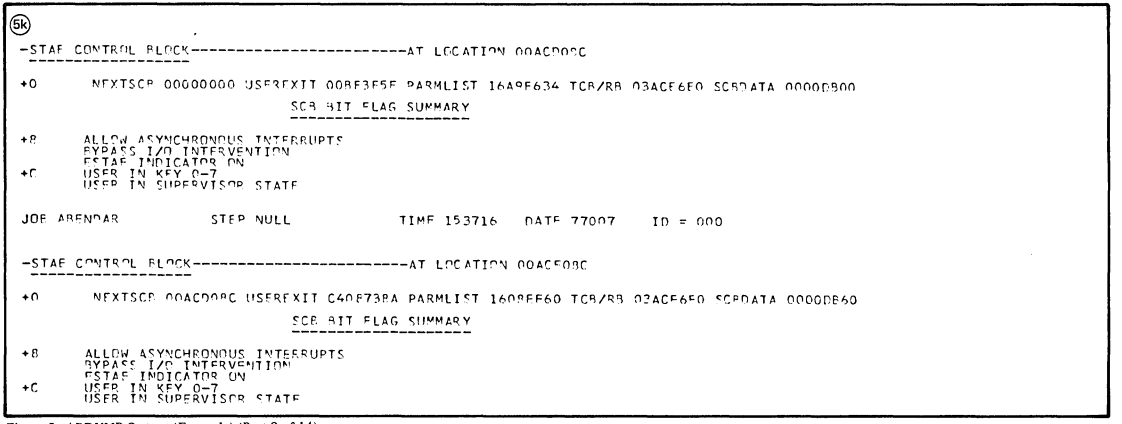


Figure 5. ABDUMP Output (Example) (Part 9 of 14)

(5j) EED -- extended error descriptor blocks queued to the TCB. This information is only formatted if available, for the ERR option. No example of this information is shown in the corresponding figure.

(5k) SCB -- STAE control blocks queued to the TCB. This information is only formatted if available, for the ERR option.

```
(6) SAVE ARLA TRACE

(6a) GO        WAS ENTERED VIA LINK

(6b) SA   084FB0  WD1 00000000   HSA 00000000   LSA 00085FB8   RET 00025DF8   EPA 00085EB8   R0  00B9D2C8
                  R1  00084FF8   R2  00000040   R3  00B92464   R4  00B92440   R5  00B9CCD0   R6  00B78C18
                  R7  FD000000   R8  00B9C9B0   R9  80B9CF88   R10 00C00C00   R11 00B9CF50   R12 70F0094A

    SA    085FB8  WD1 00000000   HSA 00084FB0   LSA 00000000   RET 00000C00   EPA 00000000   R0  00000000
                  R1  00000C00   R2  00000000   R3  00000000   R4  00000000   R5  00000000   R6  00000000
                  R7  00000000   R8  00000000   R9  00000000   R10 00000000   R11 00000000   R12 0000000C

(6d) INTERRUPT AT 085EFE

(7) PROCEEDING BACK VIA REG 13

    SA    085FB8  WD1 00000000   HSA 00084FBC   LSA 00000000   RET 00000000   EPA 00000000   R0  00000000
                  R1  00000000   R2  00000000   R3  00C00000   R4  00C00000   R5  00000C00   R6  00000000
                  R7  00000000   R8  00000000   R9  00000000   R10 00000000   R11 00000000   R12 00000003

    GO        WAS ENTERED VIA LINK

    SA    084FB0  WD1 00000000   HSA 00000000   LSA 00085FB8   RET 00025DF8   EPA 00085EB8   R0  00B9D2C8
                  R1  00084FF8   R2  00000040   R3  00B92464   R4  00B92440   R5  00B9CCD0   R6  00B78C18
                  R7  FD000000   R8  00B9C9B0   R9  80B9CF88   R10 00000C00   R11 00B9CF50   R12 70F0094A
    ADDRESS FOR ATCVT       NOT AVAILABLE

    VTAM NOT ACTIVE FOR THIS CALL
```

Figure 5. ABDUMP Output (Example) (Part 10 of 14)

(6) SAVE AREA TRACE -- Save area trace and linkage information requested by the PDATA=SA or PDATA=SAH option. (All current save areas for the task are printed in the order in which they were created. The first save area is created by the control program, and queued to the TCBFAS field of the TCB; it is used by the module at the highest level of control within the task. Additional save areas, if any, are created by the task and used by modules at successively lower levels of control.)

(6a) Linkage information (for each linkage that creates a request block).

cccccccc WAS ENTERED -- Name of the module that received control and stored registers in the next-printed save area.

VIA LINK or CALL ddddd -- Identifies the linkage macro instruction (LINK or CALL), and its ID operand (ddddd), if coded.

AT EP cccc... -- Module entry point identifier (optional operand of the SAVE macro instruction), if coded.

(6b) SA hhhhhh -- Save area address, followed by formatted fields.

WD1 hhhhhhhh -- First word of the save area (used by PL/I programs).

HSA hhhhhhhh -- Second word of the save area; address of the previous save area, or zero (for the first save area).

LSA hhhhhhhh -- Third word of the save area; address of the next save area, or zero (for the last save area).

RET hhhhhhhh -- Fourth word of the save area; contents of general register 14 (return address).

EPA hhhhhhhh -- Fifth word of the save area; contents of general register 15 (entry-point address).

R0 hhhhhhhh R1 hhhhhhhh ... R12 hhhhhhhh -- Words 6 through 18 of the save area; contents of general registers 0 through 12.

(6c) INCORRECT BACK CHAIN -- Indicates that the next-printed save area may be invalid because the second word does not point to the previous save area. No example of this information is shown in the corresponding figure.

(6d) INTERRUPT AT hhhhhh -- Address of the next instruction to be executed in the last module that was entered (obtained from the resume PSW in the last PRB on the active RB queue). This marks the end of the forward save area trace.

(7) PROCEEDING BACK VIA REG 13 -- Save area back trace requested by the PDATA=SA option. The next two save areas are for the lowest level module of the task (pointed to by general register 13) and the next higher level module. These two save areas are printed only if register 13 is nonzero and points to a fullword in storage.

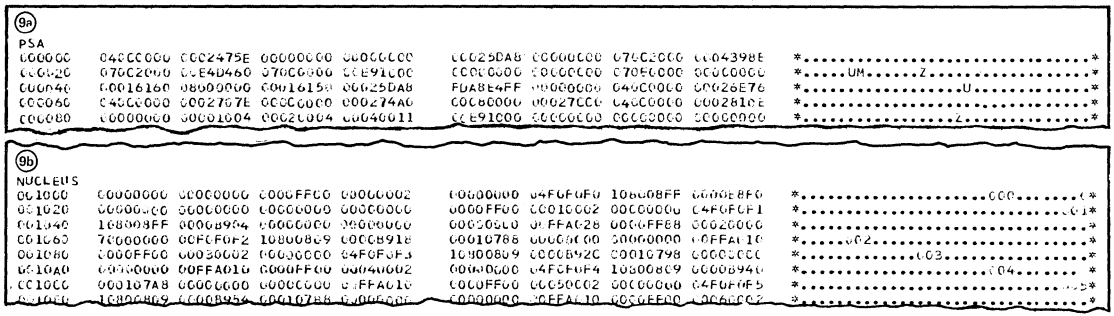


Figure 5. ABDUMP Output (Example) (Part 11 of 14)

(8) (TCAM Control Blocks) -- Formatted control blocks included in a TCAM dump, but not included in this example. (Most of these control blocks are described in Volume 2 and Volume 3; for additional information, refer to the OS/VS TCAM Users Guide, which describes the TCAM portion of the dump in full detail.)

(9) Information requested by the SDATA=NUC option.

(9a) PSA nhhhhh -- Prefixed storage area (PSA) address, followed by formatted fields. For the contents of each field, refer to the PSA description in Volume 3.

(9b) NUCLEUS -- Contents of the nucleus.[1]

[1]Each line of print includes:

- The virtual address of the first printed word of storage.
- Eight words of storage in hexadecimal format.
- The same eight words in EBCDIC format, with periods representing characters other than blank, A-Z, and 0-9.

```
(10)
ALLOCATED LSQA
C0C000    E2E2D6C2 00140005 00C0D118 00000000    00000000 00000018 00C0C000 00000000   *SSOB.....J.....................*
C0C020    00C0C030 000251B0 00000000 80C0C038    00080000 0000ED00 00100003 00000000   *.....................8.....06.*
C0C040    00000000 00000000 00000000 00000000    00000000 00000000 00000000 00BFE008   *.....................8.....06.*
C0C060    46000000 00020001 00000000 00000000    00110083 00FEB8F8 070C1000 00F0F680   *.....................8.....06.*

(15)
ACTIVE LOAD MODULES

LPA/JPA MODULE (15a)ABENDAR
085F60                                                 90ECD00C 05C050D0 C04641E0 C04250E0   *                  ..............*
085F80    D00818DE 41100076 410000A0 89000018    16104100 C0220A0D 00500000 00000000   *.....................8.....06.*
085FA0    00000000 00000000 00000000 58D0D004    98ECD00C 07FE0000 00000000 00084FB0   *.....................8.....06.*
085FC0    00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000   *.....................8.....06.*
      LINE 085FE0 SAME AS ABOVE

LPA/JPA MODULE    AMDPRFMT
CB0CF0                                                                 90ECD00C 05B047F0   *                  .......0*
CB0D00    B01AC1D4 C4D7D9C6 D4E3D9F0 F2F0F0F0    F0F861F1 F861F7F3 18C11BFF 12804780   *..AMDPRFMTR0200008.18.73.A.....*
CB0D20    B2A45860 C0089180 C0074780 B04C4100    0068BF08 C0064510 B03C0A0A 50D10004   *.....................8.....06.*
CB0D40    501D0008 18D18200 D0601BAA 91FF8000    4780B272 91018000 4710806F 91048000   *.....J.....................*
CB0D60    4710B0C4 418A8004 47F0B04C 43308001    5430B2AE 88300004 91028000 4780B08E   *...D.....0.....................*
CB0D80    58708004 41AA0004 47F0B092 5870C02C    48473001 5040C02C 1B444340 80024144   *.........0.....  .....  ....*
CB0DA0    60004110 60781941 47B0B05E 41943001    899147D0 B0BC1B91 1B394430 B26047F0   *.....................8.....06.*
CB0DC0    B05E4330 80015430 B2B24133 00019108    80004708 B0E4587A 800441AA 000447F0   *.....................8.....06.*
CB0DF0    B0F85870 C0304147 30005040 C0301B44    43408003 9180C007 4780B1BC 91108000   *.Y.....  .....  .....*
CB0E00    4710B18C 91208000 4120D048 4780B12E    41246000 41106078 182147B0 B0664192   *.....................8.....06.*
CB0E20    30001991 47D0B12E 18911B39 9023D058    18071B1C 58F0C014 05EF12FF 4780B14A   *.....................8.....06.*
CB0E40    9680D060 47F0B066 18101357 5450B2B6    4770B15A 41500004 1B3547D0 B1701A75   *.....0.....................*
CB0E60    06504450 825A4125 200147F0 B1321A35    4780B17C 06304430 B25A9823 D0589120   *...........0...................*
```

Figure 5. ABDUMP Output (Example) (Part 12 of 14)

(10) ALLOCATED LSQA -- Dump of the local system queue area (LSQA) and subpools 229 and 230, requested by the SDATA=LSQA option.[1] (There is no special heading for subpools 229 and 230; when both LSQA and SQA are dumped, the subpools are printed before SQA.)

(11) ALLOCATED SQA -- Dump of the system queue area (SQA) requested by the SDATA=SQA option.[1] No example of this information is shown in the corresponding figure.

(12) Dump of scheduler work area (SWA) requested by the SDATA=SWA option. No example of this information is shown in the corresponding figure.

SP ddd -- Virtual storage subpool number (236 or 237) and dump.[1]

(13) REGS AT ENTRY TO ABEND or SNAP -- Register data requested by the PDATA=REGS option (data obtained in registers on entry to the ABEND or SNAP routine). In the case of a program error, the data may or may not be the same as when the error occurred. No example of this information is shown in the corresponding figure.

FLTR 0-6 -- Floating-point registers 0, 2, 4, and 6 (contents in hexadecimal format).

REGS 0-7, REGS 8-15 -- General registers 0 to 15 (contents in hexadecimal format).

(14) STORAGE -- Dump of virtual storage requested by the LIST= or STORAGE= option.[1] No example of this information is shown in the corresponding figure.

(15) Dump of the job pack area requested by the PDATA=JPA option, or dump of the link pack area requested by the PDATA=LPA option, or both (requested by the PDATA=ALLPA option).
When both pack areas are dumped, the job pack area is dumped first.

(15a) ACTIVE LOAD MODULES cccccccc -- Load module on RB queue or load list (name followed by dump).[1] The load module is resident in the Link Pack Area (LPA) or the Job Pack Area (JPA).

LPA/JPA ACTIVE SVC MODULE cccccccc -- Dump of the SVC routine represented by SVRB on the active RB queue and resident in the link pack area (SVC module name followed by dump.[1])

---

[1]Each line of print includes:

- The virtual address of the first printed word of storage.
- Eight words of storage in hexadecimal format.
- The same eight words in EBCDIC format, with periods representing characters other than blank, A - Z, and 0 - 9.

(16a) TRACE TABLE

```
  SVC  OLD PSW   070C200A 00F86298  R15/R0 03000000 E600004C  R1  00B50F48  IDS 50000001  TCB 00B557B8  TME A053C7F0
 *RET  NEW PSW   070C800A 00F86298  R15/R0 00000000 00B50FB8  R1  00B50F48  IDS 50000301  TCB 00B557B8  TME A053F250
  SVC  OLD PSW   070C200A 00F865CA  R15/R0 00000000 E6000048  R1  00B50FB8  IDS 50000001  TCB 00B557B8  TME A0541E80
 *RET  NEW PSW   070C800A 00F865CA  R15/R0 00000000 00B51000  R1  00B50FB8  IDS 50000001  TCB 00B557B8  TME A0568F80
  SVC  OLD PSW   070C203C 00F865D0  R15/R0 00000000 00000094  R1  00B50EB8  IDS 50000001  TCB 00B557B8  TME A056F180
 *RET  NEW PSW   070C803C 00F865D0  R15/R0 00000000 00000084  R1  00B50EB8  IDS 50000001  TCB 00B557B8  TME A0570F80
  SVC  OLD PSW   070C200A 00F865E0  R15/R0 00000000 E6000070  R1  00B53F90  IDS 50000001  TCB 00B557B8  TME A0573C70
 *RET  NEW PSW   070C800A 00F865E0  R15/R0 00000000 00B54000  R1  00B53F90  IDS 50000001  TCB 00B557B8  TME A0581440
  EXT  OLD PSW   070C1004 00F865E0  R15/R0 00000000 00B54000  R1  00B53F90  IDS 10000000  TCB 00014030  TME A0582FA0
  DSP  NEW PSW   070C700A 00F865E0  R15/R0 00000000 00B54000  R1  00B53F90  IDS 50000001  TCB 00B557B8  TME A058C7F0
  SVC  OLD PSW   070C207B 00028CCA  R15/R0 00028CB4 000003CC  R1  00028D14  IDS 40000001  TCB 00B557B8  TME A058FF80
  SVC  OLD PSW   070C204F 0004569C  R15/R0 000133D0 C0000300  R1  00000000  IDS 40000001  TCB 00B557B8  TME A0593260
 *RET  NEW PSW   070C804F 0004569C  R15/R0 00000000 C0000040  R1  00000000  IDS 40000001  TCB 00B557B8  TME A0597B70
  SVC  OLD PSW   070C204F 000457C8  R15/R0 000133D0 0000000D  R1  FFFFFFF3  IDS 50000001  TCB 00B557B8  TME A05A99E0
 *RET  NEW PSW   070C804F 000457C8  R15/R0 00000000 C0000040  R1  FFFFFFF3  IDS 50000001  TCB 00B557B8  TME A05AE250
 *RET  NEW PSW   070C807B 00028CCA  R15/R0 00000000 C0000040  R1  FFFFFFF3  IDS 40000001  TCB 00B557B8  TME A05AF790
```

```
SP 252
(17)
 078800   05008000 070C1000 0004810E 00000000   80000000 80500000 00C0BDD0 00C0BDF4  *.............................4*
 078820   00C0BDF8 40047290 00000040 0000000A   00000000 00000000 00000000 00000000  *...8 ..... ....................*
 078840   00000C5C 00022098 80047296 00C0BCB8   00000000 00000000 00000000 00000000  *.............................*
 078860   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
        LINES 078880-078FE0 SAME AS ABOVE
SP 000

                                                                                     *..  .... DSP  NEW PSW    070C700*
 0845C0   065D4040 007D0000 40C4E2D7 4040D5C5   E640D7E2 E6404040 40F0F7F0 C3F7F0F0  *4 40E5B2E6   R15.R0 00000000 000*
 0845E0   F440F4F0 C5F5C2F2 C5F64040 40D9F1F5   61D9F040 F0F0F0F0 F0F0F0F0 40F0F0F0  *00000   R1   00BDA664    IDS 00410*
 084600   F0F0F0F0 F0404040 D9F14040 F0F0C2C4   C1F6F6F4 404040C9 C4E240F0 F0F4F1F0  *004   TCB 00BDAEA8    TME 41D083A*
 084620   F0F0F440 4040E3C3 C240F0F0 C2C4C1C5   C1F84040 40E3D4C5 40F4F1C4 F0F8F3C1  *7.... SVC  OLD PSW    070C2004 4*
 084640   F7007D00 0040E2E5 C34040D6 D3C440D7   E2E64040 4040F0F7 F0C3F2F0 F0F440F4  *0E5B352   R15.R0 00000660 000000*
 084660   F0C5F5C2 F3F5F240 4040D9F1 F561D9F0   40F0F0F0 F0F0F6F6 F040F0F0 F0F0F0F0  *00   R1   00BDA664    IDS 00410004*
 084680   F0F04040 40D9F140 40F0F0C2 C4C1F6F6   F4404040 C9C4E240 F0F0F4F1 F0F0F0F4  *   TCB 00BDAEA8    TME 41D083B1..*
 0846A0   404040E3 C3C240F0 F0C2C4C1 C5C1F840   4040E3D4 C540F4F1 C4F0F8F3 C2F1007D
```

Figure 5. ABDUMP Output (Example) (Part 13 of 14)

(16) Trace table information requested by the SDATA=TRT option.

(16a) TRACE TABLE -- Formatted dump of the system trace table (included only when the system trace is active and GTF is inactive). Each line of print represents one entry in the table; the first field indicates the entry type, as follows:

DSP -- Trace dispatch
EXT -- External interruption
I/O -- Input/output interruption
PGM -- Program interruption
SIO -- Start input/output
ISD -- Initial SRB dispatch
SSR -- Suspended SRB redispatch
SVC -- SVC interruption
RET -- SVC return

The entry type is followed by formatted fields, which are described under "TTE" in Volume 3.

(16b) GTF TRACE TABLE -- Formatted dump of GTF trace records (included only when GTF is active; not included when the dump results from abnormal termination of GTF). For descriptions of these records, refer to "GTF Trace Records" later in this section. No example of this information is shown in the corresponding figure.

(17) Dump of virtual storage subpools requested by the PDATA=SPLS option. Subpools 0 thru 127 will be dumped first, followed by subpool 252, SWA (supervisor work area).

SP ddd -- Virtual storage subpool number (0-127, 252) and dump.[1]

[1]Each line of print includes:

- The virtual address of the first printed word of storage.
- Eight words of storage in hexadecimal format.
- The same eight words in EBCDIC format, with periods representing characters other than blank, A - Z, and 0 - 9.

If the dump is directed to a 3800 printer and the JCL specifies CHARS=DUMP on the SYSUDUMP, SYSABEND or SNAP-related DD statement, each line of print includes:

- The virtual address of the first printed word of storage.
- Sixteen words of storage in hexadecimal format, printed in groups of four words with the first character of each word underlined.
- The same 16 words in EBCDIC format, with periods representing characters other than blank, A-Z, and 0-9. This will appear as two 32-character groups separated by one blank.

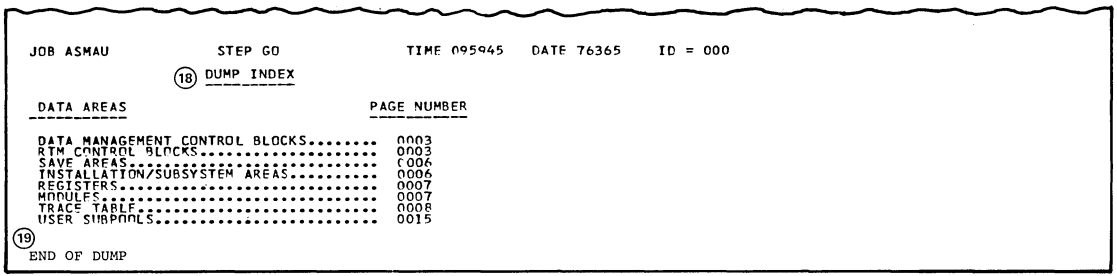


Figure 5. ABDUMP Output (Example) (Part 14 of 14)

(18) INDEX OF DUMP -- An index will be printed at the end of the dump of each task. The index will contain the page numbers for key areas of the dump.

(19) END OF DUMP -- Indicates normal completion of ABDUMP processing. If this line is missing, the ABDUMP routine was abnormally terminated before the dump was completed.

```
VSAM JSCBSHR BCC39C CCCCC4 (1)
B0C380                                                  00CBBE4E                                  *                ....*
(2)
VSAM GSR DEB B3A748 00C058
(3)
ACB 08C11C 00CC4C

08D100                                                                                A00CC04C    *                            ....*
08D120     00BA1EBE 0CEFC730 52CC0101 CCC200C1    20CCCCCC CCC00008 C0000000 00000000    *......G.........................*
08D140     0C0A749C 007C0011 00F3A748 12C00000    00CCCCC0 COC00000 C00000C0 000C0CCC    *................................*
08D160     CCCCCCCC CCC00C00                                                              *........                        *
(4)
AMBL BA1BB8 CCCC44

BA1BA0                                                               C00B86A8 00000000    *                        ........*
BA1BC0     0008D11C 00000000 00BAB808 000131C2    00CCCCCC 0C021000 50A04820 C0000CC1    *..J.............E...............*
BA1BE0     CCCCCCCC CCCC0000 00CC0000 00B59670    0CCCC0CC 00EAB440 00BA1B80              *........................       *
(5)
CMB BA1B8C 0CCC3E

BA1B80     11000038 8001000C 00B57A64 00E57A50    00CCCCCC CCC00C00 CC0000CC C00C0CC0    *................................*
BA1BA0     CCCCCCCC CCC00000 00B57A8C 00B57A7E    CCCCC0C0 00000000                      *........................        *

VSAM GSR DEB B55F9C CCCC88
(6)
AMB B5967C 000C78

B59660                                                  40CCCC78 CCC00C00 C0B53BCC C0B5989C    *                ................*
B59680     00BAB8C8 0085D280 0CCCC0CC8 00C00000    BAE8C800 01310000 C07C0211 00B55F9C    *......K.........................*
B596A0     1300000CC CCBA1B4E 00B5D26E C0CBC4C0    0CE5D31C CCC00C00 CCC000C0 00BA1BF8    *...........K...D...L............*
B596C0     00B5989C 20000000 00000000 00EA1E90    00E5ECF8 CC000C00 CCC00000 0080012C    *................................*
B596E0     C188C1CC CCCCCCC0                                                              *........                        *
(7)
BIB BABA4C CCCCEE

BABA40     1000CC68 C0C00000 00B533C0 CCEABA10    C0CCCC00 00BA1B88 C0B36A68 00B36A68    *................................*
BABA60     CCB36A7C CCB36B8C CCCC0C00 00CC0000    00CC0000 00B3CA28 00BA1DF0 00EEC73C    *.............................O..G.*
BABA80     0C000CC0 00FC4820 CCCCC0CC C1D6C4E5    C8F2F3F1 E2E3C5D7 F0F84040 C5E2C4E2    *..............ACDVH231STEPCE..ESDS*
BABAA0     FCF1404C 00FEB140                                                              *C1......                        *
```

Figure 6. VSAM SNAP Output (Example) (Part 1 of 6)

Figure 6 is an example of VSAM SNAP output. Contents are:

(1) VSAM JSCBSHR hhhhhh hhhhhh -- Location and length of JSCBSHR. This field is used by VSAM to locate the VAT. The information is the actual data in the JSCBSHR field of the JSCB.

(2) VSAM GSR DEB hhhhhh hhhhhh -- Location and length of a VSAM GSR DEB. The DEB chain is used to locate open VSAM GSR DEBS. The DEB is displayed as shown in Figure 5 number (5d). VSAM DEBs which are not GSR will not be identified.

(3) ACB hhhhhh hhhhhh -- Location and length of an ACB associated with the VSAM GSR DEB. This ACB-identifies a VSAM data set opened with the GSR option. The information is the actual data in the ACB.

(4) AMBL hhhhhh hhhhhh -- Location and length of the AMBL associated with the ACB. The information is the actual data in the AMBL.

(5) CMB hhhhhh hhhhhh -- Location and length of a CMB. This may or may not be present in every dump. The CMB contains pointers to HEB entries in the PSB which describes storage obtained when the VSAM data set was opened. The information is the actual data in the CMB associated with the AMBL.

(6) AMB hhhhhh hhhhhh -- Location and length of an AMB. The information is the actual data in the AMB.

(7) BIB hhhhhh hhhhhh -- Location and length of a BIB. The information is the actual data in the BIB.

Figure 6. VSAM SNAP Output (Example) (Part 2 of 6)

(8) VMT hhhhhh hhhhhh -- Location and length of a VMT. The VMT describes DASD volumes mounted by VSAM. The information is the actual data in the VMT.

(9) PSB hhhhhh hhhhhh -- Location and length of a PSB. The PSB contains HEBs which have entries that explicitly describe storage obtained for VSAM data set control blocks. The information is the actual data in the PSB.

(10) PSB OWNED STORAGE hhhhhh hhhhhh -- Location and length of storage owned by the PSB. This display is obtained from the HEB entry in an HEB which resides in the preceding PSB. The information is the actual data in the PSB storage.

(11) PSAB hhhhhh hhhhhh -- Location and length of a PSAB. The PSAB contains dummy AMBLs for any VSAM alternate index control block structures. The information is the actual data in the PSAB storage.

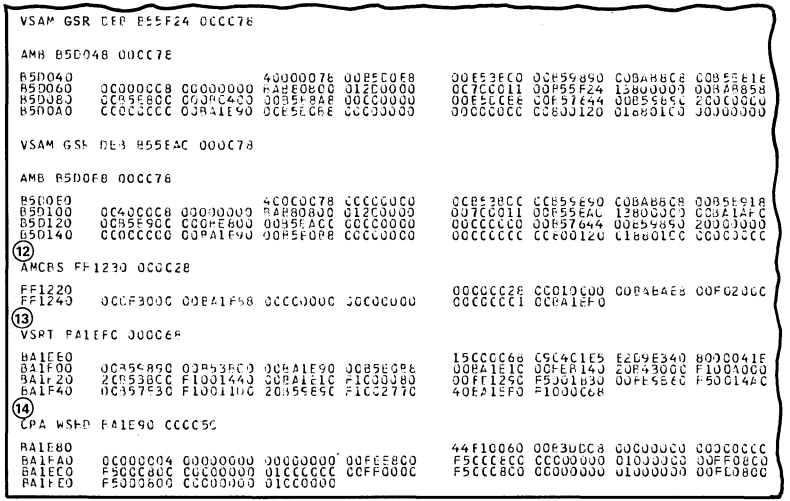


Figure 6. VSAM SNAP Output (Example) (Part 3 of 6)

(12) AMCBS hhhhhh hhhhhh -- Location and length of the AMCBS. The information is the actual data in the AMCBS.

(13) VSRT hhhhhh hhhhhh -- Location and length of the GSR VSRT. The information is the actual data in the VSRT.

(14) CPA WSHD hhhhhh hhhhhh -- Location and length of a CPA WSHD. The CPA WSHD is a VSAM WSHD used to describe storage obtained for shared resource channel program areas. The information is the actual data in the CPA WSHD.

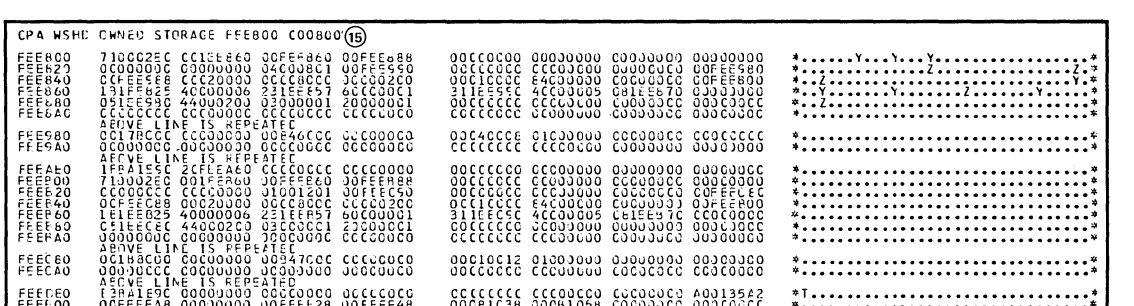


Figure 6. VSAM SNAP Output (Example) (Part 4 of 6)

(15) CPA WSHD OWNED STORAGE hhhhhh hhhhhh -- Location and length of storage owned by the CPA WSHD. This data display is of the actual channel program areas. The information is the actual data in the CPA WSHD storage.

```
(16)
WSHD B5E0B8 000C5C

B5E0A0                                                                44F10060 00000000   *                        .1......*
B5E0C0   00000000 00C00000 00000004 C0CCC0C0   00B5ECE8 20F34000 F1000600 00000000   *...................Y....1.......*
B5E0E0   01000000 20B33800 F1000800 00B5E0E8   01C00000 20B33000 F1000800 00B5E0C8   *........1...............1......H*
B5E100   01000000 20B328CC F1CC08C0 C0B5E0C8   01CCC000                              *........1......Q....            *

WSHD OWNED STORAGE B338CC CCC8C0 (17)

B33800   0C040000 04000000 50000006 00000006   00CC00C0 0CC00C00 00000000 C00CC000   *................................*
B33820   0CCCCCCC CCC0C0C0 03CC0000 000000C0   00CCC0C0 0C000000 0000000C 00000000   *................................*
         ABOVE LINE IS REPEATED
B338E0   CCCCCCCC CCC00000 0CCC0000 0CC00000   00C00000 0CC00C00 C00000CC 7FB5ECC8   *...............................H*
B33900   00040C00 C4000001 0C0000C6 00000006   00CC0000 00000000 C000000C 00000000   *................................*
B33920   00000000 00000000 00000000 00C00000   00CCC0CC CCC00CC0 C0C000C0 CC0C00CC   *................................*
         ABOVE LINE IS REPEATED
B339E0   0C000000 00000000 0CC00C00 00CC00C0   0CCCCCCC CCCC0C00 C0000000 BFB5E0C8   *...............................H*
B33A00   0C040000 04000000 10000006 00000006   00C0C000 00000000 00000000 C00C0C00   *................................*
B33A20   0C0000CC C0000000 0CCC000C CCCC00C0   00CCC0C0 00000000 00000000 00000000   *................................*
         ABOVE LINE IS REPEATED
B33AE0   CCCCCCCC CCCC0CCC 0CCC0000 00CC0000   00CC0000 00000000 00000000 CFB5E0C8   *...............................H*
B33B00   0C040000 04000000 5CCC0006 C0CC00C6   0CCCCCCC CCC00000 C0000000 000C0000   *................................*
B337E0   0C0000CC 00C00000 00000000 00CC00C0   00CCCCC0 CCC00000 00000000 FEB5ECD8   *...............................Q*
(18)
VSRT OWNED STORAGE B43000 00A000

B43000   0C0000CC 00000000 0CCC0000 CCCC00C0   CCCCCCCC CCC00000 C00000CC 00000000   *................................*
B43800   0C040001 04000000 140000C7 90000400   01C400C0 0C150000 C6000004 00010400   *................................*
B43820   0C00160C 00C86400 04CC01C4 CCCCC017   00CCC8C8 0C040001 C4000000 18000009   *..................H.............*
B43840   2C000400 01C40000 00190000 05900004   00C104CC 00001A00 C0CA0CCC 040C0104   *................................*
B43860   0C0CC01B CCC00A64 00C40001 04000000   1CC0000A C8000400 01040000 001C0000   *....................H...........*
B43880   0B2CC004 C0010400 000C1E0C CCCB9CC0   04CCC1C4 CC0001F C0000CC0 00040001   *................................*
B438A0   04000000 2000000C 64000400 01040000   00210000 0CC80004 00010400 000C220C   *.....................H..........*
B438C0   0C0D2C00 C40001C4 CCCC0C23 0CCCCD50   00C40001 04000000 2400000E 00000400   *................................*
B438E0   01040000 00250000 0E640004 00C10400   00C026C0 0C0EC800 CC000CC0 C00C000C   *......................H.........*
B43900   0C000CCC CCC00000 00CC0000 0C000000   00CCCC00 0C000000 00000000 00000000   *................................*
         ABOVE LINE IS REPEATED
B439E0   0C0000CC 00C00000 00000000 C0000000   00CCCCCC 00000800 134000CD C0F700FF   *............................7..*
B43A00   0C040001 C4000001 4C0000C7 90C00400   01C4C000 01500000 08000004 00010400   *................................*
```

Figure 6. VSAM SNAP Output (Example) (Part 5 of 6)

(16) WSHD hhhhhh hhhhhh -- Location and length of a WSHD. The information is the actual data in the WSHD.

(17) WSHD OWNED STORAGE hhhhhh hhhhhh -- Location and length of storage owned by the WSHD. The information is the actual data in the WSHD storage.

(18) VSRT OWNED STORAGE hhhhhh hhhhhh -- Location and length of storage owned by the VSRT. Imbedded in the VSRT are explicit descriptions of miscellaneous storage obtained for the VSAM shared resource pool. The information is the actual data in the VSRT storage.

```
(19)
VGTT BA1CFC 00C020

BA1DE0                                       E5C7E3E3 4C0000F1 C0000020 00BABCC8   *                VGTT...1........*
BA1E00    00BABA4C 00000001 00B36A6E COCC00C0                                       *................                *

VGTT BABCC8 0CCC2C

BAB000                     E5C7E3E3 080000F1    00CCCC20 00C00C00 000BBC38 CC0000C0   *        VGTT...1................*
BAB020    00BA12C8 CGC0C000                                                         *...H....                        *

PSB BA12C8 CCC16C

BA12C0                     F1000160 00EA12C8    00CCCCCC CCC00C00 C0BA1428 1300014C   *        1......H................*
BA12E0    0CCC00CC C0000010 FA000188 0CCA7538    00CCC0C0 CC000000 000A76C0 FA000158   *................................*
BA1300    00084378 00C00000 00C00000 0CCE44C0    FCCCCC3E CC0BC2B0 C0C00000 000000C0   *........................B.......*
BA1320    000BC2EE E6CCC088 00AEB930 COC000C0    00C0C0C0 0CAEB9B8 FA0001CC C00AB768   *..BYh...........................*
BA1340    00000000 00000000 000AB928 FACC0400    00CBDCCC CCC00000 C0000000 000BD400   *..............................M.*
BA1360    FC00C630 000BE000 00000000 00C00000    00CBE63C F5C00CC8 00FFC4F8 CC0C000C   *......................W.5..H..C8*
BA1380    CCCCCCCC 00FEC5C0 0CCC0000 000000C0    00CCC000 00BA139C 00000000 00000000   *......E.........................*
BA13A0    00000000 00C00000 00BA13B0 00C000C0    0CCCCCCC CC000000 C0000000 00BA13C4   *...............................D*
BA13C0    CCC0000C 00000000 0CCC0C00 0CCCC000    00EA13D8 CC000000 C0C00000 CC0C0000   *..................Q.............*
BA13E0    00000000 00BA13EC 00C00C0C 00CC0000    00CCC0C0 00000000 00BA1400 00000000   *................................*
BA1400    000CCC00 00000000 00CC0000 00EA1414    00CCC0CC 0C000C00 C000C0C0 C00C0CCC   *................................*
BA1420    CC0CCCCC C0C00000                                                         *........                        *

END OF VSAM DATA (20)
```

Figure 6. VSAM SNAP Output (Example) (Part 6 of 6)

(19) VGTT hhhhhh hhhhhh -- Location and length of a VGTT. The information is the actual data in the VGTT.

(20) END OF VSAM DATA -- Indicates that this is the end of information supplied by the VSAM SNAP Dump Facility.

The purpose of this page is to present an example of the high-density dump format supported for SNAP and ABEND dumps printed on a 3800 printer.

The dump contents are unchanged; only the format of storage dumps lines have changed. Each line contains a 6 digit address. A maximum of 16 fullwords of storage can be formatted in the 4 four-word pieces. The full-words are followed by an EBCDIC translation (up to 64 characters in length), separated into two 32 character portions. The first digit of each fullword of storage is underlined.

```
SP 230

8F6F40                                   E60000B000000000000000000009019A0 00000000050E6C3B600E6CB78008F5B7C  008F6FDC008F6FF00000000000000904428                 W................ .....WC..W............0......
8F6F80 008F598400904428008F595880DE77E2 008F5F20008F6FF0008F6F5850E6C01C 80E6C0FA00E6C000008F5B7C008F5958  008F5B40008F5940008F598450E6CAA4 ...............S.......0.....W.. .W...W............. ... .....W
8F6FC0 00E0A9FE008F595880DE77E2008F5F20 00000000008F6F5850E6C01C008F6FF0 00000000000000000000000000000000  80800000008F59580000000000000000 ...........S............W.....0 ................................

SP230

8F7E40                                                                    0000000000000000  000000000000000004E000000000001                                   ........................
8F7E80 00004000000000001040000015400000000 00400020008F5BA492FA1DF000FAA7D0 0700CC8A00000AA4280228284l074F78  00C3C36000C3C360000000D100000001 .. .............. .........0.... ................CC..CC....J..
8F7EC0 8F8F7E6800901EB8008F7F50008F7ED8 90436800000B080000000000000000000 04000000000000000000000000000000  000000000000000018F7EFC008F7F50 ...............Q................ ................................
8F7F00 9043682950DEB2FE00E6CA8800000198 008F7EC4008F9048008 0E600009 01EB8 00000000000000000000000000000000  00000000000000000000000000000000 .........W.........D......W..... ................................
8F7F40 00000000000000000000000000000000 00000000000000004040404040404040 00000000C4E4D4D74040404040404040  40404040000000000000000000000000 ........................         ....DUMP            ..........
8F7F80 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000  00000000000000000000000000000000 ................................ ................................
        LINE 8F7FC0 SAME AS ABOVE
```

Figure 7. High Density Dump for SNAP and ABEND

```
DUMP TITLE SAMPLE LOW SPEED DUMP FROM MP SYSTEM (1)

CURRENT PSW (2)       070C0000 00ECA74C            PR       001F5000   CPU ID       00

GR 0-7      80D5BD48 00FF40EC C9C7C5F0 00024868   00CAE2F0 00CAF188 00FF4018 00FF40EC  *.N.... .IGE0.......SO..1... ... .*
GR 8-F      80D5BD48 00005D48 00ECA730 80D5BD48   00000000 00CAE530 00CAE4D0 00ECA730  *.N............N.........V...U......*

CR 0-7      C080EC40 0F082C00 FC000000 00000000   00000000 00000000 00000000 00000000  *... ..............................*
CR 8-F      00000000 00000000 00000000 00000000   00000000 00000000 EFC00000 001EE2F0  *.............................SO*

FR 0-2      00000000 00000000 00000000 00000000                                       *...............                 *
FR 4-6      00000000 00000000 00000000 00000000                                       *...............                 *

CURRENT PSW (3)       050C1000 00033CCE            PR       001F1000   CPU ID       01

GR 0-7      AD040264 00FF6330 00FFFAD8 FFFFC000   00FFD078 00033640 00FFFE94 00FFFD20  *...........Q........... ........*
GR 8-F      00FFFAD8 B00493D4 40049388 00000000   00049544 00000C5C 80049520 00000000  *...Q...M ......................*

CR 0-7      C080C000 0F1CEC00 FC000000 00000000   00000000 00000000 00000000 00000000  *..............................*
CR 8-F      00000000 00000000 00000000 00000000   00000000 00000000 EFC00000 001E9560  *..............................*

FR 0-2      00000000 00000000 00000000 00000000                                       *...............                 *
FR 4-6      00000000 00000000 00000000 00000000                                       *...............                 *


(5)      (4) STORAGE KEY      06
00000000 V 00080000 0000730C 00000000 00007FF0   0C000000 00007FB8 00000000 00000000  *.............0.............*
00000020 V 00000000 00000000 00000005 E005A932   00000000 00000000 00000000 00000000  *..............................*
00000040 V 00007FF0 0C000000 00007FE8 00024868   FFD4CAFF C0FCEDC0 040E0000 00025986  *...0.............M.............*
00000060 V 040E0000 0001492E 000E0000 00031BB0   00020000 0010ADF2 040E0000 00030566  *.....................2........*
00000080 V 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *..............................*
(6)
000000C0 V 00000000 00000000 00000000 00000000   00000000 00000000 7FFFFFF2 6015B000  *.............................2....*
000000E0 V 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...5..........................*
```

Figure 8. Low-Speed Stand-Alone Dump (Example)

SADMP Output (Low-Speed Stand-Alone Dump)

Figure 8 is an example of a low-speed stand-alone dump. Contents are as follows:

(1) DUMP TITLE cccc... -- Title supplied by the operator in response to message AMD011A.

(2) Data for the processor on which the stand-alone dump program was IPLed:

CURRENT PSW -- current program status word
PR -- prefix register
CPU ID -- processor identification
GR 0-7, GR 8-F -- general registers
CR 0-7, CR 8-F -- control registers
FR 0-2, FR 4-6 -- floating-point registers

All data is shown in hexadecimal format. On the righthand side of the page, registers are shown in EBCDIC format, with periods representing characters other than blank, A - Z, and 0 - 9.

The PSW, general registers, control registers, and floating-point registers contain data saved by the STORE STATUS operation. If this operation was not performed, the data is meaningless, except for the general registers, which contain data saved by the stand-alone dump program immediately after IPL.

(3) Data for alternate processor, corresponding to data described under (2) above. The PSW, general registers, control registers, and floating-point registers contain data saved by a STORE STATUS operation initiated by the stand-alone dump program.

(4) KEY hh -- Storage protection key for succeeding blocks of storage.

(5) Real or virtual storage specified by the operator in response to message AMD008A. Each line of print includes:

- The hexadecimal address of the first printed word of storage.
- A single letter (R or V) indicating whether the address is real absolute or virtual.[1]

[1] An absolute address is an address to which prefixing has been applied. Prefixing is described in GA22-7000; in effect, it simply interchanges the 4K bytes of storage at real address 0 with the 4K bytes of storage at the real address contained in the prefix register. Prefixing is applied to all references to real storage and to keys in storage, except for references to the PSA during a STORE STATUS operation, and except for references by a channel to extended logout locations, to indirect-data-address words, and to CCWs.

Virtual addresses correspond to the address space current at the time of the STORE STATUS operation for the processor on which the stand-alone dump program was IPLed. If the STORE STATUS operation was not performed, virtual addresses correspond to the master scheduler address space.

Note: As printed in the dump, virtual address zero corresponds to absolute address zero. The corresponding real address is not zero, but the address contained in the prefix register.

Each line of print includes:

- The virtual address of the first printed word of storage.
- Eight words of storage in hexadecimal format.
- The same eight words in EBCDIC format, with periods representing characters other than blank, A-Z, 0-9.

(6) A blank line, representing a range of storage addresses that has not been printed for one of the following reasons:

- The storage contains data identical to that contained in the preceding eight words of storage.
- The storage is paged-out or unallocated (virtual storage only).
- The storage is offline (real storage only).

```
①
SAMPLE SVC DUMP                                    MODULE RTMSD350  DATE 08/16/73  TIME 22.28.05  PAGE 0001
②
TITLE FROM DUMP: SVC DUMP TESTCASE -SVCD00
③
ERRORID FOR THIS DUMP=SEQ00001 CPU41  ASID 0001 TIME 14.59.18.3
```

Figure 9. AMDPRDMP Page Heading and Dump Title (Example)

AMDPRDMP Output (SVC Dump, SYSMDUMP ABEND DUMP, and Unformatted Stand-Alone Dump)

AMDPRDMP output consists of:

- A page heading and dump title.
- Output of AMDPRDMP control statements.
- AMDPRDMP message log.
- AMDPRDMP output comments.

Page Heading and Dump Title: Figure 9 is an example of an AMDPRDMP page heading and dump title, which appears on the first page of a dump.

① Page heading (repeated on every page).

ccc... -- Title specified when AMDPRDMP was executed (1 to 64 characters). If a title was not specified, the first 64 bytes from the "TITLE FROM DUMP" will be printed starting on page 2.

MODULE cccccccc -- Name of the module that created the dump.

DATE dd/dd/dd -- Date when the dump was taken (month/day/year).

TIME dd.dd.dd -- Time of day when the dump was taken (hour.minute.second).

PAGE dddd -- AMDPRDMP output page number. (Pages are numbered consecutively starting with 0001.)

② TITLE FROM DUMP: cccc... -- Title specified when the dump was taken (1 to 100 characters). This title appears only on page 0001. If a TITLE control statement was not entered, the first 64 bytes of this title will appear in the heading starting on page 2.

③ ERRORID -- The RTM component produces an error id for inclusion by SVC dump. The error id is also included on the software LOGREC record and console via message IEA911A. The error id in these 3 locations provides a mechanism for correlating data pertinent to this error.

```
 * * * *    A M D P R D M P   M E S S A G E   L O G    * * * *
(1)
 SUMMARY
 AMD174I SYSUT1 LOADED
 CPUDATA
 FORMAT
 P CURRENT
 LPAMAP
 QCBTRACE
 CVTMAP
 END

(2)
  1   NO OF ENTRIES TO READ ROUTINE - 0000011330
  2   NO OF TIMES REQ ADDR WAS NOT IN A BUFFER - 0000003092
  3   NO OF BLOCKS READ - 0000000125
  4   NO OF PERMANENT I/O ERRORS - 0000000000
  5   AVERAGE NO OF BUFFERS PER FUNCTION - 0030.00

 RATIO OF ITEM 1 TO ITEM 2 - 003 TO 1

  6   NUMBER OF TRACE RECORDS PROCESSED - 0000000000
```

Figure 10. AMDPRDMP Message Log (Example)

AMDPRDMP Message Log: Figure 10 is an example of an AMDPRDMP message log written on the SYSPRINT data set; the AMDPRDMP message log appears on the last page of AMDPRDMP output. Notice that this is not an error message log. Contents are as follows:

(1) List of AMDPRDMP control statements. The list indicates the order in which control statements were entered, and the order in which control statement output appears in the dump listing.

(2) AMDPRDMP output comments.

AMDPRDMP Output Comments: Output comments are unnumbered messages, which are interspersed with other AMDPRDMP output. Each message is explained in the topic "AMDPRDMP Output Comments," which follows the descriptions of control statement output.

```
(1)                                    * * * FORMAT OF ASM DATA * * *

ASMVT   000330B0 ANCHORED IN CVTASMVT
+   0 FLAG1 00          FLAG2 08          RSV4  0000        SART  00FE0190   PART  00FE10C8   GOS   00031E90
+  10 TRPAG 000316BC    EREC  00FCD700    MSGBF 00FE9458    RSV5  00000000   STAGF 00000000   STAGL 00000000
+  28 IORQR 00021F71    IORQC 00021F6F    RSV6  00000000          00000000         00000000         00000000
+  40 PCCWA 00076AA8    PCCWC FFFDE089    PCCWN 00000028    BURST 0000C350   IOCNT 00000001   SRBCT 00000001
+  58 PSRB  00FE9400    RSV7  00000000          00000000          00000000         00000000   BKSLT 00005F05
+  70 SLOTS 00008BIC    VSC   0000005B    NVSC  000005A1    ERRS  00000006   SLOTV 000334B0   LGVT  00FE9000
+  88 STGXA 00FEDB68    CINV  00001644    LSAIL 00000001    LSAIH FFFFFFFF   GOSWT 00000000   GOSWK 00000000
+  A0 REQCT 00000002    TCBPT 00AF9220    TMECB 80AF9CA0    RLGRQ 00000000   RLGWQ 00000000   TASCB 00039CB0
+  B8 RSV8  00000000          00000000    PQIOE 0001DFC6    PFRSL 0001E358   PFRSW 0001E35E   PTM   0002BED0
+  D0 PSWPD 00032E48    PSRMT 0001FB02    PSRBC 00031A68    VRMTR 00031A80   PEX   000334B8   PCMPD 800308CC
+  E8 PCMPN 00028540    PCMPA 00030A08    PCMP  000307E0    PSAV  00CC5750   PACT  00CC5100   PRLG  00CC3000
+100 PFRLG 040C0000    PMSG0 0004020E    PMSGS 00000000    PVACQ 00000000   PIOFR 0003AB78   PVACE 00000000
+118 RSVI0 070C0000          00035954          078D0000          00C5F50E

POOL CONTROLLERS
+128 CPID  IOEP    CPSIZ 0010    CPEXT  0000    CPAVL 00FEA888    CPCNT FFFDE08B
+138       BWKP          0100          0004          00FEA040          FFFFFF39
+148       SWKP          0200          0004          00000000          00000000
+158       ACEP          0028          0032          00FE9600          FFFFFFBB    CPTAK 00000000    CPRES 00FE9DA8

WORK SAVE AREAS
+170 PAGIO      00FD36E4  00FED228  000454A8  00AFC060  00AE0770  00001649  00075820  0002165A  5002065C  00FD36E4
+198            00000000  00000000  00045B30  40020D2E  00000004  00FED440  00000000  00000000  00000000  00000000
+1C0 QIOE       00FED228  000330B0  00000CDC  00FD2724  84AFF4C4  00AFF4C4  00033620  6001DD78  00000C00  00000DA9
+1E8            00000000  00000000  9001DFA8  00FEA7B8  00000000  00000000  40000000  A001DFF0  00000000  00000000
+210 PAGCM      00000000  00000000  00000000  00000000  00000000  A002D898  00000000  00FFB700  00000000  00000000
```

Figure 11. Format of ASM Data (Part 1 of 13)

ASMDATA OUTPUT

Figure 11 is an example of ASMDATA output. Contents are as follows:

(1) ASMVT hhhhhhhh anchored in CVTASMVT -- Address of the auxiliary storage management vector table (ASMVT) contained in the CVTASMVT field of the communication vector table (CVT).

```
+218                 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+260 SWAP            00FDEAC8 000454A8 00AFCEE6 000000FF 00000008 00FD3580 00FDEB10 0002F2AE 5002E2B0 00FF1598
+288                 00000000 00FD3580 4002E54A 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+2B0 SLSQA           00FDEAC8 000330B0 00FFBA78 00FDEF18 00FFB9BC 00FFDE28 6002D7F4 00000000 00FCF190 00FCF1B4
+2D8                 00000000 00000000 4002DD78 8001D5A8 00000001 00FCE674 80000000 00000000 00000000 00000000
+300 POS/VIOCM       00000088 00FD2724 00FED228 000330B0 00000CDC 00FD2724 84AFF4C4 00AFF4C4 00033620 00000000
+328                 6001DD78 00000DA9 00FED440 00000000 7001DE94 00000000 0000000A 00000000 00FF2D7C 00000CDC
+350 PEX/MSG00       00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+378                 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+3A0 STANDARD        00000000 00FEDC04 00000000 4003169E 0001E358 00000088 00030120 00FED228 000330B0
+3C4                 00000CDC 00FE0310 700311A0 00FD2724 00000000 00ABFAE0 00ABFB34 90031358 00FE9000
+3E8                 00000000 00000000 00000000 00000000 00000000
```

(2)

```
EREC    00FCD700 ANCHORED IN ASMEREC
     CURNT 00FCD720    CFRST 00FCD70C    CLAST 00FCDAFC
00FCD70C        00040317    0004039B    0003032B    00030152    000402D1    0003039C    00000000    00000000
00FCD72C        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD74C        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD76C        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD78C        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD7AC        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD7CC        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD7EC        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD80C        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD82C        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD84C        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD86C        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD88C        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD8AC        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD8CC        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
00FCD8EC        00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
```

Figure 11. Format of ASM Data (Part 2 of 13)

(2) EREC hhhhhhhh anchored in ASMEREC -- Address of the error record table (bad slots) contained in the ASMEREC field of the ASMVT.

```
00FCD90C   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCD92C   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCD94C   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCD96C   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCD98C   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCD9AC   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCD9CC   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCD9EC   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCDA0C   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCDA2C   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCDA4C   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCDA6C   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCDA8C   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCDAAC   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCDACC   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FCDAEC   00000000   00000000   00000000   00000000   00000000

(3)
MSGBF   00FE9458 ANCHORED IN ASMMSGBF
     CURNT 00FE9468    FIRST 00FE9468    LAST  00FE9508     TERM  00FE9558
00FE9468   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
           00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FE94B8   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
           00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
00FE9508   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
           00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
TERMINATION BUFFER
00FE9558   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
           00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
           00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
```

Figure 11. Format of ASM Data (Part 3.of 13)

(3) MGSBF hhhhhhhh anchored in ASMMSGBF -- Address of ASM message buffers contained in ASMMSGBF field of the ASMVT.

```
PCB/AIA 00FEE700 ANCHORED IN ASMSTAGQ (4)
PCB
     CQN   88          FQPA  FF1B98      BQPA  00FF18D8      FL1   02          RTPA  000000
     FXC   00          RLPA  000000      FL2   80            XPTA  AD87FC      FL3   00          PGTA  AD87AA
     RBN   0B10        VBN   1450        SRB   00000000      ASCB  00FE0F18
AIA
     FQPA  00FEE7A4    BQPA  11111111    OP    00            FLG1  C0          FLG2  00          FLG3  00
     NXAIA 00FF18FC    ID    0000000000000000      GRPSZ/LGE/DPXCT  00000000
PCB/AIA 00FF18D8
PCB
     CQN   88          FQPA  FEE700      BQPA  00FEE780      FL1   02          RTPA  000000
     FXC   00          RLPA  000000      FL2   80            XPTA  AD8808      FL3   00          PGTA  AD87AC
     RBN   0C70        VBN   1460        SRB   00000000      ASCB  00FE0F18
AIA
     FQPA  00000000    BQPA  55555555    OP    00            FLG1  C0          FLG2  00          FLG3  00
     NXAIA 00FF1BBC    ID    0000000000000000      GRPSZ/LGE/DPXCT  00000000
ACE      00FEE724 ANCHORED IN ASMRLGRQ
     FQPA  00FEE7A4    BQPA  11111111    OP    00            FLG1  C0          FLG2  00          FLG3  00
     LGE   00FF18FC    LGID  00000000    RPN   00000000      SYM   0000000010FEECB0  00FF1998
     SRBWK 41AE3EB8
ACE      00FEE7A4
     FQPA  00FEE9E4    BQPA  22222222    OP    00            FLG1  C0          FLG2  00          FLG3  00
     LGE   00000000    LGID  00000000    RPN   00000000      SYM   0000000010FFECB8  00FEEB00
     SRBWK 41AE3EB8
ACE      00FF1BBC ANCHORED IN ASMRLGWQ
     FQPA  00FF22FC    BQPA  66666666    OP    00            FLG1  C0          FLG2  00          FLG3  00
     LGE   00FF26FC    LGID  0033333A    RPN   0044444A      SYM   0055555A10FFECB8  00FF21D8
     SRBWK 41AE3EB8
```

Figure 11. Format of ASM Data (Part 4 of 13)

(4) PCB/AIA hhhhhhhh anchored in ASMSTAGQ --
Address of the first PCB/AIA on the staging queue contained in the ASMSTAGQ field of the ASMVT.

```
ACE     00FF22FC
     FQPA  00000000    BQPA  77777777    OP    0C          FLG1  C0          FLG2  00          FLG3  00
     LGE   00000000    LGID  00DDDDDD    SRCID 00FFFFFF    AIAPT 10FEE980    ECB   00FF1A58
     SRBWK 03000000


PART    00FE10C8 ANCHORED IN ASMPART
     TIDEN PART        TSIZE 00000006    TEUSE 00000005    TAIAE 00000000    TCIR1 00000000
     TCIR2 00FE1218    TDSNL 00C2E060    TPCTQ 00FE1298    TLCNT 0002        TLKUP 00010E0B
     TLOCK 00000000    TCOMF 00000000    TCOML 00000000    TSPLF 00000000    TSPLL 00000000
     TDUPF 00000000    TSUPL 00000000    TLOCF 00FEA7B8    TLOCL 00FEA7B8

PCT     00FE1298 ANCHORED IN PARTPCTQ
     ID    PCT         DTYPE 3330        SMAX  5B88        DTYPX 2009        CYLSZ 003A        NEXT  00000000
     DMASK 00000000000003F               DPGWT 001E        SSECN 003A        RQTIM 00001388    MAXTK 30D8
     MSSB  01B0

PCB/AIA 00FEE700 ANCHORED IN PARTAIAE
PCB
     CQN   88          FQPA  FF1B98      BQPA  00FF18D8    FL1   02          RTPA  000000
     FXC   00          RLPA  000000      FL2   80          XPTA  AD87FC      FL3   00          PGTA  AD87AA
     RBN   0B10        VBN   1450        SRB   00000000    ASCB  00FE0F18
AIA
     FQPA  00FEE7A4    BQPA  11111111    OP    00          FLG1  C0          FLG2  00          FLG3  00
     NXAIA 00FF18EC    ID    0000000000000000      GRPSZ/LGE/DPXCT  00000000

PCB/AIA 00FF18D8
PCB
     CQN   88          FQPA  FEE700      BQPA  00FEE780    FL1   02          RTPA  000000
     FXC   00          RLPA  000000      FL2   80          XPTA  AD8808      FL3   00          PGTA  AD87AC
     RBN   0C70        VBN   1460        SRB   00000000    ASCB  00FE0F18
AIA
     FQPA  00000000    BQPA  55555555    OP    00          FLG1  C0          FLG2  00          FLG3  00
     NXAIA 00FF1BBC    ID    0000000000000000      GRPSZ/LGE/DPXCT  00000000
```

Figure 11. Format of ASM Data (Part 5 of 13)

```
IOE     00FEB8C8 ANCHORED IN PARTCOMQ
     NXT   00FEBA68     LSIDA 00000000     AIA   00FEE9E4      WORK  00000000
IOE     00FEBA68
     NXT   00FEB858     LSIDA 00000000     AIA   00FF26FC      WORK  00000000


IOE     00FEA7B8 ANCHORED IN PARTSPLQ
     NXT   00000000     LSIDA 00000000     AIA   00FD2724      WORK  00000000

IOE     00FEA7B8 ANCHORED IN PARTDUPQ
     NXT   00000000     LSIDA 00000000     AIA   00FD2724      WORK  00000000

IOE     00FEA7B8 ANCHORED IN PARTLOCQ
     NXT   00000000     LSIDA 00000000     AIA   00FD2724      WORK  00000000


PARTE   00FE1118         PLPA705
     EPARE 00FE1158     ELKUP 00010E0B     ETYPE 80           EFLG1 00          ENN   0000       EIOEQ 00000000
     ESZSL 00000828     ESLTA 00000495     ERRCT 00000000     EIORB 00FEED28    EPATP 00FEEAE0   EPCTP 00FE1298
     EEDBP 00FEEE58     EUCBP 00006148     ENODE 00000000     EWTQE 00FE1100    ERQTM 00001388   ELCYL 0003

PAT     00FEEAE0 ANCHORED IN PAREPATP
     IDENT PAT          PART  00FE1118     CYLNO 0024         CYLSZ 003A        CYLMW 0002       RSV1  0000
PATMAP
     FD1C8423    0000007F    EE9FBFFF    FA338BFF    7EFFFFFF    FEFC273F    797DF6DD    EFAFF3BF
     FF7FBFD7    FFFFFFFF    FFFFFEFF    FFDFFFFF    FFFFFFFF    FFFFFFFF    FFFFFDFF    BF7FFFFF
     FFFFFFFF    FFFFFFFF    FFFFFFFF    FFFFFFFF    FFFFFFFF    FFFFFFFF    FFFFFFFF    FFFFFFFF
     FFFFFFFF    FFFFFFFF    FFFFFFFF    FFFFFFFF    FFFFFFFF    FFFFFFFF    FFFFFFFF    FFFFFFFF
     FFFFFFFF    FFFFFFFF    FFC00000    007FFFFF    00000000    0000003F    00000000    0000003F
     00000000    0000003F    00000000    0000003F    00000000    0000003F    00000000    0000003F
     00000000    0000003F    00000000    0000003F    00000000    0000003F    00000000    0000003F
     00000000    0000003F    00000000    0000003F    00000000    0000003F    00000000    0000003F
     00000000    0000003F    00000000    0000003F    00000000    0000003F    00000000    0000003F
```

Figure 11. Format of ASM Data (Part 6 of 13)

```
IORB    00FEED28 ANCHORED IN PAREIORB
     ID    88          NUM   01          RQSZ  01          FLGS  C0          IORB  00000000    PCCW  00076550
     IOSB  00FEED98    SAVE  00FEED50    ERR   00000000    TSMP  0000000000000000              PARTE 00FE1118

IOSB    00FEED98 ANCHORED IN IORIOSB
+  0 FLA   C0          FLB   81          FLC   20          PROC  00          DVRID 01          PRLVL 00
+  6 ASID  0001        PGAD  000307E0    PKEY  04          COD   7F          OPT   10          OPT2  80
+ 10 UCB   00006148    CC    40          CSWCA 0166B0      STATUS 0000       CSWRC 0000        SRB   00FEEE08
+ 20 USE   00FEED28    RES4A 00000000    APMSK 0000        SNS   0000        IPIB  00000000    PCHN  00000000
+ 34 ERP   00000000    PCI   00000000    NRM   00028540    ABN   00030A08    DIE   800308CC    RST   00076570
+ 4C VST   00076570    DSID  00000000    RSS1B 00          AFF   00          PATH  0000        FMSK  88
+ 59 CKEY  00          MDB   00          MDM   00
+ 64 SKM   00          SKBB  0000        SKCC  0004        SKHH  000C        SKR   04

PCCW    00076550 ANCHORED IN IORPCCW
     WID   86          WSECT 62          WFLGS 00          WPCCW 00000000    WAIA  00FD2BE4
     WIORB 00FEED28    WM    00          WBB   0000        WCC   0004        WHH   000C        WR    04
     SEEK  1B07656140000006    SSEC  2307655140000001    SRCH  3107656340000005    TIC   0807657840000000
     R/W   060B500040001000    NOP   03076AC020000001

PCB/AIA 00FD2BC0 ANCHORED IN PCCWAIA
PCB
     CQN   20          FQPA  000000      BQPA  00000000    FL1   40          RTPA  AFADF0
     FXC   00          RLPA  000000      FL2   10          XPTA  FE260C      EL3   00          PGTA  FE2592
     RBN   0B50        VBN   ED90        SRB   00AFD9D0    ASCB  00FD77A8
AIA
     FQPA  00000000    BQPA  00000000    OP    00          FLG1  00          FLG2  00          FLG3  00
     NXAIA 00030027    ID    000000D500000000      GRPSZ/LGE/DPXCT  00000000

SART    00FE0190 ANCHORED IN ASMSART
     ID    SART        SIZE  00000002    USE   00000001    FXDNX 00000000    MOVNX 00FE01E0    DSNL  00C2E168
     SDCT  00FEECA8    SCCWQ 000771E0    SETCT 0000004B    WAITF 00000000    WAITL 00000000    SRBP  00FE942C
     SRBCT 00000000    SETSZ 0000000C
```


Figure 11. Format of ASM Data (Part 7 of 13)

```
SDCT    00FEECA8 ANCHORED IN SARSDCT
     ID    SDCT          SIZE  0006        RSV1  0000
     DEVTP 3340          DTYPX 200A        SLTRK 0002        CYLSZ  0018        CMASK 3F3F3F3F    SSECT 02
     DEVTP 3330          DTYPX 2009        SLTRK 0003        CYLSZ  0039        CMASK 0F0F0F0F    SSECT 02
     DEVTP 3330-1        DTYPX 200D        SLTRK 0003        CYLSZ  0039        CMASK 0F0F0F0F    SSECT 02
     DEVTP 3350          DTYPX 200B        SLTRK 0004        CYLSZ  0078        CMASK 003F003F    SSECT 02
     DEVTP 2305-2        DTYPX 2007        SLTRK 0003        CYLSZ  0018        CMASK 3F3F3F3F    SSECT 03
     DEVTP 2305-1        DTYPX 2006        SLTRK 0003        CYLSZ  0018        CMASK 3F3F3F3F    SSECT 03

PCB/AIA 00FEE700 ANCHORED IN SARWAITQ
PCB
     CQN   88            FQPA  FF1B98      BQPA  00FF18D8    FL1    02          RTPA  0000
     FXC   00            RLPA  000000      FL2   80          XPTA   AD87FC      FL3   00          PGTA  AD87AA
     RBN   0B10          VBN   1450        SRB   00000000    ASCB   00FE0F18
AIA
     FQPA  00FEE7A4      BQPA  11111111    QP    00          FLG1   C0          FLG2  00          FLG3  00
     NXAIA 00FF18FC      ID    0000000000000000      GRPSZ/LGE/DPXCT  00000000


SARTE   00FE01E0        SYS1.SWAP01
     NEXT  00FE01E0      LOCK  00000000    FLG   00          NN     0000        SCCW  00000000    TOTSL 00000064
     AVLSL 0000004B      RRCNT 00000000    IORB  00FE1400    SAT    00FE2000    SDCTE 00FEECC4    EDB   00FE1640
     UCB   00008AE0

SAT     00FE2000 ANCHORED IN SRESAT
     ID    SAT           SARTE 00FE01E0    MAPLN 0019        BYTCL  0000        OFFST 000C        MASK  40
     SCAN  00FE202C      SLTNO 000002B8    ASGN  00000000
SATMAP
     8F0F0F1F    0F0FCF8F    2FBFEE2F    EF8F0F0F    2F0F0F0F    0F0FEFCF    CF0F0F0F

SCCW    00FE92C0 ANCHORED IN SRESCCW
     WID   00            WSECT 00          WFLAG 00          WSCCW  00000000    WAIA  00000000    WIORB 00000000
```

Figure 11. Format of ASM Data (Part 8 of 13)

```
     SARG  000000      0000000000     0000000000     0000000000     0000000000     0000000000     0000000000
                       00E1000000     0000000000     0000000000     0000000000     0000000000     0000000000     00
     LCCW  00000000              SEEK  0000000000000000     SSEC  0000000000000000
     SRCH  0000000000000000      TIC   0000000000000000     R/W   0000000000000000
     SRCH  0000000000000000      TIC   0000000000000000     R/W   00000000000000C8
     SRCH  0D001B80000B5FB8      TIC   00FE0AD050D6750E     R/W   00D68A2831000000
     SRCH  00FE93FC00FE93A8      TIC   0000FFFF00013848     R/W   00FE93FC00C29018
     SRCH  00AF9E8440D67414      TIC   00FE93580OFE93D8     R/W   40F109D800D679A0
     SRCH  00FE93A808000000      TIC   0110000000D67956     R/W   00D678CE00D67526
     SRCH  F50000E8000000E4      TIC   40D6741400FE9358     R/W   00FE93D800FF9358
     SRCH  0000000000000000      TIC   0000000000000000     R/W   0000000000000000
     SRCH  0001000000000000      TIC   0000000000000000     R/W   00000000F0F0F0F3
     SRCH  F0F0F0F000000000      TIC   00AF998000000000     R/W   0000000000000000
     SRCH  0000000000000000      TIC   0000000000000000     R/W   0000000000000000
     SRCH  0000060000000000      TIC   80D6750000000000     R/W   D6E4E7C200180D13
     LTIC  0006ABE300000000

IORB    00FE1400 ANCHORED IN SREIORB
     ID    88          NUM   02          RQSZ  00          FLGS  68          IORB  00FE1510    SCCW  00000000
     IOSB  00FE1470    SAVE  00FE1428    ERR   00000000    TSMP  0000000000000000              PARTE 00FE01E0



IOSB    00FE1470 ANCHORED IN IORIOSB
+  0 FLA   40          FLB   80          FLC   20          PROC  00          DVRID 01          PRLVL 00
+  6 ASID  0001        PGAD  000307E0    PKEY  04          COD   7F          OPT   10          OPT2  80
+ 10 UCB   00008AE0    CC    40          CSWCA 0772C0      STATUS 0C00       CSWRC 0000        SRB   00FE14E0
+ 20 USE   00FE1400    RES4A 00000000    APMSK 0000        SNS   0000        IPIB  00000000    PCHN  00000000
+ 34 ERP   00000000    PCI   00000000    NRM   00028540    ABN   00030A08    DIE   800308CC    RST   00077240
+ 4C VST   00077240    DSID  00000000    RSS1B 00          AFF   00          PATH  0000        FMSK  88
+ 59 CKEY  00          MDB   00          MDM   00
+ 64 SKM   00          SKBB  0000        SKCC  0153        SKHH  0008        SKR   01
```

Figure 11. Format of ASM Data (Part 9 of 13)

```
SYSTEM COMMON AREA PAGE AND EXTERNAL PAGE TABLES

PGT/XPT 00FE60D8 FOR
    VSA 00B00000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000

PGT/XPT 00FE61B8 FOR
    VSA 00B10000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
     PGTE   0008        XPTE  00000000    00000000    00000000
```

Figure 11. Format of ASM Data (Part 10 of 13)

```
     PGTE  0008        XPTE  00000000    00000000    00000000
     PGTE  0008        XPTE  00000000    00000000    00000000
     PGTE  0008        XPTE  00000000    00000000    00000000
     PGTE  0008        XPTE  00000000    00000000    00000000
     PGTE  0008        XPTE  00000000    00000000    00000000
     PGTE  0008        XPTE  00000000    00000000    00000000
     PGTE  0008        XPTE  00000000    00000000    00000000


ASID    00000002 CONTROL BLOCKS


RSMHD   00FED868 ANCHORED IN ASCBRSM
     VSTO  00AFFC00    SPCT  00FED8B0    ASCB  00FFA9B8    FLG1  00          RSV1  000000      RSV2  00000000
     LFQF  0000        LFQL  0000        LSQAF 0B10        LSQAL 1D00        LIOQF 00000000    LIOQL 00000000
     FOEQ  00AFD6D8

SPCT    00FED8B0 ANCHORED IN RSMPSCT
     TSWRT 00000000    TFIX  0000        TLSQA 0004        TNSEG 10          TSSEG 08          TFLG1 00
     TIDEN E2          TWSSZ 000A        TSIZE 00A8
  LSQA AND FIX ENTRIES
     00000000    00000000    40000000    AFF04000    0001AFE0    40000002    AFD04000    0003AFC0
     00000000    00000000    00000000    00000000    00000000    00000000
  SEGMENT ENTRIES
     AFAFFB2001F0    AEAFD0300002    08AFCD480000    09AFC420D3B0    0AAFC3400000    0BAFC260D9E0
     0CAE6A480000    ADAE6730D860

ASMHD   00FED890 DEFINED AT  RSMASMHD
     FLAG1 00          FLAG2 00          SWPCT 0000        BKSLT 00000150    IOCNT 00000000    SWAPQ 00000000
     CAPQ  00000000    LOCK  00000000    VSRBP 00000000    LGEQ  00000000

PGT/XPT 00AFFB20 FOR
    VSA 00AF0000
     PGTE  0009        XPTE  10000000    00000000    00000000
```

Figure 11. Format of ASM Data (Part 11 of 13)

```
     PGTE  0009        XPTE  10000000    00000000    00000000
     PGTE  0009        XPTE  10000000    00000000    00000000
     PGTE  0009        XPTE  10000000    00000000    00000000
     PGTE  0009        XPTE  10000000    00000000    00000000
     PGTE  0009        XPTE  10000000    00000000    00000000
     PGTE  18A9        XPTE  18004080    0004017B    00000000
     PGTE  0AD9        XPTE  10004080    00040003    00000000
     PGTE  1869        XPTE  10004080    000400CB    00000000
     PGTE  1B59        XPTE  10004080    0004009C    00000000
     PGTE  12D9        XPTE  10004080    00040076    00000000
     PGTE  0B89        XPTE  00004080    00040035    00000000
     PGTE  0D01        XPTE  00000000    00000000    00000000
     PGTE  0BE1        XPTE  00000000    00000000    00000000
     PGTE  0BD1        XPTE  00000000    00000000    00000000
     PGTE  0B11        XPTE  00000000    00000000    00000000


LGVT    00FE9000 ANCHORED IN ASMLGVT
     IDENT LGVT        LGVEP 00FE90B8    MAXLG 0000007D    SIZE  00000400


LGVTE
     ENEXT 00000000    ELGID 00000000

LGVTE
     ELGEP 00FD52B8    EASCB 00FE38B8

LGE     00FD52B8 ANCHORED IN LGVELGEP
     PROCF 00000000    PROCL 00000000    FLAG1 00          SLTCT 0000
     ASPCT 00AC66E0    NEXT  00000000    LGID  00000001

ASPCT   00AC66E0 ANCHORED IN LGEASPCT
     IDENT ASPC        LGID  00000001    BKEY  00000000    00000000    00000000    ASCB  00FE38B8
```

Figure 11. Format of ASM Data (Part 12 of 13)

```
  IDENT ASPC        LGID  00000001    BKEY  00000000    00000000    00000000    ASCB  00FE38B8
  LGE   00FD52B8    FLAG  00          MAXPN 00000155    BKSLT 0000000A    SAVCT 00000000
  LEXCT 0000        AEXCT 0000        ASSTP 00000000    00000000    00000000    00000000
LPMES
  00000000    8004005A    8003015F    00000000    00000000    000302ED    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
  00000000    00000000    00000000    00000000    00000000    00000000    00000000    00000000
```

Figure 11. Format of ASM Data (Part 13 of 13)

```
                                                                    MODULE AMDSADMP  DATE 09/24/77  TIME 01.55.25  PAGE 0001

TITLE FROM DUMP: SLIP STOP OF V CPU(0),OFFLINE 9/24/77 T69598

                                                  * * * *  C P U   D A T A  * * * *

(1a)
CSD     FFDD30      +0  CSD   C3E2C440 CPUJS 4000  CHAD 0008   CPUAL C000   CPUOL 0002   SCFL1 00   SCFL2 00   SCFL3 00
                    +F  SCFL4 00       RV043 00000000          MF1CP 0000   ACR   00     FLAGS 80
                   +18  MAFF  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                   +40        00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                   +68  RV044 0000     DDRCT 0000  GDCC 00000001    GDINT 00000001    GDTOD 00000001   TCNT  00000000
                   +7C  UCNT  00000000       MASK 80004000 20001000 08000400 02000100 00800040 00200010 00080004 00020001
                   +A0  RESV  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                   +C8        00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                   +F0        00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                  +118        00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

(1b)
*SVT    03E518     +00  ISECT  00059F28 GSCH1 00046676  GSCH2 0004668C  MSEEP 0002BB88  MSDEP 0002BB68
                   +14  RSME2  00047A90 JSTEQ 00000000  DSREQ 00000000  GSMQ  00FD5010  GSPL  00000000
                   +28  LSMQ   00000000
                   +2C  WAS    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                   +6C  DACTV  00000000 00000000 00000000 00000000       WAIT  00000000 00000000 00000000 00000000
                   +8C  ISECR  00000000

(2a)
LCCA    FFCE28      +0  LCCA  D3C3C3C1  CPUA 0041   RV77 0000
                    +8  PGR1  00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
                   +28        00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
                   +48  PGR2  80B7A568   00FF40FC   C9C7C5F0   00023368   00096E3D8  00096DF90  000368AC   00000185
                   +68        80B7A568   0000B568   00035418   80B7A568   FFFFFFFD   000228C0   00022860   00C94600
                   +88  PPSW  070C0000 00C94600   PINT 00060011   PVAD  00C94600   MCR1 0F5D7C00       CR0   00000000
                   +A0  RESV  00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
                   +C0        00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
                   +E0  XGR2  00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
                  +100        00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
                  +120  XGR3  00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
                  +140        00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000
                  +160  RSGR  00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000

PCCA    FFD1F0      +0  PCCA  D7C3C3C1  CPID F0F1F0F6 F0F0F1F5 F0F1F6F8  CPUA 0001        CAFM 4000        TQEP 000230A8
(2b)               +18  PSAV  00FEE000  PSAR 005EF000  RV81 00000000  RV82 00000000  RV83 00000000  RV84 00000000
                   +30  RV85  00000000  RV86 00000000  RV87 00000000  RV88 00000000  RV89 00000000  RV90 00000000
                   +48  RV91  00000000  RV92 00000000  RV93 00000000  RV94 00000000  RV95 00000000  RV96 00000000
                   +60  RV97  00000000  RV98 00000000  RV99 00000000  RV9A 00000000  RV9B 00000000  RV9C 00000000
                   +78  RV9D  00000000  RV9E 00000000  TMFL 80        TODE 00        CCE  00        INTE 00
                   +84  RPB   00000000  RISP 00        EMS2 00        EMS3 00        RMSB 00        EMSP 00000000
```

Figure 12. CPU Data (Example) (Part 1 of 5)

```
                                  ***** STORE-STATUS DATA FOR CPU 0001 *****

                    (2c) PSW          000A0000 0000001B

                         GPRS 0-7     00000004   00000000   00000000   00FFAFA0   00FFAFC0   00FFAAF8   0000C000   00FFD1F0
                         GPRS 8-15    00FFCE28   00000000   00000000   00042168   60033F88   000421A8   8003408A   00000000

                         CTRS 0-7     D080EC40   0F5D7C00   FFFFFFFF   00000000   00000000   00000000   00000000   00000000
                         CTRS 8-15    00000000   00000000   00000000   00000000   00000000   00000000   EFC00000   005AAA78

                         FPRS 0-2     00000000 00000000 00022B78 00022B78
                         FPRS 4-6     00000000 00000000 00000000 00000000

                         CPU-TIMER    00FFFFEC F4A01000
                         CLOCK-COMP   8B6EB265 986E5000
                         PREFIX       005EF000

PSA (2d) 000000    +0    RNPSW 000C0000 00034180    ROPSW 00000000 00000000   CVT    00023368    RESV  00FFF000
                   +18   EOPSW 070E0000 00000000    SOPSW 070C1000 00039802   POPSW  070C0000 00C94600
                   +30   MOPSW 00000000 00000000    IOPSW 070E0000 00000000   CSW    005F5D78 08000000        CAW   005F5D70
                   +4C   CVT2  00023368    TIMER FDDF2DFF    TRACE 00FD06F0   ENPSW  040C0000 000592D8
                   +60   SNPSW 040C0000 00047DE8    PNPSW 000C0000 0002A744   MNPSW  00080000 00011EB0
                   +78   INPSW 040C0000 00058210    RESV  00000000            SPAD   0000       EICOD 1202  RESV  00
                   +89   SVILC 02       SVCN  000D        RESV  00          PIILC  06         RV049 00    PICOD 11
                   +90   TEA   00C94600 RESV  00          MCNUM 00          PERCD  00         RESV  00    PER   00000000
                   +9C   RESV  00       MTRCD 000000      RESV  00000000    MPL    0004B350   CHNID 10020000
                   +AC   IOEL  005DFA18 LCL   FFFFFFFF    RESV  0000        RESV   00         RESV  00    IOA   00000005
```

Figure 12. CPU Data (Example) (Part 2 of 5)

CPUDATA OUTPUT

Figure 12 is an example of CPUDATA output. Contents are as follows:

(1a) CSD hhhhhh -- Common system data area (CSD).[1]

(1b) SVT hhhhhh -- Supervisor vector table (SVT).

(2) Data for processor on which the dump was taken:

(2a) LCCA hhhhhh -- Logical configuration communication area (LCCA).[1]

(2b) PCCA hhhhhh -- Physical configuration communication area (PCCA).[1]

(2c) STORE-STATUS DATA FOR CPU hhhh -- stored status information for the indicated processor:
PSW -- Current program status word (hex).
GPRS 0-7, GPRS 8-15 -- General-purpose registers (hex).
CTRS 0-7, CTRS 8-15 -- Control registers (hex).
FPRS 0-2, FPRS 4-6 -- Floating-point registers (hex).
PROCESSOR TIMER -- Processor timer value (hex).
CLOCK-COMP -- Clock comparator value (hex).
PREFIX -- Prefix register (hex).

NOTE: If you do not perform a STORE STATUS operation, only the general-purpose registers appear; other items in the above list are not shown. Figure 18 shows the formats for the PSW, control registers, and floating-point registers.

(2d) PSA hhhhhh -- Prefixed storage area (PSA) bytes 0 3FF hex.[1]

[1]Data area acronym and address, followed by formatted fields. Each line of print includes:

- The offset of the first printed field, relative to the beginning of the data area.
- The name of each printed field (minus the data area acronym) and the contents of the field in hexadecimal format. In the CSD, for example, the printed name "CPUJS" identifies the CSD field CSDCPUJS. (The printed name "RESV" indicates that the field is reserved and contains no useful data.)

For a description of each printed field, see the appropriate data area description in Volume 2 or Volume 3.

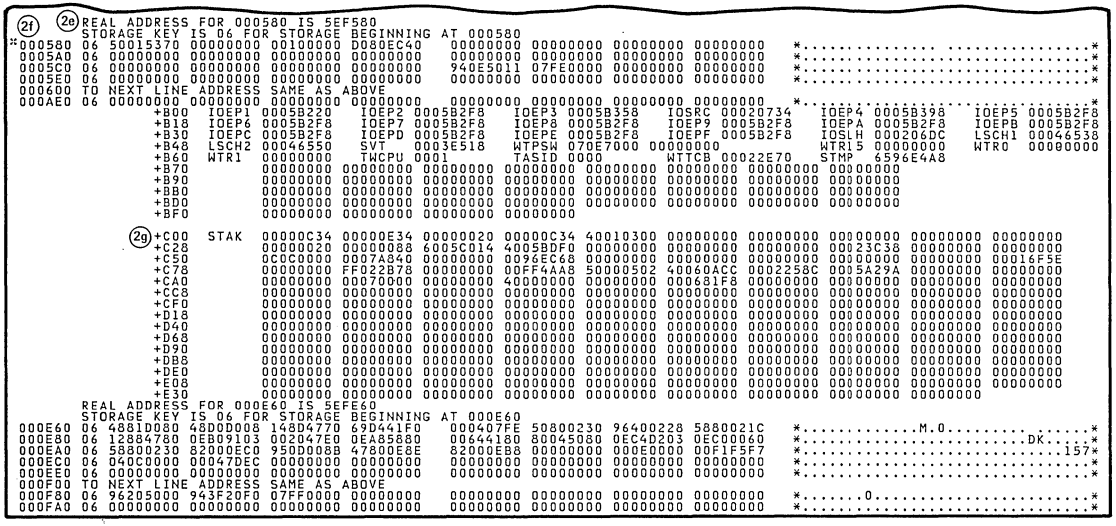


Figure 12. CPU Data (Example) (Part 3 of 5)

(2e) REAL ADDRESS FOR 000580 IS hhhhh -- Real (absolute) address corresponding to PSA offset 580 hex. (An absolute address is an address to which prefixing has been applied. Prefixing is described in GA22-7000; in effect, it simply interchanges the 4K bytes of storage at real address 0 with the 4K bytes of storage at the real address contained in the prefix register. Prefixing is applied to all references to real storage and to keys in storage, except for references to the PSA during a STORE STATUS operation, and except for references by a channel to extended logout locations, to indirect-data-address words, and to CCWs.)

(2f) Prefixed storage area (PSA) bytes 400 -- FFF hex. Except for bytes C00 - E5F hex (described below), each line of print includes:

- The PSA offset (virtual address) of the first printed word of storage.
- The storage protection key.
- Eight words of storage in hexadecimal format.
- The same eight words in EBCDIC format, with periods representing characters other than blank, A - Z, and 0 - 9.

(2g) +C00 STAK - hhhhhhhh ... -- Normal FRR stack (PSA bytes C00 - E5F hex). Each line of print includes:

- The offset of the first printed word of storage, relative to the beginning of the PSA.
- Ten words of storage in hexadecimal format.

For a description of FRR stack contents, refer to the FRRS description in Volume 2. Note that FRRS offset 0 corresponds to PSA offset C00 hex.

```
(3) LCCA  FEC290    +0   LCCA D3C3C3C1  CPUA 0041   RV77 0000
                    +8   PGR1 00000040  00CA5FB0   00C589A8   00028160   00C58BD8   00C58848   40E9E164   00FFCA38
                    +28       000EC008  00C54F40   60E685C0   000A5FB0   60F96E9C   00C54F40   60E68880   00000000
                    +48  PGR2 00000040  00CA5FB0   00C589A8   00028160   00C58BD8   00C58848   40E9E164   00FFCA38
                    +68       000EC008  00C54F40   60E685C0   00CA5FB0   60F96E9C   00C54F40   60E68880   00000000
                    +88  PPSW 070C2000 00F96EA8    PINT 00040011   PVAD  C00A5FB4   MCR1 0F3DFC00   CR0  00000000
                    +A0  XGR1 00000000  00000000   00000000   00000000   00000000   00000000   00000000   00000000
                    +C0       00000000  00000000   00000000   00000000   00000000   00000000   00000000   00000000
                    +E0  XGR2 00000000  00000000   00000000   00000000   00000000   00000000   00000000   00000000
                   +100       00000000  00000000   00000000   00000000   00000000   00000000   00000000   00000000

    PCCA  FEC048    +0   PCCA D7C3C3C1  CPID F0F0F0F2 F3F0F1F4 F0F1F5F8 CPUA 0001      CAFM 4000       TQEP 00000000
                    +18  PSAV 00FDA000  PSAR 00382000  RV81 00000000  RV82 00000000  RV83 00000000  RV84 00000000
                    +30  RV85 00000000  RV86 00000000  RV87 00000000  RV88 00000000  RV89 00000000  RV90 00000000
                    +48  RV91 00000000  RV92 00000000  RV93 00000000  RV94 00000000  RV95 00000000  RV96 00000000
                    +60  RV97 00000000  RV98 00000000  RV99 00000000  RV9A 00000000  RV9B 00000000  RV9C 00000000
                    +78  RV9D 00000000  RV9E 00000000  IMFL 80        TODE 00        CCE  00        INTE 00

                         ***** STORE-STATUS DATA FOR CPU 0001 *****
         PSW          070E0000 00000000
         GPRS 0-7     00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
         GPRS 8-15    00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
         CTRS 0-7     C080EC40  0F3DFC00  FC000000  00000000  00000000  00000000  00000000  00000000
         CTRS 8-15    00000000  00000000  00000000  00000000  00000000  00000000  EFC00000  003EE2F0
         FPRS 0-2     00000000 00000000 00000000 00000000
         FPRS 4-6     00000000 00000000 00000000 00000000
         CPU-TIMER    7FFFFFDC E86E4C01
         CLOCK-COMP   FFFFFFFF FFFFFC00
         PREFIX       00382000
```

Figure 12. CPU Data (Example) (Part 4 of 5)

(3) Data for alternate processor, corresponding to data described under (2) above. Store-status data results from a STORE STATUS operation initiated by the dump program. Addresses printed for fields within the PSA are virtual addresses used by the processor on which the dump was taken (not those used by the alternate processor).

```
PSA    FDA0C0      +0  RNPSW 040C0000 0003E59E    ROPSW 00000000 00F40000  CVT   00028160   RESV  00000000
                   +18  EOPSW 070E0000 00000000    SOPSW 070C1000 00F96F98  POPSW 070C2000 00F96EA8
                   +30  MOPSW 00000000 00000000    IOPSW 070E0000 00000000  CSW   003EA360 0C000001       CAW   0000B718
                   +4C  CVT2  00028160    TIMER FFC7B9FF     TRACE 00FC8D00  ENPSW 040C0000 0002A5F6
                   +60  SNPSW 040C0000 0002A146    PNPSW 00000000 0003D9E8  MNPSW 00080000 00044680
                   +78  INPSW 040C0000 00037C3E    RESV  00000000           SPAD  0000       ICOD  1201  RESV  00
                   +89  SVLIC 02          SVCN  000A          RESV  00      PIILC 04         RV049 00    PICOD 11
                   +90  TEA   000A5FB4    RESV  00            MCNUM 00      PERCD 00         RESV  00    PER   00000000
                   +9C  RESV  00          MTRCD 000000        RESV  00000000 00000000        CHNID 00000000
                   +AC  IOPL  003E00F0    LCL   FF000000      RESV  0000     RESV  00         RESV  00    IOA   00000537
                   +BC  RESV  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                   +E0        00000000 00000000
                   +E8  MCIC  00000000 00000000    RESV  00000000 00000000  FSA   00000000   RGNCD 00000000
                  +100  FLA   00000000   00000000   00000000   00000000   00000000   00000000   00000000   00000000

(4)                                    ***** UNPREFIXED 0-4K STORAGE *****
PSA    FF5000      +0  RNPSW 00080000 000074FA    ROPSW A003FE14 00028160  CVT   08000110   RESV  60000001
                   +18  EOPSW 00000000 00000000    SOPSW 00000000 00000000  POPSW 00000005 E0069B8C
                   +30  MOPSW 00000000 00000000    IOPSW 00000000 00000000  CSW   00007FF0 00000000       CAW   00007FB8
                   +40  CVT2  00028160    TIMER FF69F3FF     TRACE 00F08D00  ENPSW 040E0000 0002A5F6
                   +60  SNPSW 040E0000 0002A146    PNPSW 000E0000 0003D9E8  MNPSW 00020000 001E8E7E
                   +78  INPSW 040E0000 00037C3E    RESV  00000000           SPAD  0000       ICOD  0000  RESV  00
                   +89  SVLIC 00          SVCN  0000          RESV  00      PIILC 00         RV049 00    PICOD 00
                   +90  TEA   00000000    RESV  00            MONUM 00      PEROD 00         RESV  00    PER   00000000
                   +90  RESV  00          MTRCD 000000        RESV  00000000 00000000        CHNID 20000000
                   +A0  IOPL  00000000    LCL   00000000      RESV  0000     RESV  00         RESV  00    IOA   00000351
                   +B0  RESV  00000000 00000000 00000000 00000000 00000000 00000000 00000000 7FFFFFDC E86E4001
                   +E0        FFFFFFFF FFFFF000
                   +E8  MCIC  00000000 00000000    RESV  00000000 00000000  FSA   00000000   RGNCD 00000000
                  +100  FLA   070E0000   00000000   00382000   00000000   31000136   60000005   08000110   60000005
                  +120        05007800   60000800   08000140   60000001   00000000   00000000   00000400   00000000
```

Figure 12. CPU Data (Example) (Part 5 of 5)

(4) UNPREFIXED 0-4K STORAGE -- Contents of storage at absolute addresses 000000 - 000FFF, corresponding to the prefixed storage area described under (2) above. Addresses printed for fields within this area are virtual addresses used by the processor on which the dump was taken.

CVTMAP OUTPUT (COMMUNICATION VECTOR TABLE)

Figure 13 is an example of CVTMAP output. Contents are as follows:

CVT hhhhhh
CVT address, followed by formatted fields. Each line of print includes:

- The offset of the first printed field, relative to the beginning of the CVT.
- The name of each printed field (minus the acronym CVT) and the contents of the field in hexadecimal format. For example, the printed name "TCBP" identifies the CVT field CVTTCBP. (The printed name "RESV" indicates that the field is reserved and contains no useful data.)

For a description of each field, refer to the CVT description in Volume 2.

```
                                                                MODULE AMDSADMP  DATE 09/24/77  TIME 01.55.25  PAGE 0011

                                    * * * *  C O M M U N I C A T I O N   V E C T O R   T A B L E  * * * *


CVT 023368
     -8  MDL   00000148    RELNO F0F3F7C4
     +0  TCBP  00000218    0EF00 00034950    LINK  00023024    RESV  00000000    BUF   00000000    XAPG  00018480
    +18  OVL00 00034A8C    PCNVT 00035366    PRLTV 000352E0    ILK1  0000C8A8    ILK2  0000C996   *XTLER 000354F4
    +30  SYSAD 000040B8    BTERM 00035808    DATE  0077267F    MSLT  00035EC0    ZDTAB 00035F88    XITP  00036028
    +48  RESV  00000000    OFN00 00000000    EXIT  0A03        BRET  07FE        SVDCB 0002302C    TPC   00023068
    +5C  RESV  00000000    SJQ   00000000    CUCB  00036140   *QTE00 0002FCF0    QTD00 0002FD06    STB   00010498
    +74  DCB   13036840    RESV  00000000    IXAVL 0001EF00    NUCB  000A0000    FBOSV 00D93000    QDS   0001D1FC
    +8C  ILCH  0000D1F0   *RV516 00000000    MSER  00035EC0    OPT01 0001A2B8    RESV  00000000    RESV  00000000
    +A4  MZ00  00FFFFFF    1EF00 00000000    QOCR  00000000    QMWR  00036880    SNCTR 0000   OPTA AB    OPTB 20
    +B8  QCDSR 0003721A    QLPAQ 00023060    RESV  00000000    SMCA  00FE1C60    ABEND 00023830    USER  00000000
    +D0  MDLDS 00000000    QABST 0A0D        LNKSC 0A06        TSCE  00000000    PATCH 000376D8    QMS   000238F8
    +E4  RESV  00000000    DSCRI 000377A0    GTF   0002B6F0    AQAVT 00000000   *VOLM2 80000000   *RESV  00000000
    +FC  EXT1  000237A0    CBSP  00FE9098    PURG  00000000    AMFF  80000000    QMSG  00037838    DMSR  00D8A000
   +114  RESV  00000000    RESV  00000000    REAL  000B7000    PTRV  00037A58   *RESV  00000000    JESCT 00037AE8
   +12C  RESV  00000000    TZ    FFFFBCF2    MCHPR 00000000    EORM  005FFFFF   *RESV  00000000   *RESV  00000000
   +144  APF   00037B22    EXT2  000237AC    HJES  00000000    RESV  00000000    RESV  00000000    RESV  00000000
   +15C *GETL  80C537E0    LPDSR 000373DE    PVTP  00037C48    LPDIA 80B6F000    RESV  00000000    RESV  00000000
   +174  SLIDA 00000000    RESV  00000000    RESV  00000000    RESV  00000000    RESV  00000000    RESV  00000000
   +18C  RESV  00    SPVLK 00   CTLFG 10   APG 09   TRACE 07FA   TRAC2 07FB    RSCN  0002C83A    TAS   00000000
   +19C  RESV  00000000    SHRVM 00970000    0VL01 00034A7A    RESV  00000000    RESV  00000000    RESV  00000000
   +1B4  ASCRF 00FE5960    ASCRL 00FE7558    PUTL  80C52A78    SRBRT 0001BDB0    GLTQA 000383E0    SMFEX 00038488
   +1CC  CSPIE 00D69BEA    PTGT  80C52D50    IOSPL 0000        DSSAC 0000        STCK  80C51000    MAXMP 00010000
   +1E0  SCAN  80B9FA08    AUTHL 00FD8440    BLDCP 00038578    GETCL 00038820    FRECL 00038998    DELCP 00038B20
   +1F8  CRMN  00038F9E    CRAS  0003AE38    QSAS  0003AE52    FRAS  0003AE58    S1EE  0003B7C8    PARS  80C5E000
   +210  QUIS  0003B8B8    STXU  00E45720    OPTE  00028A1C    SDRM  00D436CC    IOSCS 00000000    AQTOP 0003BF88
```

Figure 13. Communication Vector Table (Example)

EDIT OUTPUT (GTF TRACE RECORDS)

Figure 14 is an example of EDIT output, consisting of GTF trace records. Record formats are described under "GTF Trace Records" later in this section.

```
                                                              EXTERNAL TRACE - DD TAPE                              PAGE 0049

EXT   ASCB 00FDE9D8  CPU 0000  PSW 070C1202 0C04AAEA  TCB 00B27B78  INT CPU 0001
           TIME    76276.502554
DSP   ASCB 00FECB40  CPU 0001  PSW 070C1000 0C07A59E  TCB 00B24988  R15 6003EE0C  R0 00000001  R1 FFF8484C
           TIME    76305.870737
LSR   ASCB 00FDE9D8  CPU 0000  PSW 070C00C0 0C04AAEA  TCB 00B27B78  R15 9007AFC8  R0 00000000  R1 00B27BA8
           TIME    76276.503834
SVC   ASCB 00FECB40  CPU 0001  PSW 070CC0C0 00C0A61E  TCB 00B24988  R15 00C0A360  R0 00000FE2  R1 00128FA0
           TIME    76305.871747
SIO   ASCB 00FDE9D8  CPU 0000  CPA 001F4628 00FF4628  CAW 10009EC8  DEV ADD 0352  STATUS 0C00  CC 0
           TIME    76276.506371
DSP   ASCB 00FDE9D8  CPU 0000  PSW 071C0000 0C0725A2  TCB 00B27B78  R15 0004A108  R0 00FFFC70  R1 0007E800
           TIME    76276.507444
DSP   ASCB 00FECB40  CPU 0001  PSW 070C0000 0CC0A61E  TCB 00B24988  R15 00C00000  R0 00000002  R1 00128FA0
           TIME    76305.874873
EXT   ASCB 00FDE9D8  CPU 0000  PSW 071C12C2 0C0725A2  TCB 00B27B78  INT CPU 0001
           TIME    76276.507988
SVC   ASCB 00FECB40  CPU 0001  PSW 070CCC70 0C07A790  TCB 00B24988  R15 92BE4400  R0 0008B000  R1 0008C000
           TIME    76305.875676
SIO   ASCB 00FECB40  CPU 0000  CPA 001F4190 00FF4190  CAW 0000A6B8  DEV ADD 0581  STATUS 0C00  CC 0
           TIME    76276.509075
DSP   ASCB 00FDE9D8  CPU 0000  PSW 071C0000 0C0725A2  TCB 00B27B78  R15 0004A108  R0 00FFFC70  R1 0007E800
           TIME    76276.510198
EXT   ASCB 00FDE9D8  CPU 0000  PSW 071C1201 0C0725A2  TCB 00B27B78  INT CPU 0001
           TIME    76276.510710
DSP   ASCB 00FDE9D8  CPU 0000  PSW 071C0000 0C0725A2  TCB 00B27B78  R15 0004A108  R0 00FFFC70  R1 0007E800
           TIME    76276.511763
```

Figure 14. GTF Trace Records (Example)

```
                                                    MODULE AMDSADMP  DATE 02/07/73  TIME 18.28.41  PAGE 0001

TITLE FROM DUMP: 0C6 AFTER XMPOST

                                          * * * *  F O R M A T  * * * *

(1)
GLOBAL SERVICE MANAGER QUEUE

SRB    00B040      +0  ID   E2D9C240   FLNK 00002060   ASCB 00024DF0   CPAFF 0010      PASID 0020      PTCB  000102A0
                  +14  EP   00F093D0   RMTR 00F12340   PARM 00FF1040   SAVE  00F0B458  PKF   00        PRIOR 00
                  +26  RESV 0000       RESV 00000000

ERROR FINDING SRB 002060

SRB CHAIN BROKEN
(2)
LOCAL SERVICE MANAGER QUEUE

SRB    00B0A0      +0  ID   E2D9C240   FLNK 0000B0E0   ASCB 000A1850   CPAFF 0010      PASID 0018      PTCB  000210F4
                  +14  EP   00D002F0   RMTR 00D04000   PARM 00D45440   SAVE  00D01180  PKF   00        PRIOR 00
                  +26  RESV 0000       RESV 00000000

SRB    00B0E0      +0  ID   E2D9C240   FLNK 0000B120   ASCB 30044780   CPAFF 0010      PASID 0030      PTCB  30085020
                  +14  EP   300CD203   RMTR 200CF100   PARM 000F5584   SAVE  000CE100  PKF   00        PRIOR 00
                  +26  RESV 0000       RESV 00000000
```

Figure 15. System-Related Control Blocks (Example) (Part 1 of 2)

FORMAT, PRINT JOBNAME=, AND PRINT CURRENT OUTPUT

The FORMAT, PRINT JOBNAME=, and PRINT CURRENT control statements produce similar types of output, as follows:

FORMAT
- System-related control blocks (SRB queues)
- Job-related control blocks (for each address space)
- Task-related control blocks (for each task in each address space)
- TCB summary (summary of TCBs for all address spaces)

PRINT JOBNAME=
- Job-related control blocks (for selected address spaces)
- Task-related control blocks (for each task in each selected address space)
- PSW, registers, and virtual storage (for each selected address space)
- TCB summary (summary of TCBs for all selected address spaces)

PRINT CURRENT
- Job-related control blocks (for the current address space)
- Task-related control blocks (for each task in the current address space)
- PSW, registers, and virtual storage (for the current address space)
- TCB summary (summary of TCBs in the current address space)

Each type of output is described in the following series of topics:
- System-related control blocks
- Job-related control blocks
- Task-related control blocks
- PSW, registers, and virtual storage
- TCB summary

System-Related Control Blocks

Figure 15 is an example of system-related control blocks, which are identified as follows:

(1) GLOBAL SERVICE MANAGER QUEUE -- Queue of newly created SRBs, awaiting transfer to the global service priority list (GSPL).

SRB hhhhhh -- Service request block (SRB).[1]

(2) LOCAL SERVICE MANAGER QUEUE -- Queue of newly created SRBs, awaiting transfer to the local service priority list (LSPL).

SRB hhhhhh -- Service request block (SRB).[1]

[1]SRB address, followed by formatted fields. Each line of print includes:

- The offset of the first printed field, relative to the beginning of the SRB.
- The name of each printed field (minus the prefix "SRB") and the contents of the field in hexadecimal format. For example, the printed name "ID" identifies the SRB field SRBID. (The printed name "RESV" indicates that the field is reserved and contains no useful data.)

For a description of each printed field, refer to the SRB description in Volume 3.

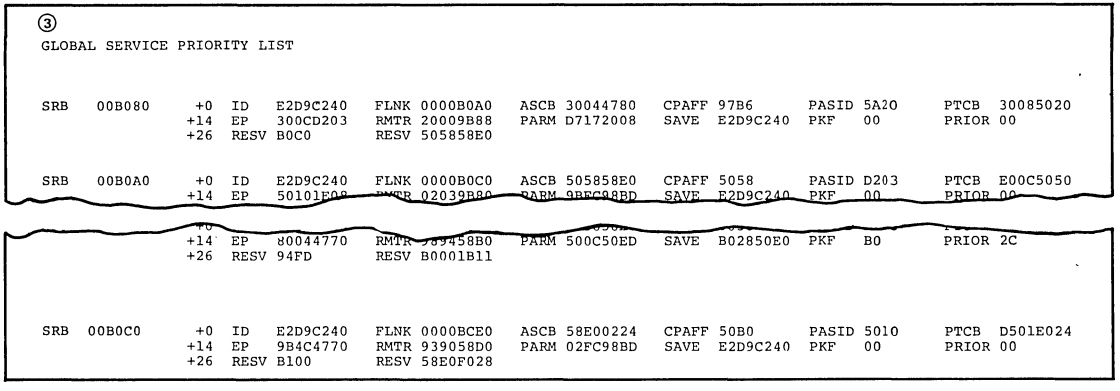


Figure 15. System-Related Control Blocks (Example) (Part 2 of 2)

③ GLOBAL SERVICE PRIORITY LIST -- Global SRB dispatching queue.

```
(1)                                                    MODULE AMDSADMP  DATE 09/24/77  TIME 01.55.25  PAGE 0014
JOB *MASTER*

ASCB   022460      +0  ASCB C1E2C3C2   FWDP 00FE5870   BWDP 00000000   CMSF 00000000   SVRB 00FECA40   SYNC 0000240C
(2)               +18  IOSP 00000000   TNEW 00022B78   CPUS 00000001   ASID 0001       SEQN 0001       IOSM 0016
                  +2A  HLHI 03         DP   FF         STOR 0F5D7C00   LDA  0096EC68   RSM  C0022908   CSCB 00000000
                  +3C  TSB  00000000   EJST 00000008 EECBB8C4  EWST 8B6EB265 97973800   JSTL 00FFFFFF   ECB  00000000
                  +58  UBET 00000000   TLCH 00000000   DUMP 00965DD0   AFFN FFFF   RCTE 00   FLG1 83   TMCH 00000000
                  +6C  ASXB 00022568   SWCT 0B31       DSP1 00         FLG2 E0     RV48 0000 SRBS 0000 VSC  0000
                  +7A  NVSC 0000       RCTP 00000000   LOCK 00000041   LSQH 00000000   QECB 00000000   MECB 00000000
                  +90  OUCB 00FE6F58   OUXB 00FE5000   FMCT 0023       RS01 0000       XMPQ 00000000   IQEA 00000000
                  +A4  RTWA 00000000   MCC  00000000   JBNI 00000000   JBNS 00023040   SRQ  00000000   VGTT 00000000
                  +BC  PCTT 00000000   SSRB 0000       SMCT 00         RS14 00         SWTL 00000000
                  +C8  SRBT 00000000 96C44B3C         LSMQ 00000000   LSPL 00000000   TCBS 00000001   TCBL 00000000
                  +E0  WPRB 00022550   NDP  FF    TNDP FF    NTSG FF   IODP FF         RV49 00000000   RV50 00000000
(3)
ASXB   022568      +0  ASXB C1E2E7C2   FTCB 00022670   LTCB 00965040   TCBS 000D       RS00 0000       MPST 00000000
                  +14  LWA  00000000   RV14 00000000   RV15 00000000   IHSA 00023968
                  +24  FLSA 00FF4AA8 000A5FA0 00967944 0F969C38 4001758E 00FF4AA8 000010A8 00FF40FC A0017960
                  +48       800186A6 00FF4D48 00000C7C 0002258C 900183EA 90010100 00017D78 00000C5C 40060A02
                  +6C  FRWA 00023C38   SPSA 0004ADF8   RSMD 00000000   RCTD 00000000   DDR  00000000   OUSB 000240C0
                  +84  TAXE 00000000   PRG  40404040 40404040 40404040 40404040   PSWD 40404040 40404040
                  +A0  SIRB 00022860   ETSK 00022670   FIQE 00000000   LIQE 00000000   FRQE 00000000   LRQE 00000000
                  +B8  FSRB 00000000   LSRB 00000000   USER 00000000 000000 SECR   00   SENV 00000000   RV19 00000000
                  +D0  RV20 00000000   RV21 00000000   RV22 00000000   RV23 00000000   RV24 00000000   RV25 00000000
```

Figure 16. Job-Related Control Blocks (Example) (Part 1 of 2)

Job-Related Control Blocks

Figure 16 is an example of the major control blocks associated with a specific job (address space), but not with a specific task. Contents are as follows:

(1) JOB cccccccc -- Name of the job associated with the address space. The following names identify system address spaces:
MASTER -- Master scheduler.
HASP -- JES2.

(2) ASCB hhhhhh -- Address space control block (ASCB).[1]

(3) ASXB hhhhhh -- Address space extension block (ASXB).[1]

[1]Data area acronym and address, followed by formatted fields. Each line of print includes:

- The offset of the first printed field, relative to the beginning of the data area.
- The name of each printed field (minus the data area acronym) and the contents of the field in hexadecimal format. In the ASCB, for example, the printed name "FDWP" identifies the ASCB field ASCBFDWP. (The printed name "RESV" indicates that the field is reserved and contains no useful data.)

For a description of each printed field, refer to the appropriate data area description in Volume 2 or Volume 3.

```
(4) LOCAL SERVICE MANAGER QUEUE

    QUEUE IS EMPTY

(5) LOCAL SERVICE PRIORITY LIST

    SRB FAILED VALIDITY CHECK

    SRB   F09650    +0  ID   00000000   FLNK 00BC3F38   ASCB 00FAD638   CPAFF 0000   PASID 0001   PTCB  00018908
                   +14  EP   000388F0   RMTR 00038784   PARM 00FD967C   SAVE  00000000 PRF   00     PRIOR A0
                   +26  RESV 0000       RESV 000389FC

    SRB   FC5F38    +0  ID   E2D9C240   FLNK 00FE8804   ASCB 00FAD638   CPAFF 0000   PASID 001A   PTCB  00000000
                   +14  EP   00BDA620   RMTR 00BDA826   PARM 00BC3F64   SAVE  00000000 PKF   00     PRIOR 00
                   +26  RESV 0000       RESV 00000000

    SRB FAILED VALIDITY CHECK

    SRB   FE8804    +0  ID   40404040   FLNK 00000000   ASCB 00FA0638   CPAFF 0000   PASID 001A   PTCB  009CC9FB
                   +14  EP   0002603E   RMTR 00024B28   PARM 80FE8708   SAVE  00000000 PKF   00     PRIOR 00
                   +26  RESV 85D8       RESV 00000058
```

Figure 16. Job-Related Control Blocks (Example) (Part 2 of 2)

(4) LOCAL SERVICE MANAGER QUEUE -- Local SRB staging queue.

(5) LOCAL SERVICE PRIORITY LIST -- Local SRB dispatching queue.

```
(1)                                                                    MODULE AMDSADMP  DATE          TIME 00.40.18  PAGE 1013
JOB TCKPI303   STEP STI30301   PROCSTEP
                                                       (2)
                                                      *****   CURRENT TASK   *****
(3)
TCB  B9C468   RBP    00B9DD98   PIE      00000000   DEB   00B75780   TIO 00B78020   CMP    00000000   TRN   00000000
              MSS    00B9C828   PK-FLG   80010000   FLG   0000FFFF   LLS 00B9C7B8   JLB    00000000   JPQ   00B9C988
              FSA    01084FB0   TCB      00000000   TME   00000000   JST 00B9C468   NTC    00000000   OTC   00B9CCD0
              LTC    00000000   IQE      00000000   ECB   00B9C9D4   TSF 20000000   D-PQE  00B9F590   AQE   00B9C338
              STAB   00B9D474   TCT      80B9CF88   USER  00000000   SDF 00000000   MDID   00000000   JSCB  00B9CAA4
              RESV   00000000   IOBRC    00000000   XCPD  00000000   EXT 00000000   BITS   00000000   DAR   00000000
              EXT2   00B9C590   AECB     00000000   RESV  00000000   BAK 00B9CCD0   RTMWA  00000000   IOTM  00000000
              TMSAV  00000000   ABCR     00000000   RESV  00000000   FOE 00000000   SWA    00B9C688   RESV  00000000
              BID    E3C3C240   RTM1     00000000   ESTA  00000000   UKY 00B9C7F8   CPVI   004CFFFF   BYT1  40040000
              RPT    00000000   DBTB     00000000   SWAS  00B81E78   SCB 00000000   GTF    00000000   SVAB  00000000
              EVENT  00000000   RESV     00000000   RESV  00000000   RES 00000000   RESV   00000000   RESV  00000000

(4)
ACTIVE RBS

PRB   868C70  RTOPSW 00000000 00000000   RTIL-IC 00000000   TRAN 00000000   FLG1 00000000   WC-L-IC 0002003F
              RESV    00000020   APSW   00000000   SZ-STAB     00110080   FL-CDE 00882390   PSW 078D3000 00092B4A
              Q       00000000   WT-LNK 00868D00   NM CSTART08   EPA 092AC0   STA 092AC0   LN 00C540   ATR1 CB   ATR2 20

SVRB 88DB10   RTOPSW 0000000C 00000300   RTIL-IC 00000000   TRAN 00000000   FLG1 00000000   WC-L-IC 00000000
              RESV    00000000   APSW   00000000   SZ-STAB     00190022   LPDE     00000000   PSW 070C0000 00A7F4FA
              Q       00000000   WT-LNK 00868C70
              RG 0-7    000000E0   80092B40   80092AF4   00000008   00883610   00882A80   0085AC18   FD000000
              RG 8-15   008822EC   00000000   00000000   00091FA8   00092AC0   00092FB4   40092C54   00000000
              EXTSA     0000000C   00000000   00000000   00000000   00000000   00000000   00000000   00000000
                        0085A020   00000000   00000000   00000000   00000000   00000000   00000000   00000000
                        00000000   00000000
```

Figure 17. Task-Related Control Blocks (Non-MVS/System Extensions Example) (Part 1 of 10)

```
(1)                                                               MODULE AMDSADMP  DATE 09/24/77  TIME 01.55.25  PAGE 0018
JOB CONSOLE     STEP SYSTEM      PROCSTEP VMS    (2)
                                                 *****   CURRENT TASK   *****
(3)
TCB  022B78     RBP    00FEC898   PIE      00000000   DEB    00967944   TIO    0096C9C8   CMP    00000000   TRN    00000000
                MSS    0096F4A0   PK-FLG   00000080   FLG    0000FFFF   LLS    00969C18   JLB    00000000   JPQ    00000000
                RG 0-7    FF000030   00FFFFF1   00036108   00023368   00022B78   00FEC898   00CFEB58   00B65D78
                RG 8-15   00B65D78   00969C38   00036704   00FEC8F8   00CF59A8   009698F0   B0CF5A9A   00CF59A8
                FSA    00000000   TCB      0096EA80   TME    80FF9A88   JSTCB  00022B78   NTC    00000000   OTC    00022670
                LTC    00000000   IQE      00000000   ECB    00000000   TSFLG  20000000   D-PQE  0096F224   AQE    00969490
                STAB   00095EB0   TCT      00000000   USER   00000000   NDSP   00000000   MDIDS  00000000   JSCB   0096E154
                RES    00000000   IOBRC    00000000   EXCPD  00000000   EXT1   00000000   BITS   00000004   DAR    00000000
                EXT2   00022CA0   AECB     00000000   TIRB   00000000   BACK   00022970   RTWA   00000000   IOTIM  00000000
                TMSAV  00000000   ABR-TID  0000FD00   RES    80000041   FOE    00000000   SWA    00000000   RESV   00000000
                TCBID  E3C3C240   RTM12    00000000   ESTAE  00000000   UKYSP  0096F7B0   CPU-AF 0003FFFF   FBYTES 00000000
     EXT2       GTF    00000000   RCMP     00000000
(4)
ACTIVE RBS

PRB  022CE0     RTOPSW 00000000 00000000   RTIL-IC 00000000   TRAN 00E43190   FLG1 00000004   WC-L-IC 00020048
                RESV   00000000   APSW     00000000   SZ-STAB    00110080   FL-CDE 00000000   PSW 070C3000 00E41B5A
                Q      00000000   WT-LNK   00022B78

SVRB FEC898     RTOPSW 00000000 00000000   RTIL-IC 00000000   TRAN 00000000   FLG1 02000000   WC-L-IC 0002000A
                RESV   00000000   APSW     00000000   SZ-STAB    001CD022   LPDE   00000000   PSW 070C1000 00CF5AB6
                Q      00000000   WT-LNK   00022CE0
                SCBB   00095F64   00000000   12FEC8F8   03FEC898   6400DB00
                FEPARM 00000000   80E35AD0   80E35AF0   00000000   0002D728   00000000
                RG 0-7    00000001   00036108   00036140   000362F4   00036098   0003613C   000010A8   00036098
                RG 8-15   90E41B12   90E41B58   50E419DE   00036704   00000050   000367A4   50E4255E   00E419D8
                EXTSA     00B65D78   00000000   C9C7C3F0   F1F0F7C2   00036704   00036140   00CFED40   00CFED0A
                          00B65D96   00000000   B0CF5A9A   009698F0   00095F64   00000000   12FEC8F8   03FEC898
                          6400DB00   00000000
```

Figure 17. Task-Related Control Blocks (MVS/System Extensions Example) (Part 2 of 10)

Task-Related Control Blocks

Figure 17 is an example of the major control blocks associated with a specific task. Contents are as follows:

(1) JOB cccccccc STEP cccccccc PROCSTEP cccccccc -- Name of job, job step, and cataloged procedure step (where applicable).

(2) ***** CURRENT TASK ***** -- Identifies control blocks for the current task.

(3) TCB hhhhhh -- Task control block (TCB).

(4) ACTIVE RBS -- Request blocks (RBs) queued to the TCB.

IRB hhhhhh -- Interruption request block (RB).
PRB hhhhhh -- Program request block (RB).
SIRB hhhhhh -- Supervisor interruption request block (RB).
SVRB hhhhhh -- Supervisor request block (RB).

The following fields, printed with the request block, are taken from the CDE, LPDE, or XTLST:

NM cccccccc -- Load module name (CDNAME field of CDE or LPDENAME field of LPDE, for PRB and SVRB only; RBEXRTNM field of RB for SIRB only).

EPA hhhhhh -- Load module entry point address (CDENTPT field of CDE or LPDENTP field of LPDE, for PRB, SIRB, and SVRB only).

STA hhhhhh -- Load module storage address (XTLMSBAD field of XTLST, for PRB only).

LN hhhhhh -- Load module length in bytes (XTLMSBLN field of XTLST, for PRB only).

ATR1 hh -- Load module attribute flags (CDATTR field of CDE or LPDEATTR field of LPDE, for PRB only).

ATR2 hh -- Load module attribute flags (CDATTR2 field of CDE or LPDEATT2 field of LPDE, for PRB only).

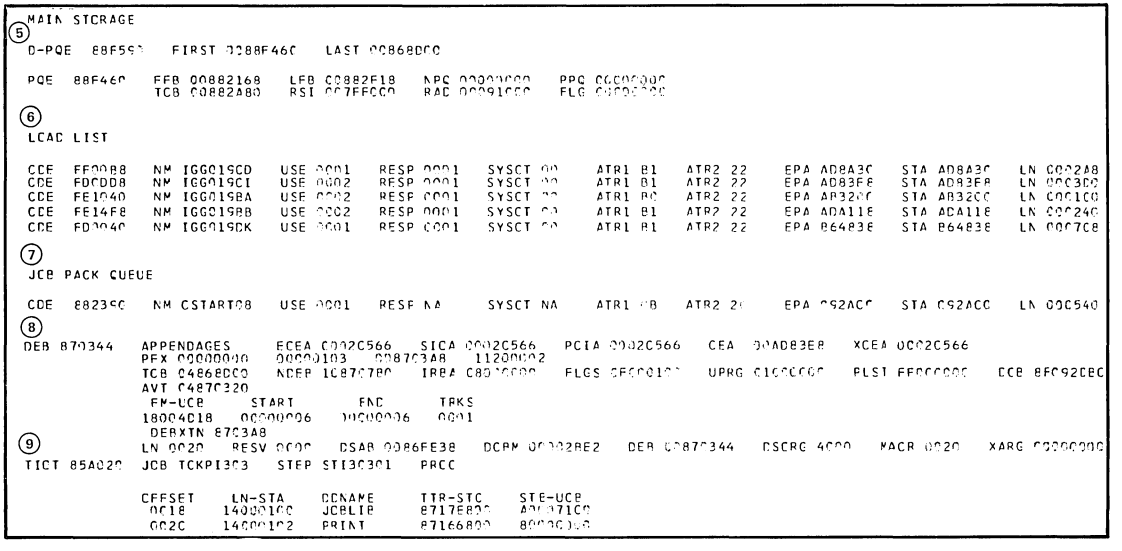


Figure 17. Task-Related Control Blocks (Part 3 of 10)

(5) MAIN STORAGE -- Partition queue for the region.

D-PQE hhhhhh -- Dummy partition queue element (address minus eight bytes, followed by formatted fields).

FIRST hhhhhhhh -- Address of first PQE.
LAST hhhhhhhh -- Address of last PQE.

PQE hhhhhh -- Partition queue element (PQE).*

(6) LOAD LIST -- List of load modules requested by LOAD macro instructions. Each line of print represents one load module, and includes fields from the CDE (contents directory entry), LLE (load list element), and XTLST (extent list).

CDE hhhhhh -- Address of CDE
NM cccccccc -- Load module name (CDNAME field of CDE).
USE hhhh -- Module use count (CDUSE field of CDE).
RESP hh -- Module responsibility count (LLECOUNT field of LLE).
SYSCT hh -- System responsibility count (LLESYSCT field of LLE).
ATR1 hh -- Module attribute flags (CDATTR field of CDE).
ATR2 hh -- Module attribute flags (CDATTR2 field of CDE).
EPA hhhhhh -- Module entry point address (CDENTPT field of CDE).
STA hhhhhh -- Module storage address (XTLMSBAD field of XTLST).
LN hhhhhh -- Module length in bytes (XTLMSBLN field of XTLST).

For a complete description of each field, refer to the appropriate data area description in Volume 2 or Volume 3.

(7) JOB PACK QUEUE -- List of load modules in the job pack area. Each line of print represents one load module and includes the fields described above for modules on the load list. (The RESP and SYSCT fields are not applicable, and are printed as "RESP NA" and "SYSCT NA.")

(8) DEB hhhhhh -- Data extent block (DEB).[1]

(9) TIOT hhhhhh -- Task I/O table (TIOT) associated with the TCB.[1]

---

[1]Data area acronym and address, followed by formatted fields. For a description of each field, refer to the appropriate data area description in Volume 2 or Volume 3. (Note that all request blocks are described under the heading "RB".)

```
FOR DEB AT LOCATION 00A90B94

(10)
-DCB-      (PSAM)                              AT LOCATION 00A92E08
-----

+10      002022DE 03003001 00004000 00000001 04000001 90000000 003C0020 00A90B94
+30      92EA6DE0 00EAA770 08000001 00001008 30040048 41084E58 01C2D5E0 00C2D5E0
+50      00001008 00000001

(11)
-IOB-      (SAM,BPAM-NORMAL SCHEDULING)     AT LOCATION  00084E60
-----

-8       41084E58 48000000
+0       FLAG     0200      SENS     0000      ECBPT    7E084EE0 CSW      000B4EA8 0C000000
+10      START    40084E88 CCWPT    00A92E08 RESTR    00000000 VARIABLE 00000000

  FOR DEB AT LOCATION 00A90B94

(13)
-UCB-      (DIRECT ACCESS TYPE)             AT LOCATION  00AA0B70
-----

-8       LOCK     00000000 IOQ      00000000
+0       FLGS     8100      IO      FF       STATUS   A9       CHAN     3FFF0000 DEVFLAG  00000000
+C       UNITNAME C0E5C9D6 DEVTYPE  30502009 EXTPT    00000000
+18      00000000 40404040 4040B601 00000000 00000000 00000000
```

Figure 17. Task-Related Control Blocks (Part 4 of 10)

(10) DCB -- Data control block associated with each DEB. The DCB is formatted only if the TCB was an abended TCB.

(11) IOB -- I/O control block chain associated with each DEB. The IOB is formatted only if the DCB format is successful and the IOBs could be found.

ICB -- Interruption control block replaces the IOB for chained scheduling.

LCB -- Line control block replaces IOB for TCAM access method.

(12) XDBA -- IOS EXCP debugging area pointed to by the TCB. The XDBA is formatted only if the task was an abended TCB and the XDBA was available. No example of this information is shown in the corresponding figure.

(13) UCB -- Unit control block chain associated with each DEB. The UCB is formatted only if the task was an abended TCB and the UCBs were available.

```
1EAVTRT2 - UNRECOVERABLE ABEND FAILURE                    MODULE SVCDUMP   DATE 01/21/77  TIME 09.02.05  PAGE 0013

-RTCT----------------------------------AT LOCATION  00FE7060
 ----
+0  (14)  NAME     RTCT     SYSABEND 2FC0BE60 SYSUDUMP 0F40BE00 SYSMDUMP FC000000
+10       RESV     0000     SDMPASID 0003     MEMTECB  80AB85F0 ASCBP    00000000 RECECB 80AE8150 LOGRECB 00FE0000

                  SDUMP ENTRIES

+24       DSNAME 00   FLAGS E0 DEBCBP 0007B760 DEVICE 30502009
+30       DSNAME 01   FLAGS 20 DEBCBP 0007B796 DEVICE 30502009
+3C       DSNAME 02   FLAGS 20 DEBCBP 0007B700 DEVICE 30502009
+48       DSNAME       FLAGS 00 DEBCBP 00000000 DEVICE 00000000
+54       DSNAME       FLAGS 00 DEBCBP 00000000 DEVICE 00000000
+60       DSNAME       FLAGS 00 DEBCBP 00000000 DEVICE 00000000
+6C       DSNAME       FLAGS 00 DEBCBP 00000000 DEVICE 00000000
+78       DSNAME       FLAGS 00 DEBCBP 00000000 DEVICE 00000000
+84       DSNAME       FLAGS 00 DEBCBP 00000000 DEVICE 00000000
+90       DSNAME       FLAGS 00 DEBCBP 00000000 DEVICE 00000000

+9C       SDPAKML  80BE79C0 FORMAT   00000000 MEMLOCK  00000001 MEMSRB   00FFE910
+AC       RTMTEST  00000000 RECFLG   0000     FRIDSFGN 002A     SDSW     00FE5108
+B8       TAPEDCBS 00000000 00000000 00000000 00000000 00000000
+CC                00000000 00000000 00000000 00000000
+DC       SDMPWA   00EC6E40 ERROR ID 00000000 00000000 00000000
+EC       SADPTION 2FC0BE00 SUDPTION 0F40BF00 SMDPTION FC000000 SDUPTION 00000000
+FC       ASDMPOPT 0000     SDMPINFO 010200   RESV     00000000 000000   SDMPFLG  0000     RESV     0000

                  SDUMP MULTIPLE ADDRESS SPACE FLAGS

+10C      ASID     0003     FLAGS    02       RESV     00
+110      ASID     0000     FLAGS    00       RESV     00
+114      ASID     0000     FLAGS    00       RESV     00
+118      ASID     0000     FLAGS    00       RESV     00
+11C      ASID     0000     FLAGS    00       RESV     00
+120      ASID     0000     FLAGS    00       RESV     00
+124      ASID     0000     FLAGS    00       RESV     00
+128      ASID     0000     FLAGS    00       RESV     00
+12C      ASID     0000     FLAGS    00       RESV     00
+130      ASID     0000     FLAGS    00       RESV     00
+134      ASID     0000     FLAGS    00       RESV     00
+138      ASID     0000     FLAGS    00       RESV     00
+13C      ASID     0000     FLAGS    00       RESV     00
+140      ASID     0000     FLAGS    00       RESV     00
+144      ASID     0000     FLAGS    00       RESV     00
+148      ASID     0000     FLAGS    00       RESV     00

+14C      MRMQ     00000000 RESERVED 00000000 00000000
```

(14) RTCT -- Recovery termination control table is formatted with the first TCB of the current address space on the CPU which took the dump.

Figure 17. Task-Related Control Blocks (Part 5 of 10)

```
(15) -RTM2WA-----------------------------------AT LOCATION  00AA1AC8

     +0      D9E3D4F2 00AA1AC8 FF0003C8 00039048 00AA1EB8 00ABE5F0 00FED658 000C1000
     +20     00000000 00000000 00000000 00000000 00AA1EB8 00ABE5F0 00FE7060

     -EED TYPE1 REGS AND PSW
     +3C     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
     +5C     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
     +7C     00000000 00000000 00000000 00000000
     -----

     +8C     00000000 00000000 00000000

     -EED TYPE3 MACHINE CHECK
     +98     00000000 00000000 00000000 00000000 00000000 00000000 00000000
     -----

     +B4     00000000 00000000 00000000 00000000 00000000 00000001 00000000 00000000
     +D4     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
     +F4     00000000 00000000 00000000

     -SNPPARMS
     +100    00000000 00000000 00000000 00000000 00000000

     -DUMP STORAGE RANGES
     +114    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
     -----

     +134    00000000 00000000 00000000 00000000 00AA1C1C 00AA1C20 00AA1C24 00AA1C28
     +154    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Figure 17. Task-Related Control Blocks (Part 6 of 10)

(15) RTM2WA -- RTM2 work area chain pointed to by the TCB. The RTM2WA is only formatted when the work areas are available.

```
-SUBFUNCTION REGISTER SAVE AREA
+174    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+194    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+1B4    00000000 00000000
-----

+1BC    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+1DC    00000000 00000000 00000000 00000000 00BE77FC 00000000

-RECURSION REGISTERS
+1F4    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+214    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

-SAVE AREA FOR VTRT2
+234    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+254    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+274    00000000 00000000

-SAVE AREA FOR TAS2/TAS3
+27C    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+29C    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+2BC    00000000 00000000
-----

+2C4    00AA1D90

-RMPL
+2C8    00000000 00000000 00000000 00000000 00AA1DA8 00000000
```

Figure 17. Task-Related Control Blocks (Part 7 of 10)

```
-WORK AREA FOR RESOURCE MANAGER
+2E0     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+300     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

-SAVE AREA FOR RESOURCE MANAGER
+320     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+340     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+360     00000000 00000000

+368     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+388     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+3A8     00000000 00000000 00000000 00000000 00000000

                              RTM2WA SUMMARY
                              --------------

+1C      COMPLETION CODE           000C1000
+8C      ABENDING PROGRAM NAME
+94      ABENDING PROGRAM ADDR     00000000

+3C      REGS AT TIME OF ERROR     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   (0-7)
+5C                                00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   (8-F)
+7C      EC PSW AT TIME OF ERROR   00000000 00000000 00000000 00000000
+DC      SDWACOMP                  00000000

+E8      RETURN CODE FROM RECOVERY ROUTINE-00,CONTINUE WITH TERMINATION-IMPLIES PERCOLATION
+E0      RETRY ADDR RETURNED FROM RECOVERY EXIT   00000000
+E4      RB ADDR FOR RETRY                        00000000
```

Figure 17. Task-Related Control Blocks (Part 8 of 10)

```
+C      CVT  ADDR                      00039048
+38     RTCT ADDR                      00FE7060
+C8     SCB  ADDR                      00000001
+D4     SDWA ADDR                      00000000
+14     SVRB ADDR                      00ABE5F0
+16C    PREV RTM2WA FOR THE TASK       00000000
+170    PREV RTM2WA FOR RECURSION      00000000

+B8     ASID OF ERROR IF CROSS MEMORY ABTERM       0000
+36C    ERROR ASID                 0000

+37C    CURRENT TRACE ENTRY FOR SAVED TRACE TABLE  00000000
+380    FIRST TRACE ENTRY FOR SAVED TRACE TABLE    00000000
+384    LAST TRACE ENTRY FOR SAVED TRACE TABLE     00000000

                              RTM2WA BIT FLAG SUMMARY
                              -----------------------

+2C8    TASK TERMINATION


                               ESA BIT FLAG SUMMARY (RTM1 RELATED INFORMATION) FROM SVRB AT 00ABE5F0          00ABE5F0
                               ------------------------------------------------------------

+89     RECURSION DURING INITIALIZATION
```

Figure 17. Task-Related Control Blocks (Part 9 of 10)

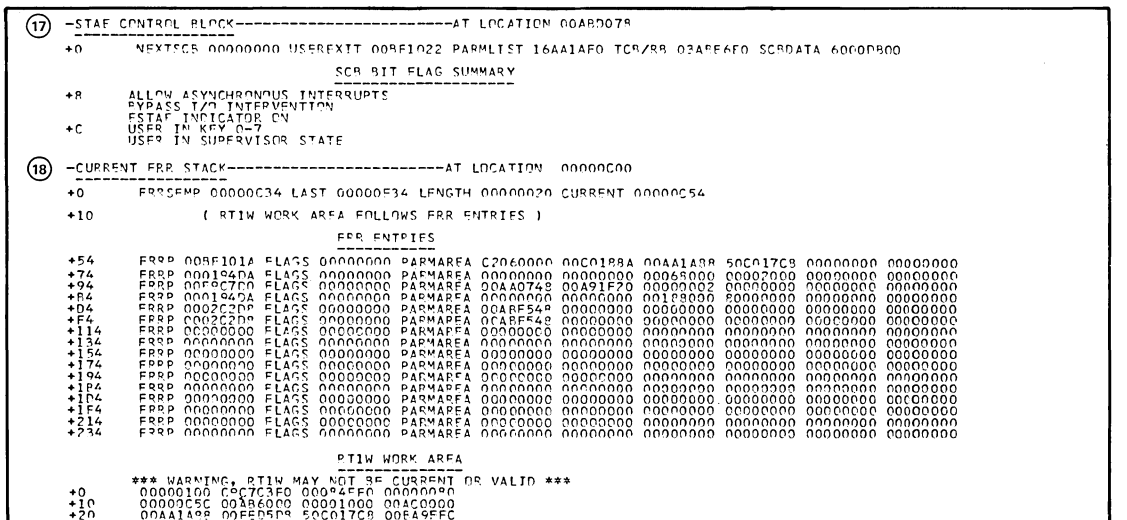


Figure 17. Task-Related Control Blocks (Part 10 of 10)

(16) EED -- Extended error descriptor chain pointed to by the TCB. The EED is only formatted if the EEDs are available. No example of this information is shown in the corresponding figure.

(17) SCB -- STAE control block chain pointed to by the TCB. The SCB is only formatted if the TCB is an abended TCB and the SCBs are available.

(18) FRRs -- Functional recovery routine stack (which has the RT1 work area embedded in it) is associated with the TCB. The FRRs are only formatted for the current task if it is holding the local lock. (This block is mutually exclusive with the IHSA.)

(19) IHSA -- Interruption handler save area (which nas the RT1 work area embedded in it) is associated with the TCB. It is formatted only for task interrupted or suspended SRB while holding the local lock. (This control block is mutually exclusive with the FRRs.) No example of this information is shown in the corresponding figure.

PSW, Registers, and Virtual Storage

Figure 18 is an example of the PSW, registers, and virtual storage printed for a specific address space. Contents are as follows:

(1) CURRENT PSW hhhhhhhh hhhhhhhh -- Current program status word (current when the dump was taken).[1]

(2) Register contents in hexadecimal format:[1]

GPRS 0-7, GPRS 8-F -- General purpose registers.
CTRS 0-7, CTRS 8-F -- Control registers.
FPRS 0-2, FPRS 4-6 -- Floating-point registers.

(3) Virtual storage (address space private area). Each line of print includes:

- The virtual address of the first printed word of storage.
- The storage protection key.
- Eight words of storage in hexadecimal format.
- The same eight words in EBCDIC format, with periods representing characters other than blank, A-Z, and 0-9.

[1]Note: If you do not perform a STORE STATUS operation, only the general purpose registers appear.

```
                                                    MODULE RTMSD350  DATE 08/16/73  TIME 22.28.05  PAGE 0027
(1)
CURRENT PSW  070C1000  001775A0
(2)
GPRS 0-7     00177560     90177540     00000040     00182FF8          00FBBC80     00FB4438     00FB2458     FD000000
GPRS 8-F     00FB4D88     00000000     00000000     00FBD438          4017750A     001775F8     0001BF48     00177558

CTRS 0-7     C080EC40     0F1B0C00     FFFFFFFF     FFFFFFFF          00000000     00000000     00000000     00000000
CTRS 8-F     00000000     00000000     00000000     00000000          00000000     00000000     C6000000     00000200

FPRS 0-2     00000000 00000000     00000000 00000000
FPRS 4-6     00000000 00000000     00000000 00000000
(3)
180000 THROUGH 182000 COULD NOT BE PRINTED
       STORAGE KEY IS 84 FOR STORAGE BEGINNING AT 18200
182000 84 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *.............................*
182020 TO NEXT LINE ADDRESS SAME AS ABOVE
       STORAGE KEY IS 86 FOR STORAGE BEGINNING AT 182800
```

Figure 18. PSW, Registers, and Virtual Storage (Example)

```
                                                             MODULE AMDSADMP  DATE            TIME 00.40.18  PAGE 1015
(1)                                      * * * * T C B   S U M M A R Y * * * *
JOB *MASTER*    ASID 0001    ASCB 0001FB00    FWDP 00FF0640    BWDP 00020698    PAGE 0982
 (2) TCB AT 01FCD8    CMP 00000000    NTC 00000000    OTC 00000000    LTC 0088CB98    TCB 0001FFC0    BACK 00000000    PAGE 0983
     TCB AT 01FFC0    CMP 00000000    NTC 000201C8    OTC 0001FCD8    LTC 00000000    TCB 000201C8    BACK 0001FCD8    PAGE 0984
     TCB AT 0201C8    CMP 00000000    NTC 00000000    OTC 0001FCD8    LTC 00000000    TCB 0088CB98    BACK 0001FFC0    PAGE 0985
     TCB AT 88CB98    CMP 00000000    NTC 0001FFC0    OTC 0001FCD8    LTC 0088C790    TCB 0088C790    BACK 000201C8    PAGE 0986
     TCB AT 88C790    CMP 00000000    NTC 00000000    OTC 0088CB98    LTC 00888BF8    TCB 00889EB8    BACK 0088CB98    PAGE 0987
     TCB AT 889EB8    CMP 00000000    NTC 00000000    OTC 0088C790    LTC 00000000    TCB 00889B10    BACK 0088C790    PAGE 0988
     TCB AT 889B10    CMP 00000000    NTC 00889EB8    OTC 0088C790    LTC 00000000    TCB 008899A0    BACK 00889EB8    PAGE 0989
     TCB AT 8899A0    CMP 00000000    NTC 00889B10    OTC 0088C790    LTC 00000000    TCB 00889768    BACK 00889B10    PAGE 0990
     TCB AT 889768    CMP 00000000    NTC 008899A0    OTC 0088C790    LTC 00000000    TCB 00889530    BACK 008899A0    PAGE 0991
     TCB AT 889530    CMP 00000000    NTC 00889768    OTC 0088C790    LTC 00000000    TCB 0088BBF8    BACK 00889768    PAGE 0992
     TCB AT 88BBF8    CMP 00000000    NTC 00889530    OTC 0088C790    LTC 00000000    TCB 00000000    BACK 00889530    PAGE 0993

JOB HASP        ASID 0003    ASCB 00FF2840    FWDP 00020698    BWDP 00000000    PAGE 0999
     TCB AT 88E2F0    CMP 00000000    NTC 00000000    OTC 00000000    LTC 0088D750    TCB 0088D988    BACK 00000000    PAGE 1000
     TCB AT 88D988    CMP 00000000    NTC 00000000    OTC 0088E2F0    LTC 00000000    TCB 0088D750    BACK 0088E2F0    PAGE 1001
     TCB AT 88D750    CMP 00000000    NTC 0088D988    OTC 0088E2F0    LTC 00888AAC    TCB 00888AA0    BACK 0088D988    PAGE 1002
     TCB AT 888AAC    CMP 00000000    NTC 00000000    OTC 0088D750    LTC 0087FC20    TCB 00888930    BACK 0088D750    PAGE 1003
     TCB AT 888930    CMP 00000000    NTC 00000000    OTC 00888AAC    LTC 00000000    TCB 008886B8    BACK 00888AAC    PAGE 1004
     TCB AT 8886B8    CMP 00000000    NTC 00888930    OTC 00888AAC    LTC 00000000    TCB 0087FEB8    BACK 00888930    PAGE 1005
     TCB AT 87FEB8    CMP 00000000    NTC 008886B8    OTC 00888AA0    LTC 00000000    TCB 0087FC20    BACK 008886B8    PAGE 1006
     TCB AT 87FC20    CMP 00000000    NTC 0087FEB8    OTC 00888AA0    LTC 00000000    TCB 00000000    BACK 0087FEB8    PAGE 1007

JOB TCKPI303    ASID 0004    ASCB 00FF0640    FWDP 00000000    BWDP 0001FB00    PAGE 1008
     TCB AT 88E2F0    CMP 00000000    NTC 00000000    OTC 00000000    LTC 0088D750    TCB 0088D988    BACK 00000000    PAGE 1009
     TCB AT 88D988    CMP 00000000    NTC 00000000    OTC 0088E2F0    LTC 00000000    TCB 0088D750    BACK 0088E2FC    PAGE 1010
     TCB AT 88D750    CMP 00000000    NTC 0088D988    OTC 0088E2F0    LTC 00882A80    TCB 00882A80    BACK 0088D988    PAGE 1011
     TCB AT 882A80    CMP 00000000    NTC 00000000    OTC 0088D750    LTC 00868D00    TCB 00868D00    BACK 0088D750    PAGE 1012
```

Figure 19. TCB Summary (Example)

TCB Summary

Figure 19 is an example of a TCB summary. Contents are as follows:

(1) Address space information for each address space (requested by a FORMAT control statement), for selected address spaces (requested by a PRINT JOBNAME= control statement), or for the current address space (requested by a PRINT CURRENT control statement).

JOB cccccccc -- name of the job associated with the address space. The following names identify system address spaces:

MASTER -- Master scheduler
HASP -- JES2
JES3 -- JES3

ASID hhhh -- address space identifier.

ASCB hhhhhhhh -- address of address space control block.

FDWP hhhhhhhh -- ASCBFDWP field; address of next ASCB on the ASCB ready queue.

BWDP hhhhhhhh -- ASCBBWDP field; address of previous ASCB on the ASCB ready queue.

PAGE dddd -- number of printed dump output page that contains a formatted dump of the ASCB.

(2) Task control block information (limited to the current address space in the case of an SVC dump). Each line of print represents a single TCB, and includes the following information:

TCB AT hhhhhh -- Address of the task control block (TCB).

CMP hhhhhhhh -- TCBCMP field; task completion code (three low-order bytes).

NTC hhhhhhhh -- TCBNTC field; address of the TCB for the task that was previously attached by this task's originating task, or zero if no such task exists. (Example: If task A attaches task B and then task C, the TCBNTC field in task C's TCB points to task B's TCB, and the TCBNTC field in task B's TCB is zero.)

OTC hhhhhhhh -- TCBOTC field; address of the TCB for the originating task, or zero if this is the first task in the address space.

LTC hhhhhhhh -- TCBLTC field; address of the TCB for this task's last-created subtask, or zero if this task has no subtasks.

TCB hhhhhhhh -- TCBTCB field; address of the next TCB on the TCB ready queue, or zero if this is the last TCB on the queue.

BACK hhhhhhhh -- address of the previous TCB on the TCB ready queue, or zero if this is the first TCB on the queue.

PAGE dddd -- number of printed dump output page that contains a formatted dump of the TCB.

```
                                                   MODULE AMDSADMP  DATE 08/05/76  TIME 05.00.02  PAGE 0001
TITLE FROM DUMP: 1-PC 077C2000 00AF5418 ASID=0006 CPU=41 JOB=IMSMPR09 L=0000 TEA=080030 ILC=4 PIC=11 FNT=002 LVL=6141

                                        * * * *  L O G D A T A  * * * *
(1)
---RECORD ENTRY TYPE - UNIT CHECK      SOURCE   VTAM OUTBOARD           MODEL- 0168       SERIAL NO.  06  15
   VS 2 REL.   03
                           DAY YEAR                        HH MM SS.TH                   JOB IDENTITY
                     DATE- 044   76                   TIME 11 33 12 31

  DEVICE TYPE                           NONE
  PRIMARY   CHANNEL UNIT ADDRESS        0003FF
  ALTERNATE CHANNEL UNIT ADDRESS        0003FF

                 CC    DA     FL     CT                       K     CA     US CS  CT
FAILING CCW      00  000000   00  00 0000               CSW   00  000000   00 00 0000

UNIT STATUS                                     CHANNEL STATUS
  ATTENTION             0                          PRGM-CTLD IRPT      0
  STATUS MODIFIER       0                          INCORRECT LENGTH    0
  CONTROL UNIT END      0                          PROGRAM CHECK       0
  BUSY                  0                          PROTECTION CHECK    0
  CHANNEL END           0                          CHAN DATA CHECK     0
  DEVICE END            0                          CHAN CTL CHECK      0
  UNIT CHECK            0                          I/F CTL CHECK       0
  UNIT EXCEPTION        0                          CHAINING CHECK      0

(2)                                                               DATE        TIME         CPU       CPU       RELEASE
                                                                DAY YR    HH MM SS.TH    SERIAL     ID        LEVEL
--- RECORD ENTRY SOURCE - SOFTWARE ---  TYPE   SOFTWARE(SVC 13) 044 76    12 38 38 21    060015    0168      VS 2 REL.  3

JOBNAME                        D24RBH1
ABENDING PROGRAM NAME          TEST          BC MODE PSW AT TIME OF ERROR                 BC MODE PSW OF LAST RB
NAME OF MODULE INVOLVED        N/A
NAME OF CSECT INVOLVED         N/A               FF040030D 5002E758                           FF850001 50C25754
FUNCTIONAL RECOVERY ROUTINE    N/A

REGS AT TIME OF ERROR

REGS 0-7       00000001   80201000   00FAD840   00000001   0088CA90   0088D528   00000000   00F89A10
REGS 8-15      00000001   00000004   000798A8   0088EDD0   8002E588   00079E50   00017CA6   6002E51C
```

Figure 20. LOGDATA Output (Example) (Part 1 of 2)

LOGDATA Output

Figure 20 is an example of the LOGDATA output for some hardware and software entries in the in-storage LOGREC buffer. Contents are as follows:

(1) Format of a unit check entry in the buffer. (For additional information on the unit check record contents see OS/VS2 System Programming Library: SYS1.LOGREC Error Recording.

(2) Format of a software entry in the buffer. (For additional information on software record contents see OS/VS2 System Programming Library: SYS1.LOGREC Error Recording.)

```
(3)
 1-PC 077C2000 00AF5418 ASID=0006 CPU=41 JOB=IMSMPR09 L=0000 TEA=MODULE AMDSADMP  DATE 08/05/76  TIME 05.00.02  PAGE 0008
 MCH RECORDS REQUESTED BUT NOT FOUND
 CCH RECORDS REQUESTED BUT NOT FOUND
 SFT RECORDS REQUESTED BUT NOT FOUND
 IPL RECORDS REQUESTED BUT NOT FOUND
 DDR RECORDS REQUESTED BUT NOT FOUND
 MIH RECORDS REQUESTED BUT NOT FOUND
 EOD RECORDS REQUESTED BUT NOT FOUND
(4)
 NUMBER OF OBR TYPE OF RECORDS READ WAS     6
 NUMBER OF MDR TYPE OF RECORDS READ WAS     6
```

Figure 20. LOGDATA Output (Example) (Part 2 of 2)

(3) Summary of record types that would have been formatted if they were in the in-storage buffer.

(4) Summary of records found in the in-storage LOGREC buffer. Not all record types are formatted, so the number of records listed in the summary may not equal the number of records formatted by LOGDATA.

LPAMAP OUTPUT (LINK PACK AREA MAP)

Figure 21 is an example of LPAMAP output. Each line of print represents one module, and provides the following information:

NAME (cccccccc)
load module name or alias.

EPA (hhhhhh)
entry point address.

STA (hhhhhh)
storage address (of first control section).

LNGTH (hhhhhh)
total length of all control sections (in bytes).

TYPE (MAJOR or MINOR)
type of CDE (contents directory entry) associated with the Module.

MODULE AMDSADMP DATE TIME 12.51.31 PAGE 0103

* * * * L I N K P A C K A R E A M A P * * * *

NAME	EPA	STA	LNGH	TYPE
IGG019BC	C16400	C16400	000148	MAJOR
IGG019CC	C3BCD8	C3BCD8	000328	MAJOR
IGG019CI	C3B3E8	C3B3E8	0003D0	MAJOR
IGG019CH	C0F088	C0F088	0000D0	MAJOR
IGG019BA	C16548	C16548	0001C0	MAJOR
IGG019BB	C3D030	C3D030	000240	MAJOR
HASPSSSM	FBD000	FBD000	004320	MAJOR
IEESB670	D56BC0	D56BC0	000440	MAJOR
IEESB605	F8B000	F8B000	002778	MAJOR
IEESB665	C7F008	C7F008	0003A0	MAJOR
IEAVAR00	F9F320	F9F000	001858	MAJOR
IEEVWAIT	E7F700	E7F700	000900	MAJOR
IEAVTMTC	F2CBB8	F2CBB8	000448	MAJOR
IEAVTRET	E4A7D0	E4A7B0	000850	MAJOR
IEAVTSDT	DB0E60	DB0E60	000028	MAJOR

Figure 21. Link Pack Area Map (Example)

PRINT CSA, PRINT NUCLEUS, AND PRINT SQA OUTPUT

Figures 22, 23 and 24 are examples of PRINT SQA, PRINT CSA, and PRINT NUCLEUS output. Each line of print includes:

- The virtual address of the first printed word of storage.
- The storage protection key.
- Eight words of storage in hexadecimal format.
- The same eight words in EBCDIC format, with periods representing characters other than blank, A-Z, and 0-9.

```
                                                                MODULE AMDSADMP  DATE 12/20/73  TIME 05.42.21  PAGE 0294

                                          * * * *  S Q A  P R I N T  * * * *

FC0000 THROUGH FC8000 COULD NOT BE PRINTED
       REAL ADDRESS FOR FC8000 IS 3BB000
       STORAGE KEY IS 00 FOR STORAGE BEGINNING AT FC8000
FC8000 00 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
FC8020 TO NEXT LINE ADDRESS SAME AS ABOVE
       STORAGE KEY IS 06 FOR STORAGE BEGINNING AT FC8800
FC8DA0 06 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
FC8DC0 06 00FCA860 00FC8DE0 00FCBFE0 F5003240   00000000 00000000 00000000 00000000   *............5.. ................*
FC8DE0 06 070E1201 00000000 00000000 00000000   00000000 00400000 00027C70 61E7512B   *..................... .......X..*
FC8E00 06 070E7000 00000000 00000000 00000000   00000000 00400000 00027C70 61E75147   *..................... .......X..*
FC8E20 06 070C3011 F0F3BD1A 00000000 00072F78   00072F78 00410001 00027308 61E751AB   *....03.....................X..*
FC8E40 06 070E1201 00000000 00000000 00000000   00000000 00400000 00027C70 61E75299   *..................... .......X..*
FC8E60 06 070E7000 00000000 00000000 00000000   00000000 00400000 00027C70 61E752B3   *..................... .......X..*
FC8E80 06 070E1202 00000000 00000000 00000000   00000000 00400000 00027C70 61E752ED   *..................... .......X..*
FC8EA0 06 070C2001 50F3C220 00000004 00000001   FFFB6598 00410001 00027308 61E752F1   *.....3B.....................X.1*
FC8EC0 06 070E7000 00000000 00000000 00000000   00000000 00400000 00027C70 61E75309   *..................... .......X..*
```

Figure 22. SQA Print (Example)

```
                                                                        MODULE AMDSADMP  DATE          TIME 00.40.18  PAGE 0001
TITLE FROM DUMP: 0C11 IN IEAOVL00 TEST CASE TCKPI303

                                                    * * * *  C S A   P R I N T  * * * *

890000 THROUGH 919000 COULD NOT BE PRINTED
       REAL ADDRESS FOR 919000 IS 19A000
       STORAGE KEY IS 1E FOR STORAGE BEGINNING AT 919000
919000 1E 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
919020 TO NEXT LINE ADDRESS SAME AS ABOVE
919040 1E 00000000 00000000 00000000 00000000   00000000 00000000 C9D6E340 08002800   *........................IOT ....*
919060 1E 47C0F59D 01138501 00000000 00000000   00000000 00000000 01139706 01138603   *..5.............................*
919080 1E 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
9190A0 TO NEXT LINE ADDRESS SAME AS ABOVE
9190C0 1E 00000000 00000000 00000000 00000000   00000100 00000000 00000000 00000000   *................................*
9190E0 1E 00000000 000001EC 28800050 0000F705   0001C101 80000000 00200000 00000000   *..............7...A.............*
919100 1E E2E3C44B 5C5C5C5C 5C5C5C5C 40404040   40404040 00000000 00000000 20C20085   *STD.........         .........B..*
919120 1E 01138502 0002C101 80000000 00200000   00000000 E2E3C44B 5C5C5C5C 5C5C5C5C   *......A.............STD.........*
919140 1E 40404040 40404040 00000000 00000000   0080005A 01138503 0003C101 80000000   *        ..................A.....*
919160 1E 00200000 00000000 E2E3C44B 5C5C5C5C   5C5C5C5C 40404040 40404040 00000000   *........STD.........        ....*
919180 1E 00000000 00C000A0 01138505 0004C101   80000000 00200000 00000000 E2E3C44B   *..............A.............STD.*
9191A0 1E 5C5C5C5C 5C5C5C5C 40404040 40404040   00000000 00000000 00900050 01138603   *........        ................*
9191C0 1E 0065C101 80000000 00000000 00000000   E2E3C44B 5C5C5C5C 5C5C5C5C 40404040   *..A.............STD.........    *
9191E0 1E 40404040 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *    ............................*
919200 1E 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
919220 1E 00000000 00000000 00000000 00000000   00000000 00000000 00000000 E3C3D2D7   *............................TCKP*
919240 1E C9F3F0F3 0001F393 0040002C C8F1F5F5   40404040 40404040 00E00000 00000000   *I303..3.. ..H155        ........*
919260 1E 00000000 0001F5A7 0040002C 00000000   0806E000 404040D4 F2F060F6 80180012   *......5.. ..........    M20.6....*
919280 1E 40404040 E2F3F6F0 80200012 40404040   E2F3F7F0 80200012 40404040 F1F1F3F0   *    S360....    S370....    1130*
9192A0 1E 80400012 40E2E8E2 E3C5D4F3 80800012   40404040 D3C9D5C5 1001000C D9D6E4E3   *. .. SYSTEM3....    LINE....ROUT*
9192C0 1E C5C3C4C5 10020010 D7C1E2E2 E6D6D9C4   0A08001C 404040D5 E4D4D9C4 10010015   *ECDE....PASSWORD....    NUMRD....*
```

Figure 23. CSA Print (Example)

```
                                                                MODULE AMDSADMP  DATE          TIME 00.40.18  PAGE 0001

TITLE FROM DUMP: 0C11 IN IEA0VL00 TEST CASE TCKPI303


                                          * * * *  N U C L E U S  P R I N T  * * * *


       REAL ADDRESS FOR 000000 IS 000000
       STORAGE KEY IS 06 FOR STORAGE BEGINNING AT 000000
000000 06 00080000 00007402 00007000 90007122   08000110 60000001 070C0000 00A7F4FA   *.....................................4.*
000020 06 071C0000 000923CC 040C1000 0022A61C   00000000 00000000 070E0000 00000000   *.....................................*
000040 06 00007158 0C000000 00007148 00021020   FAB10FFF 00FC7DC0 04080000 00066EC8   *.....................................H*
000060 06 04080000 00069188 00080000 00066000   00080000 0001B520 04080000 00065D98   *.....................................*
000080 06 00000000 00001004 00020001 00040001   00227A10 00000000 00000000 00000000   *.....................................*
0000A0 06 00000000 00000000 20000000 00000000   00000000 00000000 00000382 00000000   *.....................................*
0000C0 06 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *.....................................*
0000E0 TO NEXT LINE ADDRESS SAME AS ABOVE
000100 06 00000000 00000000 00000000 00000000   01007000 60006000 0100D000 60006000   *.....................................*
000120 06 07007000 60006000 02007000 B0000018   02007000 60005FE8 02007000 B0000090   *.....................Y...............*
000140 06 0200CFE8 60005F70 02007000 70006000   02007000 30006000 50D0A008 4190B26C   *...Y.....................................*
000160 06 5090A00C 5080A010 B611B57C 41A0B578   50A0C004 18AA5890 D00450A0 901058F0   *.....................................0*
000180 06 00868F16 00069E9C 0020AE76 0000702A   00000000 03007031 090079EE 80007484   *.....................................*
0001A0 06 000075E0 0B00766D 00000009 00007148   00007000 90007122 A0007726 00000004   *.....................................*
0001C0 06 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *.....................................*
0001E0 TO NEXT LINE ADDRESS SAME AS ABOVE
000200 06 D7E2C104 00000040 00FFFAD8 001FFAD8   00FFF710 001FF710 00868D00 00868D00   *PSA.... ...Q...Q..7...7.........*
000220 06 00FF0640 00FF0640 00000000 00000000   00000000 00000000 00000000 00000C54   *... ... ...........................*
000240 06 070C0000 00A7F4FA 00000000 00000000   040C0000 00A7F534 00000000 00000000   *.....4.................5.........*
000260 06 AD000264 00000000 AD00026C 00000000   00000000 00000000 00000000 00000000   *.....................................*
000280 06 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *.....................................*
0002A0 06 00000000 00000000 00000000 00000000   00000000 00000000 00000040 00FF0218   *.....................................*
0002C0 06 00FF523C 00000C00 00868D00 0088DBDC   00031D08 00032D08 00000000 400189AE   *.....................................*
0002E0 06 4004C138 00000000 00000001 00031DE8   4001BE1A 00000000 00000001 000449C8   *.A.............Y ...............H*
```

Figure 24. Nucleus Print (Example)

PRINT REAL OUTPUT (REAL STORAGE PRINT)

Figure 25 is an example of PRINT REAL output. Contents are as follows:

CURRENT PSW hhhhhhhh hhhhhhhh -- Current program status word (current when the dump was taken).[1]

Register contents in hexadecimal format:[1]

GPRS 0-7, GPRS 8-F -- General purpose registers.
CTRS 0-7, CTRS 8-F -- Control registers.
FPRS 0-2, FPRS 4-6 -- Floating-point registers.

Contents of real storage. Each line of print includes:

- The real (absolute) address of the first printed word of storage.[2]
- The storage protection key.
- Eight words of storage in hexadecimal format.
- The same eight words in EBCDIC format, with periods representing characters other than blank, A-Z, and 0-9.

[1]NOTE: If you do not perform a STORE STATUS operation, only the general purpose registers appear.

[2]An absolute address is a real address to which prefixing has been applied. Prefixing is described in GA22-7000; in effect, it simply interchanges the 4K bytes of storage at real address 0 with the 4K bytes of storage at the real address contained in the prefix register. Prefixing is applied to all references to real storage and to keys in storage, except for references to the PSA during a STORE STATUS operation, and except for references by a channel to extended logout locations, to indirect-data-address words, and to CCWs.

```
                                                MODULE AMDSADMP  DATE             TIME 00.40.18  PAGE 0001

TITLE FROM DUMP: 0C11 IN IEAOVL00 TEST CASE TCKPI303

                                  * * * *  R E A L   S T O R A G E   P R I N T  * * * *

GPRS 0-7      00868F16      00069E9C      0020AE76      0000702A            00000000      03007031      090079EE      800074B4
GPRS 8-F      000075E0      0B00766D      00000009      00007148            00007000      90007122      A0007726      00000004

ONLY GENERAL PURPOSE REGISTERS AVAILABLE

       STORAGE KEY IS 06 FOR STORAGE BEGINNING AT 000000
000000 06 00080000 00007000 06000130 6000002B   08000130 60000001 071D2000 000E073E   *.....................................*
000020 06 040C0000 000164B2 040C0000 00FDCA40   00000000 00000000 071D2000 000E073E   *.....................  .....................*
000040 06 00DAFEF0 0C000001 0007C4F8 00016460   FB3BFFFC 0002C034 040C0000 00017FE8   *...0......D8.....................Y*
000060 06 040C0000 000188E0 000C0000 0001819A   00000000 0001DC90 040C0000 00018114   *.....................................*
```

Figure 25. Real Storage Print (Example)

```
                                                                MODULE AMDSADMP  DATE 12/20/73  TIME 05.42.21  PAGE 0388
                                    * * * *  V I R T U A L   S T O R A G E   P R I N T  * * * *
(2)                                                          NUCLEUS
GPRS 0-7      00000168      00C58BB8      00C58BD8      0000FD01            00000000      140C0000      00000010      00FFF710
GPRS 8-F      8003FF68      4003D9FE      50036B66      4003DA92            A003FE14      00028160      00000000      0003AA30
ONLY GENERAL PURPOSE REGISTERS AVAILABLE
     (3a) REAL ADDRESS FOR 000000 IS 3F5000
          STORAGE KEY IS 06 FOR STORAGE BEGINNING AT 000000
000000 06 140C0000 0003E59E 00000000 4003C598   00028160 00000000 00C58BD8 000A5E34   *......V..... .E..........E.Q....*
000020 06 00028160 00C58BD8 040C0000 0003DAD8   00000168 00C58BB8 00C58BD8 0000FD01   *.....E.Q.......Q.....E...E.Q....*
000040 06 00C5C468 00000C5C 0003FB90 00026C40   7FC1FF68 4002A15C 0000000A 00C5C4F0   *.ED............ .A.. ........ED0*
000060 06 A003FE14 00000000 6003FE4A 47806506   00080000 00044680 040C0000 00037C3E   *.............................*

                                                             ASID 0001
     (3b) REAL ADDRESS FOR 070000 IS 3BE000
          STORAGE KEY IS 08 FOR STORAGE BEGINNING AT 070000
070000 08 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *.............................*
070020 TO NEXT LINE ADDRESS SAME AS ABOVE
0705A0 08 00000000 00000000 90ECD00C 05A058C0   001058B0 C094D201 B03AA014 98ECD00C   *.....................K.........*
0705C0 08 07FE0014 5D404040 058047F0 801AC9C5   C5E5C9D7 D340D9F0 F2F0F0F0 F0F961F2   *.....   ...0..IEEVIPL R0200009.2*
0705E0 08 F861F7F3 5800854A 45108022 0A0A1841   D20F4004 85A6D765 40004000 4130838C   *8.73.....K. ...P. - .....*
070600 08 41104004 D7021001 10014130 30005630   10005030 10009610 10005040 10049620   *.. .P..................... ....*

(3c)                                                         COMMON
C60000 THROUGH CE0000 COULD NOT BE PRINTED
       REAL ADDRESS FOR CE0000 IS 28B000
       STORAGE KEY IS 0E FOR STORAGE BEGINNING AT CE0000
CE0000 0E 00CFD400 000000C0 00000007 00CE057C   E7300000 00CE0020 00CE1000 00000000   *..M.............X.............*
CE0020 0E 01CE0080 21200015 40F0F6F5 4040E6F9   405C5C5C 5C5C5C4B 4B4B4B4B 4B4B4B4B   *........ 065  69 .............*
CE0040 0E 4B000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *.............................*
CE0060 0E 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *.............................*
CE0080 0E 01CE00E4 20200015 40F0F6F5 4040F6C1   405C5C5C 5C5C5C5C 5C4B4B4B 4B4B4B4B   *...U.... 065  6A .............*
CE00A0 0E 4B000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *.............................*
```

Figure 26. Virtual Storage Print (1 ple)

PRINT STORAGE OUTPUT (VIRTUAL STORAGE PRINT)

Figure 26 is an example of Print Storage output. Contents are as follows:

(1) CURRENT PSW hhhhhh hhhhhh -- Current program status word (current when the dump was taken).[1] No example of this information is shown in the corresponding figure.

(2) Register contents in hexadecimal format:[1]

GPRS 0-7, GPRS 8-F -- General purpose registers.
CTRS 0-7, CTRS 8-F -- Control registers.
FPRS 0-2, FPRS 4-6 -- Floating-point registers.

[1]If you have not performed a STORE STATUS operation, CURRENT PSW, CTRS, and FPRS do not appear. The formats of these fields are shown in Figure 18.

(3) Contents of virtual storage, identified as follows:

(3a) NUCLEUS -- Nucleus and system area.

(3b) ASID hhhh -- Private area of indicated address space.

(3c) COMMON -- Common area.

Each line of print includes:

- The virtual address of the first printed word of storage.
- The storage protection key.
- Eight words of storage in hexadecimal format.
- The same eight words in EBCDIC format, with periods representing characters other than blank, A - Z, and 0 - 9.

```
                                                        MODULE ABC       DATE          TIME 11.59.51  PAGE 0001

 TITLE FROM DUMP: VIRTUAL DUMP...ABC TEST"

                          * * * *   Q U E U E   C O N T R O L   B L O C K   P R I N T   * * * *

(1)
 MAJOR FF10A0    NAME SYSZEC16

    (2)MINOR FF28D0  STEP     ASID 0003  NAME PURGE
         (3)QEL FE70A0  TCB  D6E2F0  EXCLUSIVE  ASID 0003  SVRB D6DBE0


 MAJOR FDCC90    NAME SYSVSAM

      MINOR FFD580  SYSTEM   ASID 0000  NAME 0001B100D93C58D6                          *....R..O            *
            QEL FF2800  TCB  00B550  EXCLUSIVE  ASID 0002  SVRB 6FE454

      MINOR FF6048  SYSTEM   ASID 0000  NAME 1234567891012345
            QEL FFD3D8  LIST 00B550  RESERVE    ASID 00002             UCB  555555     STEP MUST COMPLETE

      MINOR FDCCA8  SYSTEM   ASID 0000  NAME 0001B000D93C58D6                          *....R..O            *
            QEL FFD550  TCB  00B550  RESERVE    ASID 0002  SVRB 6FE454

      MINOR FE9048  SYSTEM   ASID 0000  NAME SRMTST
            QEL FE0708  TCB  C56DB0  SHARED     ASID 000E  ECB  C4F71C
            QEL FDA038  TCB  C56DB0  SHARED     ASID 000F  ECB  C4F71C
            QEL FF3208  TCB  C56DB0  SHARED     ASID 000D  ECB  C4F71C
            QEL FD9DB8  TCB  C56DB0  SHARED     ASID 000C  ECB  C4F71C
```

Figure 27. Queue Control Block Print (Example)

QCBTRACE OUTPUT (QUEUE CONTROL BLOCK PRINT)

Figure 27 is an example of QCBTRACE output consisting of formatted QCBs (queue control blocks) and QELs (queue elements). For a full description of these control blocks, see "QCB" and "QEL" in Volume 3.

(1) Major QCB, representing a queue of system resources.

MAJOR hhhhhh -- address of major QCB.

NAME cccccccc -- queue name (MAJNAME field of QCB), established by the qname operand of an ENQ macro instruction.[1]

(2) Minor QCB, representing a specific resource.

MINOR hhhhhh -- address of minor QCB.

STEP, SYSTEM, or SYSTEMS -- operand of ENQ macro instruction (MINFLAGS field of QCB) indicating whether the resource is used by tasks of a single address space (STEP), tasks of a single system (SYSTEM), or tasks of several loosely-coupled systems (SYSTEMS).

ASID hhhh -- address space identifier (MINASID field of QCB), for STEP resources only (resources used by tasks of a single address space).

NAME cccc... -- resource name (MINNAME field of QCB), established by the rname operand of an ENQ macro instruction.[1]

(3) Queue element (QEL), representing a request for a resource.

QEL hhhhhh -- address of QEL.

TCB or LIST hhhhhh -- QELTCB field; address of TCB associated with this request, or address of QEL for another resource (indicating a request for a list of resources).

EXCLUSIVE or SHARED -- indicates a request for exclusive or shared use of the resource (QELQFLGS field).

ASID hhhh -- address space identifier (QELASID field) for address space associated with this request.

SVRB or ECB hhhhhh -- QELSVRB field; address of SVRB or ECB associated with this request.

UCB hhhhhh -- address of the UCB (printed only if the resource is a device requested by the RESERVE macro instruction).

[1] If the name contains printable characters, it is printed in hexadecimal format. On the right hand side of the page, the name is printed in EBCDIC format, with periods representing characters other than blank, A - Z, and 0 - 9.

```
SUMDMP06 NOALLPSA,CSA,SWA,TRT                                   MODULE IEAVTSDT  DATE 01/12/77  TIME 16.07.40  PAGE 0009
                                  * * * * SUMMARY DUMP PORTION OF SVC DUMP * * * *
                (1)                        (2)                        (3)                     (4)
   -----  TRACE TABLE          -----RECORD ID X'0004'   COPIED FROM DUMP ADDRESS 00FDBDC0 FOR A LENGTH OF X'3240' BYTES.
(5)
0004B540                            0004B578 0004B5C8   0004BA58 0004BE58 0004BEA8 0004BEE0  *         .. ... H...... ...  ..  *
0004B560   0004BF28 0004C0B8 0004C108 0004C230   0004C2D8                             *.. ...  ..A...B...BQ             *

*****  --A PARTIAL SUMMARY DUMP RECORD HAS BEEN READ--  ***** (6)

   -----  PSW AREA   -----         RECORD ID X'0031'    COPIED FROM DUMP ADDRESS 00F889A8 FOR A LENGTH OF X'1000' BYTES.

00F889A0                            5C5C5C5C 5C5C5C5C   5C5C5C5C 5C5C5C5C 5C5C5C5C 5C5C5C5C  *        ************************
00F889C0   5C5C5C5C 5C5C5C5C 5C5C5C5C 5C5C5C5C   5C5C5C5C 5C5C5C5C 5C5C5C5C 5C5C5C5C  ********************************
00F889E0   5C5C5C5C 5C5C5C5C 5C5C5C5C 5C5C5C5C   5C5C5C5C 5C5C5C5C 5C5C5C5C 5C5C5C5C  ********************************

00F88FE0   5C5C5C5C 5C5C5C5C 5C5C5C5C 5C5C5C5C   5C5C5C5C 5C5C5C5C 5C5C5C5C 5C5C5C5C  ********************************
00F89000   47F0F016 10C9C5C6 D1C4E2D5 C14040F7   F44BF0F1 F00090EC D00C05C0 1FFF58E0  *.00..IEFJDSNA  74.010... .. ... *
00F89020   101019EF 4780C07C 58B0E030 19BF4780   C07C58A0 E03419AF 4780C07C 58E0E004  *...... @.. .. .. @.. .. .. @.  .*

00F89900   BF273001 D2032008 113C1B22 BF273001   D5012004 C1984770 C078D503 2000C1A4  * ...K....... ...N...A .. .N...A *
00F89920   4780C086 41F00014 58E0D00C 980CD014   07FE1B22 BF273001 58202008 D5012004  *..  .0...  . . ..... .......N...*
00F89940   C19A4770 C0A4D503 2000C1A8 4780C0B2   41F00014 58E0D00C 980CD014 07FE5810  *A...  N...A ..  .0...  . . .....*
00F89960   00105810 11285820 101847F0 C17E1B11   BF173001 58101008 D5032008 10084770  *..........0A=.. .......N.......*
00F89980   C178BFFF 20104770 C0EC41F0 000858E0   D00C980C D01407FE 1B11BF17 300148F0  *A. ..... ..0...  . . ..... ....0*
00F899A0   100649F0 C19C4720                                                         *...0A ..
```

Figure 28. SUMSUMP Output (Example)

SUMDUMP OUTPUT (Summary Dump Portion of SVC Dump)

Figure 28 is an example of SUMDUMP output. The output for each summary dump record contains the following information:

(1) Record description -- The descriptions of all summary dump records are in Volume 3. SMDLR is the mapping macro for summary dump logical records. For information on how to use the records for debugging see OS/VS System Programming Library: Diagnostic Techniques.

(2) Record ID in hexadecimal.

(3) Address of the data contained in the record.

(4) Length of the data in the record.

(5) Unformatted dump of the data in the record.

(6) Message indicates that in the following record, portions of the data could not be obtained by summary dump.

In reality, the default formatter suppresses the character asterisk (*) which is X'5C', and provides the user with the beginning and ending addresses of the portion of data not obtainable by summary dump.

```
                                                    MODULE AMDSADMP  DATE 09/24/77  TIME 01.55.25  PAGE 0060

                                    * * * *  S Y S T E M  S U M M A R Y  * * * *

(1)
DUMP ADDRESS RANGES

          REAL       START      END
                     00000000   004FCFFF
                     004FD000   005FFFFF

          VIRTUAL    START      END        ASID
                     00940000   00940FFF   0003
                     00951000   00951FFF   0003
                     00953000   00953FFF   0003
                     00954000   00954FFF   0003
                     00955000   00958FFF   0003
                     00960000   0096BFFF   0003
                     0096C000   0096CFFF   0003
                     0096D000   0096EFFF   0003
                     0096F000   0096FFFF   0003
                     00958000   00958FFF   0004
                     00961000   0096BFFF   0004
                     0096C000   0096CFFF   0004
                     0096D000   0096EFFF   0004
                     0096F000   0096FFFF   0004
(2)
ACTIVE CPU LIST

          CPU 0000 - NORMAL MODE
(3)       CPU 0001 - NORMAL MODE
SCHEDULED SERVICES

          GLOBAL SERVICE MANAGER QUEUE

SRB    FD5010      +0  ID    E2D9C240    FLNK 00000000    ASCB 00022460    CPAFF 0000       PASID 0000       PTCB  00000000
                  +14  EP    00E3DB68    RMTR 00000000    PARM 000DFF50    SAVE  00000000   PKF   00         PRIOR 00
                  +26  RESV  0000        RESV 00000000


          LOCAL SERVICE MANAGER QUEUE

QUEUE IS EMPTY
```

Figure 29. System Summary (Example) (Part 1 of 3)

SUMMARY OUTPUT (SYSTEM SUMMARY)

Figure 29 is an example of SUMMARY output. Contents are as follows:

(1) DUMP ADDRESS RANGES -- Real and virtual storage contained in the dump (AMDPRDMP input).

REAL -- Identifies a range of real storage, defined by starting and ending addresses.

VIRTUAL -- Identifies a range of virtual storage, defined by starting and ending addresses, and address space identifier (ASID).

(2) ACTIVE CPU LIST -- Processors that were online when the dump was taken.

(3) SCHEDULED SERVICES -- Global and local SRB queues:

GLOBAL SERVICE MANAGER QUEUE -- Global SRBs awaiting transfer to the global service priority list.*

LOCAL SERVICE MANAGER QUEUE -- Local SRBs awaiting transfer to the local service priority list for the appropriate address space.*

*Each line of print represents one address space, and includes the following information:

ddddd SRBS QUEUED FOR ASCB hhhhhh -- Number (dddddd) of SRBs to be dispatched in an address space, which is identified by ASCB address (hhhhhh).

```
(4)
GLOBAL SERVICE PRIORITY LIST
QUEUE IS EMPTY

JOB SUMMARY
(5)     (5a)JOB *MASTER*   ASCB 022460   NEXT FE5870   PREV 000000   ASID 0001
            LOCAL SERVICE MANAGER QUEUE


SLIP STOP OF V CPU(0),OFFLINE 9/24/77 T69598                    MODULE AMDSADMP  DATE 09/24/77  TIME 01.55.25  PAGE 0061
QUEUE IS EMPTY
          (5b)LOCAL SERVICE PRIORITY LIST
QUEUE IS EMPTY
              (5c)TCB 022670   NEXT 022970   PREV 000000
                  TCB 022970   NEXT 022B78   PREV 022670
                  TCB 022B78   NEXT 96EA80   PREV 022970  - ACTIVE TASK ON CPU 0001
                  TCB 96EA80   NEXT 96E748   PREV 022B78
                  TCB 96E748   NEXT 969348   PREV 96EA80
                  TCB 969348   NEXT 9699E8   PREV 96E748
                  TCB 9699E8   NEXT 969138   PREV 969348
                  TCB 969138   NEXT 969790   PREV 9699E8
                  TCB 969790   NEXT 965DD0   PREV 969138
                  TCB 965DD0   NEXT 965B78   PREV 969790
                  TCB 965B78   NEXT 965778   PREV 965DD0
                  TCB 965778   NEXT 965040   PREV 965B78
                  TCB 965040   NEXT 000000   PREV 965778
           JOB JES2       ASCB FE5870   NEXT 000000   PREV 022460   ASID 0002
             LOCAL SERVICE MANAGER QUEUE
QUEUE IS EMPTY
```

Figure 29. System Summary (Example) (Part 2 of 3)

(4) GLOBAL SERVICE PRIORITY LIST -- Global SRB dispatching queues.

(5) JOB SUMMARY -- Summary of all currently defined address spaces, including the following information for each:

(5a) Job and address space identification.

JOB cccccccc -- Name of the job associated with the address space. The following names identify system address spaces:

MASTER -- Master scheduler.
JESP -- JES2.
JES3 -- JES3

ASCB hhhhhh -- Address of ASCB (address space control block).

NEXT hhhhhh -- ASCBFDWP field; address of the next ASCB on the ASCB ready queue.

PREV hhhhhh -- ASCBBWDP field; address of the previous ASCB on the ASCB ready queue.

ASID hhhh -- ASCBASID field; address space identifier.

(5b) LOCAL SERVICE PRIORITY LIST -- Local SRB dispatching queues.

NON-QUIESCEABLE QUEUE -- SRBs to be dispatched without regard to the status of the address space (active or quiesced).[1]

SYSTEM QUEUE -- SRBs to be dispatched only when the address space is active.[1]

(5c) TCB ready queue. Each line of print represents one TCB, and includes the following fields:

TCB hhhhhh -- Address of the TCB (task control block).

NEXT hhhhhh -- TCBTCB field; address of the next TCB on the TCB ready queue.

PREV hhhhhh -- TCBBACK field; address of the previous TCB on the TCB ready queue.

```
                                                                    MODULE AMDSADMP  DATE 09/24/77  TIME 01.55.25  PAGE 0062

          JOB INIT        ASCB FE7468   NEXT 000000   PREV FE5870   ASID 0004
                              - ADDRESS SPACE IN LONG WAIT
                              - ADDRESS SPACE SWAPPED OUT

               LOCAL SERVICE MANAGER QUEUE

QUEUE IS EMPTY

               LOCAL SERVICE PRIORITY LIST

QUEUE IS EMPTY

                    TCB 96E080   NEXT 96D3D0   PREV 000000
                    TCB 96D3D0   NEXT 96D178   PREV 96E080
                    TCB 96D178   NEXT 96C9F8   PREV 96D3D0
                    TCB 96C9F8   NEXT 000000   PREV 96D178

PROBLEM LIST (6)

          JOB *MASTER*  TCB   022970 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB *MASTER*  TCB   96EA80 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB *MASTER*  TCB   96E748 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB *MASTER*  TCB   969348 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB *MASTER*  TCB   9699E8 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB *MASTER*  TCB   969138 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB *MASTER*  TCB   969790 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB *MASTER*  TCB   965B78 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB *MASTER*  TCB   965778 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB *MASTER*  TCB   965040 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB JES2      TCB   96E080 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB JES2      TCB   96D178 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB JES2      TCB   96C670 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB JES2      TCB   96C398 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB JES2      TCB   96C1A0 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB JES2      TCB   9666A8 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB RMF       TCB   96E080 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB RMF       TCB   96D3D0 SET TEMP NON-DISPATCHABLE - FLGS4 00  FLGS5 01  SCNDY 00001000  DAR 00  STPCT 00
          JOB RMF       TCB   96D178 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 01  SCNDY 00001000  DAR 00  STPCT 00
          JOB RMF       TCB   96C980 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 01  SCNDY 00001000  DAR 00  STPCT 00
          JOB RMF       TCB   96C290 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 01  SCNDY 00001000  DAR 00  STPCT 00
          JOB INIT      TCB   96E080 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 00  SCNDY 00000000  DAR 00  STPCT 00
          JOB INIT      TCB   96D3D0 SET TEMP NON-DISPATCHABLE - FLGS4 00  FLGS5 01  SCNDY 00001000  DAR 00  STPCT 00
          JOB INIT      TCB   96D178 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 01  SCNDY 00001000  DAR 00  STPCT 00
          JOB INIT      TCB   96C9F8 SET TEMP NON-DISPATCHABLE - FLGS4 04  FLGS5 01  SCNDY 00001000  DAR 00  STPCT 00
          NO MACHINE CHECKS IN PROCESS
          NO ABENDS DETECTED FOR ASCBS
          NO NON-DISPATCHABLE ASCBS DETECTED
          NO ABENDS DETECTED IN PROGRESS FOR TCBS
```

Figure 29. System Summary (Example) (Part 3 of 3)

(6) PROBLEM LIST -- List of problems diagnosed by AMDPRDMP.

AMDPRDMP OUTPUT COMMENTS

AMDPRDMP output comments are of two types: EDIT output comments and general output comments.

EDIT Output Comments

The following comments may appear with the GTF data that AMDPRDMP formats. They are listed here in alphabetic order.

EDIT DELETED UPON USER'S REQUEST

A user exit routine requested that the current EDIT execution terminate by returning to EDIT with a return code of 16 to 20.

EDIT TERMINATED UPON USER'S REQUEST

A user exit routine requested EDIT termination by returning to EDIT with a return code of 24.

ERROR IN GTF BUFFER CONTROL BLOCK CHAIN

While attempting to locate the GTF trace buffers for an internal (dump) trace data set, AMDPRDMP could not obtain the trace buffers for one of the following reasons:

- The GTF control block chain was broken.
- The dump data set being processed is not from a MVS system.
- The block containing the addressed area was missing from the dump, perhaps because the program that produced the dump encountered an I/O error while attempting to write the block. Edit processing terminates.
- AMDPRDMP encountered an I/O error while attempting to read the record containing the area addressed by the pointer.

ERROR IN GTF BUFFER CONTINUING WITH NEXT BUFFER

While processing an internal (dump) trace data set, EDIT encountered a GTF trace record with a length outside the range of 4 to 284 bytes. EDIT continues processing with the next GTF buffer.

GTF NOT ACTIVE AT TIME OF DUMP

While processing an internal (dump) GTF trace data set, EDIT determined that GTF was not active when the dump was taken. EDIT processing terminates.

GTF OPTIONS IN EFFECT - option

The GTF OPTIONS IN EFFECT message precedes the main section of the output listing and indicates the conditions under which the GTF records were produced.

FORMAT=	SYS(Comprehensive) SYSM(Minimal) SYSP(Prompting)	Type of trace requested.
SVC=	ALL SELECTIVE NONE	SVC interruption tracing requested.
SIO=	ALL SELECTIVE NONE	SIO interruption tracing requested.
PI=	ALL SELECTIVE NONE	Program interruption tracing requested.
IO=	ALL SELECTIVE NONE SIO	IO interruption tracing requested.
EXT=	YES NO	External interruption tracing requested.
USR=	YES NO	User (GTRACE) tracing requested.
GTF=	YES NO	Tracing of the GTF task requested.
DSP=	YES NO	Dispatcher interruption tracing requested.
PCI=	YES NO	Program controlled I/O interruption trace requested.
RNIC	YES NO	VTAM remote network tracing requested.
RR=	YES NO	Recovery routine tracing requested.
SRM=	YES NO	System Resources Manager tracing requested.
USERTIME=	YES NO	Time-stamping of individual records requested.

IO ERROR ON ddname - CONTINUING

AMDPRDMP encountered an I/O error while attempting to read the external trace data set identified by 'ddname'. Fewer than three consecutive I/O errors have occurred for this data set, so EDIT continues processing, ignoring the block that caused the I/O error.

I/O ERROR ON ddname - EDIT PROCESSING TERMINATED

AMDPRDMP encountered three consecutive I/O errors while attempting to read the external trace data set identified by 'ddname'. EDIT processing terminates.

TRACE RECORD IS INVALID, DD ddname BLOCK NO xxxxxx - EDIT PROCESSING TERMINATED

While processing a GTF external trace data set identified by ddname, EDIT encountered a GTF trace record with a length outside the range of 4 to 284 bytes. xxxxxx is the number of the block containing the faulty record. EDIT processing terminates.

General Output Comments

The following comments may be printed within the body of a formatted dump. They are listed here in alphabetic order followed by a separate list of explanations. Each explanation is identified by a code; this code is listed with each comment to which the explanation applies.

Summary of Output Comments

Explanation Code	Message Text
153	A PARTIAL SUMMARY DUMP RECORD HAS BEEN READ
01	ALL POINTERS IN {LCCVT / PCCAVT} ARE UNAVAILABLE OR ZERO PCCAVT
02	ASCB COUNT EXCEEDS LIMIT OF xxxxx
03	ASCB FAILED VALIDITY CHECK
04	ASVT ADDRESS IS 0
05	ASVT CONTAINS DUPLICATE ENTRY FOR ASCB xxxxxx
04	ASXB ADDRESS IS 0
03	ASXB FAILED VALIDITY CHECK
06	AVERAGE NO OF BUFFERS PER FUNCTION - xxxxxx
07	CONTINUING FROM END OF TCB CHAIN
08	CPU ENTRY SKIPPED - {PSA AT ADDRESS xxxxxx NOT ON 4K BOUNDARY / UNABLE TO ACCESS PCCA-xxxxxx- / UNABLE TO ACCESS PSA-xxxxxx-}
04	CSD ADDRESS IS 0
03	CSD FAILED VALIDITY CHECK
09	CURRENT TASK IS DUMMY WAIT TASK
150	DATA MANAGEMENT FORMATTER TERMINATED WITH ABEND CODE XXX
04	DQE ADDRESS IS 0
10	END POINTER DOES NOT POINT TO LAST SRB ON CHAIN
11	ERROR ACCESSING CDE xxxxxx, CONTINUING
12	ERROR ACCESSING EXTENT LIST
01	ERROR ACCESSING {LCCAVT / PCCAVT}
13	ERROR FINDING ACTIVE LPA CHAIN POINTER, PROCESSING TERMINATED
14	ERROR FINDING ASCB xxxxxx
15	ERROR FINDING ASVT
16	ERROR FINDING ASXB
13	ERROR FINDING CDE QUEUE POINTER, PROCESSING TERMINATED
13	ERROR FINDING CDE xxxxxx, PROCESSING TERMINATED

17 ERROR FINDING CSD

18 ERROR FINDING EXTENT LIST, POINTER ZERO

14 ERROR FINDING {LCCA|PCCA} xxxxxx

19 ERROR FINDING {LCCAVT|PCCAVT}

13 ERROR FINDING MAJOR CDE, EXTENT LIST BYPASSED

20 ERROR FINDING MAJOR QCB xxxxxx--BACKWARD TRACING FOLLOWS

21 ERROR FINDING MINOR QCB xxxxxx

20 ERROR FINDING MINOR QCB xxxxxx--BACKWARD TRACING FOLLOWS

22 ERROR FINDING NUCLEUS BOUNDARIES - DEFAULT VALUES WILL BE USED

13 ERROR FINDING POINTER TO {FIRST|LAST} SRB

13 ERROR FINDING POINTER TO FIRST SRB, PROCESSING TERMINATED FOR THIS QUEUE

23 ERROR FINDING PSA FOR PCCA xxxxxx - PSA WILL NOT BE FORMATTED

23 ERROR FINDING PSA xxxxxx - PSA WILL NOT BE FORMATTED

21 ERROR FINDING QEL xxxxxx

20 ERROR FINDING QEL xxxxxx--BACKWARD TRACING FOLLOWS

13 ERROR FINDING QUEUE POINTER, NO SRBS FORMATTED

24 ERROR FINDING REGION/LSQA BOUNDARIES FOR TCB xxxxxx

25 ERROR FINDING SPL

14 ERROR FINDING SRB xxxxxx

26 ERROR FINDING UNPREFIXED STORAGE AT xxxxxx - FORMATTING BYPASSED

23 ERROR FINDING VIRTUAL {LCCA|PCCA} ADDRESS

23 ERROR FINDING VIRTUAL PSA ADDRESS, PSA WILL NOT BE FORMATTED

23 ERROR FINDING VIRTUAL PSA ADDRESS, UNPREFIXED STORAGE WILL NOT BE FORMATTED

27 ERROR FORMATTING CVT/EXT1/EXTa

28 ERROR FORMATTING DEB EXTENSION

27 ERROR FORMATTING TCB

29 ERROR IN DEB CHAIN

30 ERROR IN EXTENT LIST

27 ERROR IN JOB PACK QUEUE

27 ERROR IN LOAD LIST

31	ERROR IN SRB CHAIN, LAST EXPECTED SRB HAS NON-ZERO FORWARD POINTER, WILL PROCESS FIVE MORE FOLLOWING FORWARD POINTER
32	ERROR IN TCB CHAIN
33	ERROR IN TIOT
34	ERROR READING ABOVE TCB FROM DUMP
27	ERROR WHILE FORMATTING {ASCB / ASXB / SPL}
27	ERROR WHILE FORMATTING {CSD / LCCA / PCCA}
04	{EXT1 / EXT2} ADDRESS IS 0
35	FORMAT ERROR IN MAIN STORAGE BLOCKS
36	FORMAT ERROR READING {CDE / LPDE}
37	FORMATTING TERMINATED FOR THIS ADDRESS SPACE
38	FORMATTING TERMINATED
39	FORWARD POINTER IS ZERO, PROCESSING TERMINATED FOR THIS QUEUE
04	GDA ADDRESS IS 0
40	INFINITE LOOP IN ACTIVE LPA CHAIN, PROCESSING TERMINATED
158	IEAVTFRD IS UNABLE TO OBTAIN SUFFICIENT STORAGE TO RECONSTRUCT SUMMARY DUMP RECORDS
41	INFINITE LOOP IN DEB CHAIN
42	INFINITE LOOP IN JOB PACK QUEUE
42	INFINITE LOOP IN LOAD LIST
43	INFINITE LOOP IN PQES
43	INFINITE LOOP IN RB CHAIN
44	INFINITE LOOP IN TIOT
45	INVALID CPU ADDRESS, STORE-STATUS WILL NOT BE FORMATTED
149	INVALID EED TYPE ENCOUNTERED - EED BYPASSED AT LOCATION XXXXXXXX
46	INVALID MAJOR QCB CHAIN ASSUMED MINOR
47	INVALID TIOT
48	I/O ERROR READING BLOCK xxxxxx
151	IOS FORMATTER TERMINATED WITH ABEND CODE XXX
13	JOBNAME NOT AVAILABLE
03	LCCA FAILED VALIDITY CHECK
49	LINK PACK AREA QUEUE EMPTY

50	LOOP ASSUMED IN DQE CHAIN
51	LOOP CONTROL VALUE IS ZERO; LPAMAP FUNCTION TERMINATED
51	LOOP CONTROL VALUE IS ZERO; PROCESSING TERMINATED FOR ALL SRB QUEUES
52	LOOP IN SRB CHAIN, MORE THAN MAXIMUM NUMBER FOUND
53	MAXIMUM OF 5 SRBs WILL BE PROCESSED FROM END POINTER
54	NO CPUS COULD BE LISTED - ALL PCCAVT ENTRIES WERE 0
55	NO ELEMENTS ON LOAD LIST
30	NO EXTENT LIST
56	NO JOBS COULD BE LISTED
57	NO JOBS WERE FOUND
58	NO MAJOR QCBS
06	NO OF BLOCKS READ - xxxxxxxxxx
06	NO OF ENTRIES TO READ ROUTINE - xxxxxxxxxx
06	NO OF PERMANENT I/O ERRORS - xxxxxxxxxx
06	NO OF TIMES REQ ADDR WAS NOT IN A BUFFER - xxxxxxxxxx
06	NO OF TRACE RECORDS PROCESSED - xxxxxxxxxx
157	NO SUMMARY DUMP DATA READ BECAUSE THE {CVT / GDA} COULD NOT BE LOCATED
59	NOTHING IN JOB PACK
154	NORMAL END OF SUMMARY DUMP REACHED
60	ONLY GENERAL PURPOSE REGISTERS AVAILABLE
60	ONLY GENERAL PURPOSE REGISTERS ARE VALID
61	ONLY PART OF SQA WILL BE PRINTED
62	PCCA UNAVAILABLE, UNPREFIXED STORAGE WILL NOT BE FORMATTED
63	PCCAVT ADDRESS IS 0 - CPU LIST TERMINATED
03	PCCA FAILED VALIDITY CHECK
64	POINTER TO LAST SRB IS ZERO
65	POINTER TO LAST SRB IS 0 - POSSIBLE ERROR
66	POINTER TO NEXT TCB IS 0 - POSSIBLE ERROR IN TCB CHAIN
04	PQE ADDRESS IS 0
67	PREFIX VALUE UNAVAILABLE FOR IPL'ED CPU, PSA WILL NOT BE FORMATTED
68	PROCESSING TERMINATED
68	PROCESSING TERMINATED FOR THIS ADDRESS SPACE

68	PROCESSING TERMINATED FOR THIS CHAIN
69	PROCESSING TERMINATED FOR THIS SRB CHAIN, MORE THAN FIVE SRBS HAVE BEEN FOUND
70	PSA ADDRESS IS NOT ON A 4K BOUNDARY, FORMATTING BYPASSED
03	PSA FAILED VALIDITY CHECK
71	QUEUE IS EMPTY
72	RB FORMAT ERROR
73	REAL ADDRESS FOR FOLLOWING BLOCK IS xxxxxx
27	READ ERROR WHILE FORMATTING THE SVT
74	REGISTERS COULD NOT BE PRINTED
152	RTM FORMATTER TERMINATED WITH ABEND CODE XXX
03	SPL FAILED VALIDITY CHECK
04	SPQE ADDRESS IS 0
75	SRB CHAIN BROKEN
76	SRB COUNT EXCEEDS LIMIT OF xxxxx - QUEUE PROCESSING TERMINATED
77	STORAGE KEY IS yy FOR STORAGE BEGINNING AT xxxxxx
78	STORE STATUS DATA MAY BE INVALID
55	TASK HAS NO OPEN DATA SETS
79	TASK HAS NO TIOT
80	TASK HAS TERMINATED
02	TCB COUNT ExcEEDS LIMIT OF xxxxxx
155	THE END OF THE AVAILABLE SUMMARY DUMP DATA WAS REACHED BEFORE THE NORMAL END OF THE SUMMARY DUMP
156	THIS DUMP CONTAINS NO SUMMARY DUMP DATA
81	UNABLE TO ACCESS cccccccc - xxxxxx- [cccc...]
26	UNABLE TO ACCESS ASCBASXB - xxxxxxxx
81	UNABLE TO ACCESS ASVT ENTRIES
26	UNABLE TO ACCESS ASXBFTCB - xxxxxxxx WILL TRY
26	UNABLE TO ACCESS ASXBLTCB - xxxxxxxx
14	UNABLE TO ACCESS CURRENT TASK
82	UNABLE TO ACCESS CVT
83	UNABLE TO ACCESS CVT, FORMAT FUNCTION TERMINATED
83	UNABLE TO ACCESS CVT, LPA FUNCTION TERMINATED
84	UNABLE TO ACCESS CVT, MINIMUM FORMATTING WILL BE ATTEMPTED
85	UNABLE TO ACCESS CVT REST OF SUMMARY SKIPPED
81	UNABLE TO ACCESS CVTGDA - xxxxxx
81	UNABLE TO ACCESS DQEBLKAD - xxxxxx

81	UNABLE TO ACCESS DQEPTR - xxxxxx
81	UNABLE TO ACCESS GDA FIELD{CSAPQEP / SQASPQEP} - xxxxxx
86	UNABLE TO ACCESS POINTER TO NEXT TCB - xxxxxx
26	UNABLE TO ACCESS POINTER TO NEXT TCB - xxxxxxxx [CONTINUING FROM END OF TCB CHAIN]
81	UNABLE TO ACCESS PQEREGN - xxxxxx
81	UNABLE TO ACCESS PQESIZE - xxxxxx
87	UNABLE TO ACCESS PSA FOR IPL CPU - ENTRY MAY BE ASXBLTCB
81	UNABLE TO ACCESS SPQE FIELD APDQEPTR - xxxxxx
15	UNABLE TO ACCESS THE SVT
88	UNABLE TO DETERMINE BOUNDS OF PRIVATE AREA
89	UNABLE TO DETERMINE CPU ADDRESS, STORE-STATUS DATA WILL NOT BE FORMATTED
90	UNABLE TO FIND ASCB DISPATCHING CHAIN
91	UNABLE TO FORMAT ANY SRBS ON THE QUEUE
13	UNABLE TO FORMAT LCCA FOR PCCA xxxxxx
19	UNABLE TO FORMAT LCCA, PCCA, and PSA
92	UNABLE TO FORMAT PCCA AND PSA FOR LCCA xxxxxx
81	UNABLE TO FORMAT STORE-STATUS DATA FOR CPU xxxx
171	UNABLE TO LOAD AMDPRFAR-NO SRB PROCESSING IS DONE
93	UNABLE TO OBTAIN POINTER TO FIRST QCB INVALID
94	UNABLE TO SATISFY STORAGE REQUEST FOR EXIT MODULE xxxxxxxx MODULE xxxxxxxx
10	UNEXPECTED END OF SRB CHAIN BEFORE END POINTER REACHED
95	VIRTUAL{LCCA / PCCA} ADDRESS IS ZERO
04	VIRTUAL PSA ADDRESS IS ZERO, PSA WILL NOT BE FORMATTED
95	VIRTUAL PSA ADDRESS IS ZERO, UNPREFIXED STORAGE WILL NOT BE FORMATTED
96	WILL PROCESS FIVE SRBS BEGINNING WITH SRB POINTED TO BY LAST QUEUE POINTER
97	WILL USE ASCB DISPATCHING CHAIN INSTEAD
98	xxxxxx FAILED VALIDITY CHECK
04	xxxx ADDRESS IS 0 [-cccc...]
74 99	xxxxxx THROUGH xxxxxx COULD NOT BE PRINTED xxxxxx TO NEXT LINE ADDRESS SAME AS ABOVE
162	*** AN INVALID I/O REQUEST WAS MADE BY THE EREP EDITOR ***
100	*** CURRENT FRR STACK NOT AVAILABLE: UNABLE TO LOCATE PSA ***

101 *** DATA MANAGEMENT FORMATTING OMITTED, DUMP IS NOT FROM AN MVS SYSTEM ***

102 *** DATA MANAGEMENT FORMATTER TERMINATED - PARAMETER LIST DID NOT CONTAIN NECESSARY INFORMATION ***

103 *** DATA MANAGEMENT FORMATTER TERMINATED - TCB COULD NOT BE ACCESSED ***

104 *** DCB NOT FORMATTED - COULD NOT DETERMINE THE ACCESS METHOD USED ***

105 *** DCB NOT FORMATTED - DATA NOT AVAILABLE TO ACCESS IT ***

106 *** ERROR IN THE FIELDS USED TO DETERMINE THE NUMBER OF FRR ENTRIES ***

107 *** EXCPD(XDBA) NOT FORMATTED - DATA NOT AVAILABLE TO ACCESS IT ***

108 *** FOR THIS DEB, THERE ARE NO UCBS ***

109 *** FOR THIS DEB THERE IS NO DCB, THE CONTROL BLOCK POINTED TO BY THE DEB IS AN ACB ***

110 *** FORMATTING ROUTINE ERROR DURING FRRS FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ***

111 *** FORMATTING ROUTINE ERROR DURING IHSA FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ***

112 *** FORMATTING ROUTINE ERROR DURING SCB FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ***

113 *** FORMATTING SERVICE ROUTINE ERROR DURING DCB FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ***

114 *** FORMATTING SERVICE ROUTINE ERROR DURING DEB FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ***

115 *** FORMATTING SERVICE ROUTINE ERROR DURING EED FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ***

116 *** FORMATTING SERVICE ROUTINE ERROR DURING EXCPD FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ***

117 *** FORMATTING SERVICE ROUTINE ERROR DURING ICB FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ****

118 *** FORMATTING SERVICE ROUTINE ERROR DURING IOB FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ***

119 *** FORMATTING SERVICE ROUTINE ERROR DURING RTCT FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ***

120 *** FORMATTING SERVICE ROUTINE ERROR DURING RTM2WA FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ***

121 *** FORMATTING SERVICE ROUTINE ERROR DURING SDWA FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ***

122 *** FORMATTING SERVICE ROUTINE ERROR DURING UCB FORMATTING (POSSIBLE CAUSE: DATA NOT IN DUMP) ***

123	*** FRRS NOT FORMATTED - DATA NECESSARY TO LOCATE OR DETERMINE ITS VALIDITY NOT AVAILABLE ***
124	*** ICBS NOT FORMATTED - DATA NOT AVAILABLE TO ACCESS THEM ***
125	*** IHSA NOT FORMATTED - DATA NECESSARY TO LOCATE OR DETERMINE ITS VALIDITY NOT AVAILABLE ***
163	*** IN-STORAGE LOGREC BUFFER COULD NOT BE ACCESSED, POSSIBLE CAUSE (DATA NOT IN DUMP) ***
164	*** IN-STORAGE LOGREC BUFFER NOT FORMATTED, DATA NECESSARY TO LOCATE THE BUFFER UNAVAILABLE ***
165	*** IN-STORAGE LOGREC BUFFER NOT FORMATTED, EREP ENHANCEMENT SU NOT AVAILABLE ON SYSTEM ***
166	*** IN-STORAGE LOGREC BUFFER NOT FORMATTED, INVALID HEADER INFORMATION FOUND ***
167	*** IN-STORAGE LOGREC BUFFER NOT FORMATTED, PARAMETER LIST DID NOT CONTAIN NECESSARY INFORMATION ***
168	*** IN-STORAGE LOGREC BUFFER NOT FORMATTED, UNABLE TO DETERMINE VALID CPU ID ***
126	*** IOBS NOT FORMATTED - DATA NOT AVAILABLE TO ACCESS THEM ***
127	*** IOS FORMATTING OMITTED, DUMP IS NOT FROM AN MVS SYSTEM ***
128	*** IOS FORMATTER TERMINATED - PARAMETER LIST DID NOT CONTAIN NECESSARY INFORMATION ***
129	*** IOS FORMATTER TERMINATED - TCB COULD NOT BE ACCESSED ***
130	*** LCBS NOT FORMATTED - DATA NOT AVAILABLE TO ACCESS THEM ***
169	*** LOGDATA FORMATTING OMITTED, DUMP IS NOT FROM AN MVS SYSTEM ***
131	*** MORE THAN 25 EED AVAILABLE, POSSIBLE LOOP CONDITION, ONLY 25 WILL BE FORMATTED ***
132	*** MORE THAN 25 RTM2WA AVAILABLE, POSSIBLE LOOP CONDITION, ONLY 25 WILL BE FORMATTED ***
133	*** MORE THAN 25 SCB AVAILABLE, POSSIBLE LOOP CONDITION, ONLY 25 WILL BE FORMATTED ***
134	*** NO EEDS FORMATTED - DATA NOT AVAILABLE TO ACCESS THEM ***
135	*** NO RTM2WAS FORMATTED - DATA NOT AVAILABLE TO ACCESS THEM ***
136	*** NO SCBS FORMATTED - DATA NOT AVAILABLE TO ACCESS THEM ***
159	*** NOT ALL DEBS ON THE DEB CHAIN COULD BE ACCESSED ***
137	*** NOT ALL EEDS AVAILABLE COULD BE ACCESSED ***

161 *** NOT ALL ICBS ON THE ICB CHAIN COULD BE ACCESSED ***

160 *** NOT ALL IOBS ON THE IOB CHAIN COULD BE ACCESSED ***

138 *** NOT ALL RTM2WAS AVAILABLE COULD BE ACCESSED ***

139 *** NOT ALL SCBS AVAILABLE COULD BE ACCESSED ***

140 *** RTCT NOT FORMATTED - DATA NECESSARY TO LOCATE IT WAS UNAVAILABLE ***

141 *** RTM FORMATTER TERMINATED - PARAMETER LIST DID NOT CONTAIN NECESSARY INFORMATION ***

142 *** RTM FORMATTING OMITTED, COULD NOT ACCESS SUBITS ***

143 *** RTM FORMATTING OMITTED, DUMP IS NOT FROM AN MVS SYSTEM ***

170 *** THERE ARE NO VALID LOGREC RECORDS IN THE IN-STORAGE LOGREC BUFFER ***

144 *** UCB NOT FORMATTED - DATA NOT AVAILABLE TO ACCESS IT ***

145 *** UCB PROCESSING TERMINATED FOR THIS DEB - UNABLE TO DETERMINE ACCESS METHOD TYPE ***

146 *** UNKNOWN UCB TYPE FOR UCB AT LOCATION XXXXXXXX ***

147 *** UNKNOWN VALUE IN ASCBLOCK - RTM1 FORMATTING HALTED ***

148 *** WARNING, RT1W MAY NOT BE CURRENT OR VALID ***

Explanations of Output Comment Codes

01 While attempting to read the required pointers from the CCAVT, either an I/O error was encountered; one of the blocks containing a required pointer was missing from the dump; or one of the required pointers was zero.

02 The number of ASCBs/TCBs in the dump exceeds the maximum limit.

03 The EBCDIC name of the control block being referenced is invalid. Processing continues.

04 The specified address was zero in the dump data set.

05 The ASCB at address xxxxxx was found again in the ASVT. Since the ASCB was already processed, it will be skipped.

06 This message supplies an execution statistic for AMDPRDMP processing. AMDPRDMP terminates.

07 An error was encountered while attempting to read the forward pointer for the specified control block chain. An attempt will be made to process from the end of the chain using the backward pointer.

08 Because the problem described in the message text has occurred, a processor entry was passed over.

09 The ASCBASID field in the current ASCB is zero indicating that the system was in an enabled wait state with no task ready.

10 The pointer to the last control block points to a control block that does not have a zero link field.

11 While accessing the specified control block, either an I/O error was encountered while attempting to read the block that contained the data; or the block containing the required data was missing from the dump.

12 While attempting to read the length value and/or entry point address in the extent list, from the dump data set, either an I/O error was encountered; or the block containing the data was missing from the dump.

13 While attempting to read the specified pointer, job name, or data entry from the dump data set, either an I/O error was encountered; or the block containing the pointer, job name, or data entry was missing from the dump. Processing continues unless the message states otherwise.

14 While attempting to read the specified control block from the dump data set, either an I/O error was encountered; or the control block was missing from the dump. Processing continues.

15 An error occurred while trying to access the specified table. Either the table or the pointer to the table could not be read from the dump.

16 While attempting to read either the pointer to the specified control block or the control block itself from the dump data set, an I/O error was encountered; or the block containing the pointer or the control block was missing from the dump; or the pointer was zero. Processing continues.

17 While attempting to read the pointer to the specified control block, either an I/O error was encountered; or the block containing the control block was missing from the dump. Processing continues.

18 The pointer to the extent list for this data element was zero. Processing continues with the next data element on the queue.

19 I/O error was encountered while attempting to read the pointer to the specified control block or data area

20 The specified control block was not in the dump. Processing will be attempted from the end of the chain using the backward pointers.

21 The specified control block is not in the dump. Further processing of this portion of the structure is impossible.

22 The address of the top of the nucleus could not be determined from the dump. Default values of X'0' and X'20000' will be used as boundaries.

23 While attempting to read the specified address either an I/O error was encountered; or the block containing the address was missing from the dump. Processing continues.

24 While attempting to determine the region or LSQA boundaries for the family of TCBs attached to the job step TCB, one of the following items could not be extracted from the dump: A PQE pointer, or a pointer within a PQE could not be extracted from the dump; or the pointer to the SPQE for the LSQA for the TCB, the dummy PQE in the SPQE, the size of the LSQA, or the LSQA itself could not be extracted from the dump.

25 While attempting to read the specified storage, either an I/O error was encountered while attempting to read the block containing this storage; or the block containing this storage was missing from the dump. Processing continues.

26 While attempting to read the pointer to the specified control block, either an I/O error was encountered; or the block containing the control block was missing from the dump; or the pointer was zero. Processing continues.

27 While formatting the specified control block or control block extension, either an I/O error was encountered while attempting to read the block of storage containing the control block or control block extension; or the required block of storage was missing from the dump. Processing continues.

28 A field of the specified extent block extension could not be read from the dump. Processing continues with the next extent block.

29 While formatting the specified area, either an I/O error was encountered while attempting to read the block containing the required data or the chain pointer for the area addressed an area that was missing from the dump. Processing continues.

30 While formatting the load list or job pack area, a CDE was encountered that had a block extent list with a relocation factor of zero or greater than twenty-five; or the extent list pointer was zero. Processing continues with the next CDE.

31 While processing the specified control block chain, the control block pointed to by the preceding control block was found to bave a non-zero link field. A maximum of five more control blocks will be formatted following the link field pointer.

32 The same TCB was encountered twice on the TCB chain for a memory, indicating an infinite loop. Processing for this memory is terminated.

33 While formatting the specified area, either the required data was missing from the dump; or an I/O error was encountered while attempting to read the block containing the required data. Processing continues.

34 A read error occurred on at least one of the preceding fields of the TCB. Processing continues.

35 While formatting the specified control block, either the area addressed by a pointer in the control block was missing from the dump; or an I/O error was encountered while attempting to read a block containing required data.

36 A field in the specified entry associated with an RB could not be read from the dump. That entry will not be formatted.

37 Processing of the present address space is terminated.

38 All remaining data will not be formatted.

39 The link field of the SRB chain was zero.

40 While following the event chain for the specified area, more than the allowable number of events were found. Processing continues with the next verb.

41 While formatting the data extent blocks, AMDPRDMP found more then 200 DEBs chained to the TCB. The first 200 DEBs are printed. Processing for the current control statement is terminated, but processing of the dumped data set is continued.

42 AMDPRDMP found more than 255 CDEs or LLEs on the specified queue/list. The first 255 elements will be printed. Processing continues.

43 The print routine found more than fifty request blocks/partition queue elements on the specified chain. Only the first fifty will be printed. Processing continues.

44 While formatting the specified table, more than 1635 DD entries were found. The first 1635 entries were formatted. Processing continues.

45 The processor address obtained from the LCCA,PCCA, or PSA was a negative value, or greater than X'3F'.

46 The number of MAJOR or MINOR QCBs exceeded the assumed maximum (200).

47 The FORMAT routine found an invalid job name in the TIOT. To be valid, the first character of the job name must be A through Z, or $,#,@, or a blank (X'40'). Processing continues.

48 An I/O error was encountered while attempting to read the specified area of data. Processing continues.

49 The pointer to the CDE queue was zero, or the CDE queue head was zero. Processing ends for the LPAMAP verb. Processing continues with the next verb.

50 More than 50 DQEs were found that describe SQA blocks. Only the first 50 will be processed.

51 The loop control value was zero. Processing of the specified function or control block ends for this request.

52 One more than the maximum number of SRBs was found on the SRB queue. Processing terminates for this queue unless there is a pointer to the last SRB. In that case, a maximum of five more SRBs will be processed.

53 Due to a preceding error, processing will continue using the pointer to the last SRB on the chain. A maximum of five SRBs will be processed from this point.

54 All of the entries to the specified table were zero. Possibly the table or the pointer to the table has been overlaid.

55 The load list pointer or the DEB pointer in the TCB is zero. The zero load list pointer indicates that either no programs were loaded by the LOAD macro instruction; or the load list pointer was overlaid with zeros. The zero DEB pointer indicates that either there were no open data sets or the DEB pointer in the TCB was overlaid with zeros. Processing continues.

56 No ASCBs were found in the dump. Processing continues.

57 None of the jobs specified were found in the dump.

58 The pointer to the chain for the specified control block was zero.

59 The job pack queue field of the TCB is zero. Processing continues.

60 Since a STORE STATUS operation was not performed prior to taking the dump, only the contents of the general purpose registers were stored and will be available to the dumping program.

61 An error occurred during the search for SQA boundaries. At least one block of the SQA was found, so it will be printed.

62 While processing input from an SVC dump, the virtual PCCA address in the PSA for the active processor was found to be zero, and the PCCA address for the PSA of the active processor could not be found in the PCCAVT. This PCCA contains the only pointer to the unprefixed 0-4K block. Processing continues.

63 The specified address was 0. No processors will be listed.

64 The pointer to the last SRB in the SPL queue was zero. Processing continues. However, if the SRB chain is broken, no recovery will be attempted.

65 The pointer, in the SPL queue, to the last SRB on the queue was zero. The queue is processed without using the end pointer.

66 While processing the specified control block chain via the backward pointers, a zero pointer was found before all of the control blocks on the chain were processed. Processing of the memory is terminated.

67 Input is from AMDSADMP. During AMDPRDMP initialization, the prefix value could not be obtained for the specified processor. This makes it impossible to obtain the PSA for this processor.

68 The processing of the specified area or the current function is terminated.

69 While processing from the last SRB, more than five SRBs were found. Processing terminates.

70 The virtual PSA address in the PCCA was not a 4K multiple.

71 The first word of the SPL or SMQ was zero. Processing continues.

72 Either the chain pointer for the specified block addressed an area of storage that could not be extracted from the dump; or the block containing the chain pointer was missing from the dump.

73 AMDPRDMP is printing virtual storage from real dump input. xxxxxx is the real storage location of the virtual address that is being printed.

74 The specified storage area could not be printed because either an I/O error was encountered while attempting to read the block that contains the registers; or the block containing this storage area was missing from the dump.

75 While processing the specified chain, either a zero link field was found before the normal end of the queue was reached; or an attempt to read the link field failed.

76 The number of the specified blocks on the queue exceeded the set limit. The rest of the queue is passed over.

77 The storage protection key has changed. yy is the protection key for the block, at address xxxxxx, that is now being processed.

78 An error condition resulted when AMDSADMP attempted to do a STORE-STATUS for the non-IPLed processor. All or part of the data formatted may be invalid.

79 While formatting the specified table, the pointer to that table was either zero, or addressed an area of storage that was not in the dump. Processing continues.

80 The termination bit of the flag byte at X'21' of the TCB is set indicating that the formatting of the TCB has been completed.

81 The specified area of data could not be read from the dump.

82 The required CVT was not found in the dump. The function requiring the CVT is, therefore, passed over, and processing continues. If no valid CVTs are found, the address of the MAJOR QCB chain will not be available.

83 The specified table could not be located during the initialization phase. Processing for the FORMAT/LPAMAP verb ends.

84 The address of the CVT, in common, is zero. Only the LCCA, PCCA, registers, and PSA for the IPLed processor will be processed.

85 The specified table could not be read from the dump. The rest of the function is passed over. Processing continues with the next control statement.

86 The specified pointer at address xxxxxx could not be read.

87 The specified area could not be read from the dump. The listing being processed may be in error.

88 AMDPRMST was unable to find the bounds of the private area in the SQA. AMDPRPCR was then unable to read the PQE for that memory. AMDPRDPS is not invoked. Processing continues.

89 While accessing the LCCA, PCCA and PSA to get the processor address, either an I/O error was encountered while attempting to read the block containing the address; or the block containing the address was missing from the dump. If the STORE-STATUS for this processor was dumped by AMDSADMP, it will be formatted after the control blocks for any remaining processors, and prior to the unprefixed 0-4K block. Processing continues.

90 Since the ASCB dispatching chain could not be found in the dump, no more ASCBs can be processed.

91 While processing the SRBs of an SPL queue, either an attempt to read the pointer to the first SRB failed and the pointer to the last SRB on the queue was zero; or an attempt to read the pointers to the first and the last SRB on the queue failed. Processing terminates.

92 The PCCAVT was unavailable (reason for unavailability stated in previous output comment); or the read to access the PCCAVT for the pointer to this PCCA failed; or the pointer to this PCCA in the PCCAVT was zero. Since the PCCA is the only pointer to the PSA, the PCA cannot be processed. Processing continues.

93 The specified pointer was not in the dump.

94 An exit module has issued a return code of 4 indicating that it was unsuccessful in obtaining enough main storage for its processing. (AMDPRDMP quarantees that each exit module has up to 8K bytes of storage for processing.)

95 The virtual PSA address in the PCCA for the PSA of the active processor is zero. This address is the only pointer to the unprefixed 0-4K block. (Input is from SVC dump.)

96 While processing the SRBs of an SPL queue, either an error was encountered while attempting to read the pointer to the first SRB or a link field in an SRB; or the link field of an SRB, which was not the last SRB on the chain was zero; or a loop was found while processing the SRB chain. A maximum of five more SRBs will be processed.

97 Due to a preceding error, the ASVT was not entirely processed. Any ASCBs on the ASCB dispatching chain which have not already been selected will be processed.

98 The specified control block did not contain its own name in its ID field. The control block may be overlaid or the pointer to it may be invalid. Processing of the control block continues.

99 Line or lines are omitted because they duplicate the last line that was printed.

100 Printed when the access service returns a non-zero return code after trying to access a field in the PSA, or while trying to follow chains to locate another PSA on a MP system.

101 The operating system flags could not be accessed or the dump system was not an MVS system.

102 Either the TCB address or the CVT address in the parameter list was zero.

103 Non-zero return code from the access service routine while accessing fields from the TCB.

104 Non-zero return code from the access service routine while accessing the MACRF or DSORG bytes from the DCB or if the DSORG byte has an unknown value.

105 Non-zero return code from the access service routine while accessing the DCB address from the DEB.

106 Error in the fields in the FRRS that indicate first entry and last entry and entry length needed to calculate number of entries.

107 Non-zero return code from access service routine while accessing the EXCPD address from the TCB.

108 There are no UCB's pointed to by this DEB.

109 The control block pointed to by the DEB is an ACB, not a DCB.

110 Non-zero return code from format service routine while formatting the FRRS or from the access service routine while accessing fields from the FRRS or FRRS section of the IHSA.

111 Non-zero return code from format service routine while formatting the IHSA or from the access service routine while trying to determine which SU's are on the dump system.

112 Non-zero return code from format service routine while formatting the SCB.

113 Non-zero return code from format service routine while formatting the DCB or from the access service routine while accessing fields from the DCB.

114 Non-zero return code from format service routine while formatting the DEB or from the access service routine while accessing fields from the DEB.

115 Non-zero return code from format service routine while formatting the EED or from the access service routine while accessing fields from the EED.

116 Non-zero return code from the format service routine while formatting the EXCPD (XDBA).

117 Non-zero return code from the format service routine while formatting the ICB.

118 Non-zero return code from the format service routine while formatting the IOB, or from the access service routine while accessing the DCB back-pointer. It will also appear if the DCB back-pointer does not match the DCB address passed in the parameter list, or if the access method flag in the parameter list is not one that requires an IOB.

119 Non-zero return code from the format service routine while formatting the RTCT or from the access service routine while trying to determine which SU's are on the dump system.

120 Non-zero return code from the format service routine while formatting the RTM2WA or from the access service routine while accessing fields in the RTM2WA or trying to determine which SU's are on the dump system.

121 Non-zero return code from the format service routine while formatting the SDWA registers.

122 Non-zero return code from the format service routine while formatting the UCB.

123 Non-zero return code from the access service routine while accessing the PSA or the PSA validity check failed.

124 Non-zero return code from the access service routine while accessing the ICB pointer from the DCB.

125 Any access error or validity check failure while trying to locate the IHSA or to locate the ASCB on the CMS suspend Queue. It also appears for a non-zero return code from the access service routine while accessing the pointer to the TCB from the IHSA.

126 Non-zero return code from the access service routine while accessing the IOB pointer from the DCB.

127 Either the operating system flags could not be accessed or the dump system was not an MVS system.

128 Either the TCB address or the CVT address in the parameter list was zero.

129 Non-zero return code from the access service routine while accessing fields from the TCB.

130 Non-zero return code from the access service routine while accessing LCB related fields in the DCB.

131 The maximum formatter limit of 25 was reached.

132 The maximum formatter limit of 25 was reached.

133 The maximum formatter limit of 25 was reached.

134 Non-zero return code from the access service routine while accessing the EED pointer from the TCB.

135 Non-zero return code from the access service routine while accessing the RTM2WA pointer from the TCB.

136 Non-zero return code from the access service routine while accessing the SCB pointer from the TCB or other fields from the TCB which indicate abended TCB.

137 Non-zero return code from the access service routine while accessing the next EED pointer from the EED.

138 Non-zero return code from the access service routine while accessing the next RTM2WA pointer from the RTM2WA.

139 Non-zero return code from the access service routine while accessing the next SCB pointer from the SCB.

140 Non-zero return code from the access service routine while accessing the pointer to the RTCT from the CVT or while trying to determine if this TCB is the first TCB of the current address space.

141 Either the CVT address or the TCB address in the parameter list was zero.

142 Non-zero return code from the access service routine while accessing either the pointer to the SU string or the SU bit string itself.

143 Either the operating system flags could not be accessed or the dump system was not an MVS system.

144 Non-zero return code from the access service routine while trying to find the UCB or information about the UCB.

145 Non-zero return code from the access service routine while accessing the DSORG and MACRF fields needed from the DCB to determine the access method, or trouble finding the DCB itself.

146 Either an access error accessing the UCB type field or an invalid value in that field.

147 Either an access error accessing the ASCB lock field or an invalid value in that field.

148 The RT1W logical phase number is zero indicating the RT1W is not currently in use.

149 An invalid value in the EEDID field.

150 Data management formatter's ESTAE received control for an unexpected abend.

151 IOS formatter's ESTAE received control for an unexpected abend.

152 RTM formatter's ESTAE received control for an unexpected abend.

153 During reconstruction of a summary dump record insufficient data was found to fill out the length indicated in the record header. The missing end of record was set to zeros.

154 The unique summary dump record which indicates the normal end of the data was found.

155 The unique summary dump record that indicates the normal end of the data was not found before the data was exhausted. Probable cause was a shortage of buffer space when the summary dump occurred or the dump data set was too small to contain the summary dump.
(If the storage areas specified with the SUMLIST parameter on the SDUMP macro are extremely large either of the two previously stated causes could occur.)

156 Probable causes: Either the dump was not an SVC dump or SUMDUMP was not requested when dump was made, or summary dump was unable to provide any data.

157 One of the two system control blocks needed to locate the summary dump data was not available in the dump.

158 A GETMAIN failed.

159 Issued for an error while scanning the DEB chain for the next DEB.

160 Non-zero return code from access service routine while accessing the IOB pointer from the previous IOB.

161 Non-zero return code from the access service routine while accessing the ICB pointer from the previous ICB.

162 A request other than a read direct or write to printer was asked for.

163 Non-zero return code from AMDPRDMP's access service routine while trying to access in-storage LOGREC buffer.

164 Non-zero return code from AMDPRDMP's access service routine while trying to access the RTCT or an in-storage LOGREC buffer address which is not a 4K boundary.

165 EREP enhancement (SU27) is not running on the system.

166 The in-storage LOGREC buffer either was not 4K in size or had a bad beginning, end, and first free pointers in the header.

167 The CVT pointer in the parameter list is zero.

168 Non-zero return code from AMDPRDMP's access service routine while trying to access either the PCCAVT or PCCA.

169 Non-zero return code from AMDPRDMP's access service routine while trying to access the CVT operating system flags or the dump system is not MVS.

170 There are no LOGREC records with data in the in-storage LOGREC buffer.

171 The SRB could not be loaded into storage while SUMMARY data is being put out. Processing continues without SRB formatting.

GTF Trace Records

The following pages describe GTF trace records in alphabetic order by type of event, as indicated in the first field of each record:

DSP or SDSP -- task dispatch or redispatch after SVC interrupt.
EOS, IO, or PCI -- end-of-sense, input/output, or program-controlled interruption.
EXT -- external interruption.
FRR -- functional recovery routine (return from).
HEXFORMAT, SUBSYS, and SYSTEM -- (unformatted trace event record).
LSR -- local supervisor routine (dispatching of).
PGM or P1 -- program interruption.
RNIO -- VTAM remote network.
SIO -- SIO instruction (execution).
SRB -- service request block routine (dispatch or redispatch).
SRM -- system resources manager (return from).
STAE -- STAE or ESTAE exit routine (return from).
SVC -- SVC instruction (execution).
UIO -- unexpected I/O interruption.
USR -- event signalled by GTRACE macro instruction.
USRF9 -- VSAM event.
USRFC -- VTAM event.
USRFE -- BSAM/QSAM/BPAM/BDAM event.
USRFF -- OPEN/CLOSE/EOV event.

In addition to these trace event records, GTF also produces time stamp and lost event records, as described below.

Time Stamp Records

```
**** DATE  DAY ddd  YEAR dddd  TIME dd.dd.dd.dddddd
```

This record precedes the printout of each GTF trace buffer. It indicates the day of the year (Julian) and the time (Greenwich mean time) of day (hour.minute.second.microsecond) when the first trace record was placed in the buffer.

```
TIME ddddd.ddddd
```

This record follows each trace event record and indicates the time (Greenwich mean time) when the trace record was placed in the GTF trace buffer. The time is expressed in the form ssss.mmmmmm (where ssss is seconds and mmmmmm is microseconds of elapsed time since midnight). This record is produced only if the GTF TIME=YES parameter is in effect. Note that the TOD clock value represents local time and that the formatted output expresses time in terms of Greenwich mean time (GMT).

Lost Event Records

```
**** ONE 4096 BYTE BUFFER LOST  TIME dd.dd.dd.dddddd
```

This record indicates that a GTF trace buffer has been lost due to an error condition. It indicates the time of day (hour.minute.second.microsecond) when the first trace record was placed in the buffer.

```
**** LOST EVENTS TOTAL ddddddddd  TIME dd.dd.dd.dddddd
```

This record indicates the total number of trace events lost due to error conditions or trace buffer overflow. This record indicates the time of day (hour.minute.second.microsecond) when the first trace record was placed in the current trace buffer.

DSP AND SDSP TRACE RECORDS

Minimal Trace Record

```
{DSP }  ASCB hhhhhhhh  CPU hhh  PSW hhhhhhhh hhhhhhhh  TCB hhhhhhhh  R15 hhhhhhhh  R0 hhhhhhhh  R1 hhhhhhhh
{SDSP}
```

Comprehensive Trace Record

```
{DSP }  ASCB hhhhhhhh  CPU hhh  JOBN cccccccc  DSP PSW  hhhhhhhh hhhhhhhh  TCB hhhhhhhh  MODN cccccccc
{SDSP}
```

A DSP record represents the dispatching of a task; an SDSP record represents the redispatching of a task after an SVC interrupt.

ASCB hhhhhhhh -- Address of address space control block.

CPU hhh -- Address of processor on which task will be dispatched.

{PSW / DSP PSW} hhhhhhhh hhhhhhhh -- Program status word under which the task is dispatched.

JOBN cccccccc -- Name of the job associated with the task being dispatched, or "N/A" for system or started task.[1]

TCB hhhhhhhh -- Address of the task control block.

{R15 / R0 / R1} hhhhhhhh -- Data that will appear in general registers 15, 0, and 1 when the task is dispatched.

MODN cccccccc -- Name of module that will receive control when the task is dispatched,[1] or one of the following:

WAITTCB -- Indicates that system wait task is about to be dispatched.

SVC-T2 -- Indicates that a type 2 SVC routine resident in the nucleus is about to be dispatched.

SVC-RES -- Indicates that a type 3 SVC routine, or the first load module of a type 4 SVC routine, is about to be dispatched. The routine is located in the pageable link pack area.

SVC-cccc -- Indicates that the second load module, or a subsequent load module, of a type 4 SVC routine is about to be dispatched. The module is located in the fixed or pageable link pack area; the last four characters of the module name are cccc.

IRB* -- Indicates that an asynchronous routine with an associated interruption request block is about to be dispatched; no module name is available.

*cccccccc -- Indicates that error fetch is in the process of loading an error recovery module; the last seven characters of the module name are ccccccc.

[1] If GTF encounters an error while trying to gather this information, the information is replaced by a string of asterisks (********). However, if the error is a page fault, the information is replaced by pppppppp.

EOS, IO, AND PCI TRACE RECORDS

Note: Duplicate records will appear in the trace output for certain types of I/O interruptions that result from a SIOF with a deferred condition code. In these records, bits 6 and 7 of the CSW will be non-zero.

Minimal Trace Record

```
{EOS}
{IO }  ASCB hhhhhhhh  CPU hhhh  PSW hhhhhhhh hhhhhhhh  TCB hhhhhhhh  DEV ADD hhhh  CSW hhhhhhhh hhhhhhhh  SNS hhhh
{PCI}
```

Comprehensive Trace Record

```
        {EOS}
        {IO } hhhh  ASCB hhhhhhhh CPU hhhh  JOBN cccccccc  OLD PSW hhhhhhhh hhhhhhhh  TCB hhhhhhhh  DSID hhhhhhhh
        {PCI}       CSW hhhhhhhh hhhhhhhh  SNS hhhh  R/V CPA hhhhhhhh hhhhhhhh  FLG hhhhhhhh hhhhhhhh hh
```

An EOS, IO, or PCI record indicates an end-of-sense, input/output, or program-controlled interruption.

{EOS / IO / PCI} {,.. DEV ADD hhhh / hhhh} -- Address of the device that caused the interrupt.

ASCB hhhhhhhh -- Address of the ASCB for the address space that started the I/O operation, or "U/A" (unavailable due to unavailability of the IOSB control block).[1]

CPU hhhh -- Address of the processor on which the interruption occurred.

JOBN cccccccc -- Name of the job associated with the task that requested the I/O operation, or "N/A" (not applicable), or "U/A" (unavailable due to unavailability of the IOSB control block).[1]

{PSW / OLD PSW} hhhhhhhh hhhhhhhh -- Program status word stored when the interrupt occurred.

TCB hhhhhhhh -- Address of the TCB for the task that requested the I/O operation, or "N/A" (not applicable), or "U/A" (unavailable due to unavailability of the IOSB control block).

DSID hhhhhhhh -- Contents of the IOSDSID field of the IOSB control block, or "U/A" (indicating that the IOSB is unavailable). IOSDSID contains the address of the data extent block (DEB) or other control block used by PURGE.[1]

CSW hhhhhhhh hhhhhhhh -- Channel status word associated with the interruption, or "U/A" (unavailable due to unavailability of the IOSB control block).

SNS hhhh -- First two bytes of sense data for the device that caused the interrupt, or "U/A" (unavailable due to unavailability of the IOSB control block).[1]

R/V CPA hhhhhhhh hhhhhhhh -- Real address of real channel program, and virtual channel program, or "U/A" (unavailable due to unavailability of the IOSB control block).[1]

FLG hhhhhhhh hhhhhhhh hh -- First four bytes are contents of the IOSFLA, IOSOPT, IOSFMSK, and IOSDVRID fields of the IOSB, or "U/A" (unavailable due to unavailability of the IOSB); last 5 bytes are contents of the UCBFLS, UCBCPU, and UCBCHAN fields of the UCB. These fields are described in Volume 2 and Volume 3. (If GTF encounters an error while gathering this information, each missing byte is replaced by an asterisk (*).)

[1] If GTF encounters an error while gathering this information, the information is replaced by a string of asterisks (********).

EXT TRACE RECORDS

Minimal Trace Record

```
EXT  ASCB hhhhhhhh  CPU hhh  PSW hhhhhhhh hhhhhhhh  TCB hhhhhhhh [INT CPU hhhh  ]
                                                                [TQE TCB hhhhhhhh]
```

Comprehensive Trace Record

```
EXT hhhh ASCB hhhhhhhh  CPU hhh  JOBN cccccccc  OLD PSW hhhhhhhh hhhhhhhh  TCB hhhhhhhh
         [PARM hhhhhhhh  SIG CPU hhh
         [TQE FIELDS:    FLG/EXI hhhhhhhh hhhhhhhh  ASCB hhhhhhhh  TCB hhhhhhhh]
```

An EXT record indicates an external interruption.

EXT hhhh -- External interruption code.

ASCB hhhhhhhh -- Address of ASCB for the address space that was current when the interruption occurred.

CPU hhh -- Address of the processor on which the interruption occurred.

JOBN cccccccc -- Name of the job associated with the interrupted task, or "N/A" (not applicable).[1]

[PSW
[OLD PSW] hhhhhhhh hhhhhhhh -- Program status word stored when the interruption occurred.

TCB hhhhhhhh -- Address of the TCB for the interrupted task, or "N/A" (not applicable, as in the case of an interrupted SRB routine).

PARM hhhhhhhh -- Signal passed on a signal processor interruption (indicated by interruption codes 12hh).

[INT CPU hhhh]
[SIG CPU hhhh] -- Address of the processor on which a signal processor interruption occurred (indicated by interruption codes 12hh).

[TQE
[TQE FIELDS] -- Indicates a clock comparator or CPU timer interruption (interruption code 1004 or 1005 hex). The following fields contain information from the timer queue element (TQE):

FLG/EXI hhhhhhhh hhhhhhhh -- Contents of the TQEFLGS field and TQEEXIT field. TQEFLGS contains flags, which are described in Volume 3; TQEEXIT contains the address of the timer exit routine that is to receive control.

ASCB hhhhhhhh -- Contents of the TQEASCB field (present only for a clock comparator interruption). The TQEASCB contains the address of the ASCB for the address space in which the timer exit routine will be executed.[1]

TCB hhhhhhhh -- Contents of the TQETCB field, or "N/A" (not applicable). The TQETCB contains the address of the TCB for the task under which the timer exit routine will be executed.[1]

[1] If GTF encounters an error while gathering this information, the information is replaced with a string of asterisks (********).

FRR TRACE RECORDS

Minimal Trace Record

```
FRR  ASCB hhhhhhhh  CPU hhhh  PSW hhhhhhhh hhhhhhhh  CC hhhhhhhh  ERRT hhhhhhhh  FLG hhhhhh  RC hh  RETRY hhhhhhhh
```

Comprehensive Trace Record

```
FRR   ASCB hhhhhhhh  CPU hhhh  JOBN cccccccc  FRRN cccccccc  ERR PSW hhhhhhhh hhhhhhhh  ABCC hhhhhhhh  ERRT hhhhhhhh
      FLG hhhhhh  RC hh  RETRY hhhhhhhh
```

An FRR record indicates a return to the recovery termination manager by a functional recovery routine. All fields (except the processor) are gathered from the SDWA control block that was passed to the FRR.

ASCB hhhhhhhh -- Address of the ASCB for the address space in which the error occurred.[1]

CPU hhhh -- Address of the processor associated with the error.

JOBN cccccccc -- Name of the job associated with the error, or "N/A" (not applicable).[1]

FRRN cccccccc -- Name of the functional recovery routine, or "U/A" (indicating that the FRR did not supply a name).[1]

{PSW / ERR PSW} hhhhhhhh hhhhhhhh -- Program status word that was current when the error occurred.[1]

{CC hhhhhhhh / ABCC hhhhhhhh} -- System completion code (first three digits) and user completion code (next three digits).

ERRT hhhhhhhh -- Error-type flags from the SDWAFLGS field of SDWA. These fields are described in Volume 3.

FLG hhhhhh -- Additional flags from the SDWAMCHD and SDWAACF2 fields of SDWA. The flags are contained in the two low-order bytes of this printed field; the high order byte is meaningless.

RC hh -- Return code from the functional recovery routine, as follows:*

0 -- Continue with termination, passing control to the recovery routine at the next higher level, if any.

4 -- Attempt recovery at the retry address supplied by the FRR.

RETRY hhhhhhhh -- Retry address supplied by the FRR, or "N/A" (not applicable, indicating an FRR return code other than 4).[1]

---

[1] If GTF encounters an error while gathering this information, the information is replaced by a string of asterisks (********).

HEXFORMAT, SUBSYS, AND SYSTEM TRACE RECORDS

```
{HEXFORMAT}
{SUBSYS   } AID hh  FID hh  EID hh  hhhhhhhh hhhhhhhh ...
{SYSTEM   }
```

A HEXFORMAT, SUBSYS, or SYSTEM indicates an event for which no record formatting could be performed.

HEXFORMAT -- Indicates an event signalled by a GTRACE macro instruction that specified no formatting routine (FID=00).

SUBSYS -- Indicates an event signalled by a GTRACE macro instruction;.the macro instruction specified a formatting routine (FID=hh) that could not be found.

SYSTEM -- Indicates a system event; the trace record could not be formatted for one of the following reasons:

- If EEEE hex appears in bytes 0 - 1 or 8 - 9 of the recorded data, an unrecoverable error occurred in a GTF data-gathering routine. Message AHL118I is written on the master console, identifying the module that caused the error and the action taken. (The message indicates that GTF will no longer trace this type of event; no more records for this type of event will appear in the trace output.)
- If EEEE hex does not appear in bytes 0 - 1 or 8 - 9 of the recorded data, the record could not be formatted because the GTF formatting routine could not be found.

For descriptions of unformatted system records, refer to OS/VS2 Service Aids Logic.

AID hh -- Application identifier, which should always be AID FF.

FID hh -- Format identifier of the routine (AMDUSRhh or AMDSYShh) that was to format this record.

EID hh -- Event identifier, which uniquely identifies the event that produced the record.

hhhhhhhh hhhhhhhh ... -- Recorded data (256 bytes maximum).

IO TRACE RECORDS

See "EOS, IO, and PCI Trace Records."

LSR TRACE RECORDS

Minimal Trace Record

```
LSR  ASCB hhhhhhhh  CPU hhh  PSW hhhhhhhh hhhhhhhh  TCB hhhhhhhh  R15 hhhhhhhh  R0 hhhhhhhh  R1 hhhhhhhh
```

Comprehensive Trace Record

```
LSR    ASCB hhhhhhhh  CPU hhh  JOBN cccccccc  LSR PSW hhhhhhhh hhhhhhhh  TCB hhhhhhhh
```

An LSR record indicates the dispatching of a local supervisor routine in an address space.

ASCB hhhhhhhh -- Address of the address space control block.

CPU hhh -- Address of the processor on which the routine will be dispatched.

{PSW / LSR PSW} hhhhhhhh hhhhhhhh -- Program status word under which the routine will receive control.

JOBN cccccccc -- Name of job associated with the routine being dispatched, or "N/A" (not applicable).[1]

TCB hhhhhhhh -- Address of the task control block associated with this routine (if the routine is executed as part of a task), or "N/A" (not applicable).

{R15 / R0 / R1} hhhhhhhh -- Data that will appear in general registers 15, 0, and 1 when the local supervisor routine is dispatched.[1]

[1] If GTF encounters an error other than a page fault while trying to gather this information, the information is replaced by a string of asterisks (********). However, if the error is a page fault, the information is replaced by pppppppp.

PCI TRACE RECORDS

See "EOS, IO, and PCI Trace Records."

PGM AND PI TRACE RECORDS

Minimal Trace Record

```
PI   ASCB hhhhhhhh  CPU hhh  PSW hhhhhhhh hhhhhhhh  TCB hhhhhhhh  R15 hhhhhhhh  R1 hhhhhhhh  VPA hhhhhhhh
```

Comprehensive Trace Record

```
PGM  hhh   ASCB hhhhhhhh  CPU hhh  JOBN cccccccc  OLD PSW hhhhhhhh hhhhhhhh  TCB hhhhhhhh  VPA hhhhhhhh  MODN cccccccc
           R0 hhhhhhhh  R1 hhhhhhhh  R2 hhhhhhhh  R3 hhhhhhhh  R4 hhhhhhhh R5 hhhhhhhh  R6 hhhhhhhh R3 hhhhhhhh
           R8 hhhhhhhh  R9 hhhhhhhh  R10 hhhhhhhh  R11 hhhhhhhh  R12 hhhhhhhh  R13 hhhhhhhh  R14 hhhhhhhh R15 hhhhhhhh
```

A PGM or PI record indicates a program interruption.

PGM hhh -- Program interruption code.

ASCB hhhhhhhh -- Address of ASCB for address space in which the interruption occurred.

CPU hhh -- Address of the processor on which the interruption occurred.

JOBN cccccccc -- Name of the job associated with the interruption, or "N/A" (not applicable).[1]

{PSW / OLD PSW} hhhhhhhh hhhhhhhh -- Program status word stored when the interruption occurred.

TCB hhhhhhhh -- Address of the TCB for the interrupted task, or "N/A" (not applicable, as in the case of an interrupted SRB routine).

VPA hhhhhhhh -- Virtual page address, in the case of a segment- or page-translation exception resulting from a reference to the page (indicated by interrupt code 10 or 11 hex); meaningless for other types or program interruptions.

MODN cccccccc -- Name of the module in which the interruption occurred,[1] or one of the following:

- WAITTCB -- Indicates that the system wait task was interrupted.
- SVC-T2 -- Indicates that a type 2 SVC routine resident in the nucleus was interrupted.
- SVC-RES -- Indicates that a type 3 SVC routine, or the first load module of a type 4 SVC routine, was interrupted. The routine is located in the pageable link pack area.
- SVC-cccc -- Indicates that the second load module, or a subsequent load module, of a type 4 SVC routine was interrupted. The module is located in the fixed or pageable link pack area; the last four characters of the load module name are cccc.
- **IRB*** -- Indicates that an asynchronous routine with an associated interrupt request block was interrupted; no module name is available.
- *ccccccc -- Indicates that an error recovery module was in control; the last seven characters of the module name are ccccccc.

{R0 / R15} hhhhhhhh -- Contents of general registers when the interruption occurred.

[1] If a page fault occurs during data gathering, GTF inserts pppppppp in the field. If an error other than a page fault occurs, GTF inserts ******** in the field.

RNIO TRACE RECORDS

Minimal Trace Record

```
RNIO   ASCB hhhhhhhh   CPU hhhh  {IN }  hhhhhhhhhhhhhhhhhhhh
                                 {OUT}
```

Comprehensive Trace Record

```
RNIO   ASCB hhhhhhhh   CPU hhhh   JOBN cccccccc   {IN }  hhhhhhhhhhhhhhhhhhhh
                                                  {OUT}
```

An RNIO record indicates a VTAM remote network I/O event.

ASCB hhhhhhhh -- Address of tne ASCB for the address space of the application associated with the event.

CPU hhhh -- Address of the processor that executed the I/O instruction.

JOBN cccccccc -- Name of the job associated with the interruption.

IN -- Indicates the direction of the I/O is from NCP to VTAM.

OUT -- Indicates the direction of the I/O is from VTAM to NCP.

hhhhhhhhhhhhhhhhhhhh -- The data recorded for inbound events is the transmission header, response header, and response unit. For outbound events, the data is the transmission header, request header, and request unit. For detailed descriptions of the data, refer to OS/VS2 VTAM Data Areas, or OS/VS2 Data Areas.

SIO TRACE RECORDS

Minimal Trace Record

```
SIO  ASCB hhhhhhhh  CPU hhhh  CPA hhhhhhhh hhhhhhhh  CAW hhhhhhhh  DEV ADD hhhh  STATUS hhh  CC h
```

Comprehensive Trace Record

```
SIO hhhh  ASCB hhhhhhhh  CPU hhhh  JOBN cccccccc  R/V CPA hhhhhhhh hhhhhhhh  CAW hhhhhhhh  DSID hhhhhhhh
          FLGS hhhhhhhh hhhh        STAT hhhh  SK ADDR hhhhhhhh hhhhhhhh  CC hh
```

An SIO record indicates the execution of a Start I/O instruction.

{SIO ... DEV ADD hhhh
{SIO hhhh -- Address of the device on which the I/O operation was started.

ASCB hhhhhhhh -- Address of the ASCB for the address space that started the I/O operation, or "U/A" (unavailable due to unavailability of the IOSB control block).[1]

CPU hhhh -- Address of the processor that executed the SIO instruction.

JOBN cccccccc -- Name of the job associated with the SIO operation, or "N/A" (not applicable), or "U/A" (unavailable due to unavailability of the IOSB control block).[1]

{CPA
{R/V CPA hhhhhhhh hhhhhhhh -- Real address of the real channel program address and virtual channel program, or "U/A" (unavailable due to unavailability of the IOSB control block).[1]

CAW hhhhhhhh -- Channel address word containing the real address of the real channel program created by IOS.

DSID hhhhhhhh -- Contents of the IOSDSID field of the IOSB control block, or "U/A" (indicating that the IOSB is unavailable). IOSDISD contains the address of the data extent block (DEB) or other control block used by PURGE.

FLGS hhhhhhhh hhhh -- Contents of the IOSAFF, IOSPATH, IOSOPT, IOSFMSK, and IOSDVRID fields of the IOSB, which are described in Volume 2.

{STATUS hhhh
{STAT hhhh -- Fifth and sixth bytes of the hardware CSW, which indicate the status of the SIO operation.

SK ADDR hhhhhhhh hhhhhhhh -- Seek address indicated in the IOSB for this SIO operation.

CC hh -- SIO condition code, or "U/A" (unavailable due to unavailability of the IOSB control block).[1]

[1] If GTF encounters an error while gathering this information, the information is replaced by a string of asterisks (********).

SRB TRACE RECORDS

Minimal Trace Record

```
SRB  ASCB hhhhhhhh  CPU hhh  PSW hhhhhhhh hhhhhhhh SRB hhhhhhhh  R15 hhhhhhhh  R1 hhhhhhhh  TYPE c
```

Comprehensive Trace Record

```
SRB        ASCB hhhhhhhh  CPU hhh  JOBN cccccccc  SRB PSW hhhhhhhh hhhhhhhh  SRB hhhhhhhh  PARM hhhhhhhh  TYPE c
```

An SRB record indicates the dispatching of an asynchronous routine represented by a service request block (SRB).

ASCB hhhhhhhh -- Address of ASCB for the address space in which the SRB routine will be dispatched. This may or may not be the address space in which the SRB was created.

CPU hhh -- Address of the processor on which the SRB routine will be dispatched.

{PSW / SRB PSW} hhhhhhhh hhhhhhhh -- Program status word under which the SRB routine will receive control.

JOBN cccccccc -- Name of job associated with the SRB being dispatched, or "N/A" (not applicable, as in the case of a global SRB -- see explanation of TYPE field).[1]

SRB hhhhhhhh -- Address of service request block.[1]

{R15 / R1} hhhhhhhh -- Data that will appear in general registers 15 and 1 when the SRB routine is dispatched.[1]

PARM hhhhhhhh -- Four-byte parameter or address of a parameter field, which will be passed to the SRB routine; or "N/A" (not applicable, in the case of a suspended SRB -- see explanation of TYPE field).

TYPE c -- indicates the type of SRB routine, as follows:

GLOBAL -- Denotes an SRB routine selected from the global service priority list (GSPL); this routine performs functions of a system-wide nature, not related to any specific task.

LOCAL -- Denotes an SRB routine selected from the local service priority list (LSPL); this routine performs functions related to a specific task.

SUSPENDED -- Denotes an SRB routine that was dispatched earlier and was subsequently interrupted (by I/O operations, for example, or by a request for a lock). The routine is about to be redispatched.

[1] If GTF encounters an error while gathering this information, the information is replaced by a string of asterisks (********).

SRM TRACE RECORDS

Minimal Trace Record

SRM ASCB hhhhhhhh CPU hhhh R15 hhhhhhhh R0 hhhhhhhh R1 hhhhhhhh

Comprehensive Trace Record

SRM ASCB hhhhhhhh CPU hhhh JOBN cccccccc R15/R0 hhhhhhhh hhhhhhhh R1 hhhhhhhh

An SRM record indicates an entry to the system resources manager.

ASCB hhhhhhhh -- Address of the ASCB for the address space that was current when the system resources manager was entered.[1]

CPU hhhh - Address of the processor used by the system resources manager.

JOBN cccccccc -- Name of the job associated with this entry to the system resources manager, or "N/A" (not applicable).[1]

{R15 / R0 / R1} hhhhhhhh -- Data that was contained in general registers 15, 0, and 1 when the system resources manager passed control to GTF. The data includes a SYSEVENT code in the low-order byte of reigster 0. Register data is explained in the SYSEVENT Summary in Section 5.

[1]If GTF encounters an error while gathering this information, the information is replaced by a string of asterisks (********).

STAE TRACE RECORDS

Minimal Trace Record

```
STAE  ASCB hhhhhhhh  CPU hhhh  PSW hhhhhhhh hhhhhhhh  CC hhhhhhhh  FLG hhhhhh  RC hh  RETRY hhhhhhhh  RTCA hhhhhhhh
```

Comprehensive Trace Record

```
STAE  ASCB hhhhhhhh  CPU hhhh  JOBN cccccccc  ESTN cccccccc  ERR PSW hhhhhhhh hhhhhhhh  ABCC hhhhhhhh  ERRT hhhhhhhh
      FLG hhhhhh  RC hh  RTRY hhhhhhhh  RTCA hhhhhhhh
```

A STAE record indicates a return to the recovery termination manager by a STAE or ESTAE exit routine. The RTCA field indicates the type of routine (STAE or ESTAE).

ASCB hhhhhhhh -- Address of the ASCB for the address space in which the error occurred.

CPU hhhh -- Address of the processor associated with the error.

JOBN cccccccc -- Name of the job associated with the error, or "N/A" (not applicable).*

ESTN cccccccc -- Name of the routine (for ESTAE only), or "U/A" (indicating that the routine did not supply a name).[1]

{PSW / ERR PSW} hhhhhh hhhhhhhh -- Program status word that was current when the error occurred, or "U/A" (unavailable due to unavailability of the SDWA (system diagnostic work area)).[1]

{CC hhhhhhhh / AB CChhhhhhhh} -- System completion code (first three digits) and user completion code (next three digits), or "U/A" (unavailable due to unavailability of the SDWA).[1]

ERRT hhhhhhhh -- Error type flags from the SDWAFLGS field of the SDWA. These fields are described in Volume 3.[1]

FLG hhhhhh -- Flag bytes (SDWAERRA, and SDWAACF2 fields of the SDWA), or "N/A" (not applicable, in the case of a STAE routine), or "U/A" (unavailable due to unavailability of the SDWA).[1]

Note: In the comprehensive trace record, SDWAERRA is omitted; the first printed byte is meaningless.

RC hh -- Return code from the STAE or ESTAE exit routine, as follows:[1]

- 00 -- Continue with termination, passing control to the recovery routine at the next higher level, if any.
- 04 -- Attempt recovery by passing control to the retry address supplied by the exit routine.
- 10 -- Continue with termination; suppress execution of STAI or ESTAI exit routines. (This return code is valid only for an ESTAI exit routine.)
- FF -- Indicates that an ESTAE exit routine returned an invalid SDWA address to the recovery termination manager.

{RTRY
{RETRY} hhhhhhhh -- Retry address supplied by the exit routine, or "N/A" (not applicable, indicating an exit routine return code other than 4). This field is zero if the exit routine returned an invalid SDWA address to the recovery termination manager.[1]

RTCA hhhhhhhh -- Address of the SDWA (system diagnostic work area) passed to the exit routine by the recovery termination manager. If the routine was a STAE (rather than ESTAE) routine, the address is in two's complement form. The address is zero if no SDWA was available, or if the exit routine returned an invalid SDWA address (indicated by "RC FF RETRY 00000000").

[1]If GTF encounters an error while gathering this information, the information is replaced by a string of asterisks (********).

SUBSYS TRACE RECORDS

See "HEXFORMAT, SUBSYS, and SYSTEM Trace Records."

SVC TRACE RECORDS

Minimal Trace Record

```
SVC  ASCB hhhhhhhh  CPU hhhh  PSW hhhhhhhh hhhhhhhh  TCB hhhhhhhh  R15 hhhhhhhh  R0 hhhhhhhh  R1 hhhhhhhh
```

Comprehensive Trace Record

```
SVC hhh  ASCB hhhhhhhh  CPU hhhh  JOBN cccccccc  OLD PSW hhhhhhhh hhhhhhhh  TCB hhhhhhhh  MODN cccccccc  DDNAM cccccccc
         R15 hhhhhhhh  R0/R1 hhhhhhhh hhhhhhhh  [additional fields]
```

An SVC record indicates an SVC interruption.

SVC hhh -- SVC interruption code (SVC number).

ASCB hhhhhhhh -- Address of the ASCB for the address space in which the interruption occurred.

CPU hhh -- Address of the processor on which the interruption occurred.

JOBN cccccccc -- Name of the job associated with the interruption, or "N/A" (not applicable).[1]

(PSW / OLD PSW) hhhhhhhh hhhhhhhh -- program status word stored when the interruption occurred.

TCB hhhhhhhh -- Address of the TCB for the interrupted task (that is, the task that executed the SVC instruction.)

MODN cccccccc -- Name of the module containing the SVC instruction,[1] or one of the following:

SVC-T2 -- Indicates a type 2 SVC routine resident in the nucleus.

SVC-RES -- Indicates a type 3 SVC routine, or the first load module of a type 4 SVC routine. The routine is located in the pageable link pack area.

SVC cccc -- Indicates the second or subsequent load module of a type 4 SVC routine. The routine is located in the fixed or pageable link pack area; the last four characters of the load module name are cccc.

IRB* -- Indicates an asynchronous routine with an associated interruption request block; no module name is available.

*ccccccc -- Indicates an error recovery module; the last seven characters of the load moudle name are ccccccc.

[1] If GTF encounters an error while gathering this information, the information is replaced by a string of asterisks (********).

DDNAM cccccccc -- Name of the DD statement associated with the SVC (where applicable).

{R15 / R0 / R1} hhhhhhhh -- Data that was contained in general registers 15, 0, and 1 when the SVC instruction was executed. The SVC Summary in Section 5 describes register contents for each SVC.

[additional fields] -- Vary with the SVC number. The SVC Summary in Section 5 describes these fields (DDNAME, PLIST, etc.) for each SVC.

SYSTEM TRACE RECORDS

See "HEXFORMAT, SUBSYS, and SYSTEM Trace Records."

UIO TRACE RECORDS

Minimal Trace Record

```
UIO  ASCB N/A     CPU hhhh PSW hhhhhhhh hhhhhhhh  DEV ADD hhhh  CSW hhhhhhhh hhhhhhhh
```

Comprehensive Trace Record

```
UIO  hhhh  ASCB N/A     CPU hhhh JOBN N/A  OLD PSW hhhhhhhh hhhhhhhh  CSW hhhhhhhh hhhhhhhh
```

A UIO record indicates an unexpected I/O interruption has occurred (an interruption from a device on which no I/O operation has been started).

{UIO ... DEV ADD hhhh / UIO hhhh} -- Address of the device that caused the I/O interruption.

ASCB N/A -- Indicates that there is no address space control block that can be associated with this interruption.

CPU hhhh -- Address of the processor on which the interruption occurred.

JOBN N/A -- Indicates that there is no job name that can be associated with this interruption.

{PSW / OLD PSW} hhhhhhhh hhhhhhhh -- Program status word that was stored when the interruption occurred.

CSW hhhhhhhh hhhhhhhh -- Channel status word associated with the interruption.

USR TRACE RECORDS

Unformatted Trace Record

```
USR  AID hh  FID hh  EID hh  hhhhhhhh hhhhhhhh ...
```

Formatted Trace Record

```
USRhh  hhh  ASCB hhhhhhhh  JOBN cccccccc
                  xxxx...
```

A USR record indicates an event signalled by a GTRACE macro instruction. The record is formatted by a user-supplied formatting routine (AMDUSRhh), or is printed without formatting if the routine cannot be found.

Unformatted Trace Record: Fields are as follows:

AID hh -- Application identifier, which should always be AID FF.

FID hh -- Format identifier of the routine (AMDUSRhh) that was to format this record.

EID hh -- Event identifier, which uniquely identifies the event that produced the record.

hhhhhhhh hhhhhhhh ... Recorded data (256 bytes maximum).

Formatted Trace Record: Fields are as follows:

USRhh -- Identifies the user-supplied formatting routine (AMDUSRhh). The following USR records are generated and formatted by components of OS/VS2, and are described on succeeding pages:

- USRF9 Trace Records (VSAM)
- USRFE Trace Records (BSAM/QSAM/BPAM/BDAM)
- USRFF Trace Records (OPEN/CLOSE/EOV)

hhh -- Event identifier specified in the GTRACE macro instruction.

ASCB hhhhhhhh -- Address of the ASCB for the address space that created the record.

JOBN cccccccc -- Name of the job associated with the address space.

xxxx... -- User-formatted trace data.

USRF9 Trace Records (VSAM)

```
USRF9   FF5   ASCB hhhhhhhh  JOBN cccccccc

              JOB NAME cccccccc  STEP NAME cccccccc

              TIOT ENT hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh

              ACB      hhhhhhhh hhhhhhhh hhhhhhhh ...
                       hhhhhhhh hhhhhhhh hhhhhhhh ...

              AMBL     hhhhhhhh hhhhhhhh hhhhhhhh ...
                       hhhhhhhh hhhhhhhh hhhhhhhh ...

              AMB      hhhhhhhh hhhhhhhh hhhhhhhh ...
                       hhhhhhhh hhhhhhhh hhhhhhhh ...

              AMDSB    hhhhhhhh hhhhhhhh hhhhhhhh ...
                       hhhhhhhh hhhhhhhh hhhhhhhh ...

              AMB      hhhhhhhh hhhhhhhh hhhhhhhh ...
                       hhhhhhhh hhhhhhhh hhhhhhhh ...

              AMDSB    hhhhhhhh hhhhhhhh hhhhhhhh ...
                       hhhhhhhh hhhhhhhh hhhhhhhh ...
```

A USRF9 trace record indicates the opening or closing of a VSAM data set.

USRF9 -- Identifies VSAM's trace-record formatting routine (AMDUSRF9).

FF5 -- Event identifier specified by the GTRACE macro instruction.

ASCB hhhhhhhh -- Address of the ASCB for the address space in which the event occurred.

JOBN cccccccc
JOB NAME cccccccc -- Name of the job.

STEP NAME cccccccc -- Name of the job step during which the event occurred.

TIOT ENT hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh -- Data set DD entry from the task I/O table (TIOT).[1]

ACB hhhhhhhh ... -- Contents of the data set's access-method control block (ACB).[1]

AMBL hhhhhhhh ... -- Contents of the AMB list (AMBL).[1]

AMB hhhhhhhh ... -- Contents of the access method block (AMB).[1] The first AMB is for data, the second for the index.

AMDSB hhhhhhhh ... -- Contents of the access method statistics block (AMDSB).[1] The first AMDSB is for data, the second for the index.

[1]For descriptions of individual fields, refer to the appropriate data area description in Volume 2 or Volume 3.

USRFD TRACE RECORDS (VTAM -- NCP LINE)

```
USRFD  FF2   ASCB   hhhhhhhh   JOBN   cccccccc   TIME hh     (SLOWDOWN)
 LINE   DNODE   cccccccc       EP hh
                LCD  h  PCF  h  TIME  hh  SCF  hh  PDF  hh  LCD  h  PCF  h  TIME  hh  SCF  hh  PDF  hh
                LCD  h  PCF  h  TIME  hh  SCF  hh  PDF  hh  LCD  h  PCF  h  TIME  hh  SCF  hh  PDF  hh
```

A USRFD (NCP line trace) record indicates the communication between a VTAM application and a VTAM destination.

USRFD -- Identifies VTAM's trace-record formatting routine (AMDUSRFD).

FF2 -- Event identifier specified by the GTRACE macro instruction.

ASCB hhhhhhhh -- Address of the ASCB associated with VTAM.

JOBN cccccccc -- Name of the job associated with the address space.

LINE -- Indicates trace data of hardware NCP lines.

DNODE cccccccc -- NCP name associated with the line being traced.

EP hh -- EP subchannel address.

TIME hh -- Time stamp value of the NCP line trace buffer.

SLOWDOWN -- If present, indicates line trace is terminating due to slowdown processing.

LCD h -- Line control definer.

PCF h -- Primary control field.

TIME -- Time stamp value of the line trace entry.

SCF hh -- Secondary field.

PDF hh -- Parallel data field.

USRFD TRACE RECORDS (VTAM -- CONTROL LAYER)

Input Trace Record

```
USRFD  FFl ' ASCB  hhhhhhhh  JOBN  cccccccc  C/L  IN  ANODE  cccccccc  TEXT  hhhhhhhh  hhhhhhhh  hhhhhhhh
                                                     DNODE  cccccccc        hhhhhhhh  hhhhhhhh  hhhhhhhh
                                                                           hhhhhhhh  hhhhhhhh   *............*
                                                                                               *..........   *
```

Output Trace Record

```
USRFD  FFl  ASCB  hhhhhhhh  JOBN  cccccccc  C/L  OUT  ANODE  cccccccc  TEXT  hhhhhhhh  hhhhhhhh  hhhh  *.......*
                                                                    DNODE cccccccc
```

A control layer record indicates the contents of the message buffers.

USRFD -- Identifies VTAM's trace-record formatting routine (AMDUSRFD).

FFl -- Event identifier specified by the GTRACE macro instruction.

ASCB hhhhhhhh -- Address of the ASCB of the VTAM application.

JOBN cccccccc -- Name of the job associated with the address space.

C/L IN -- Indicates in-bound control layer buffer information.

C/L OUT -- Indicates out-bound control layer buffer information.

ANODE cccccccc -- Application (source) node name.

TEXT hhhhhhhh -- Buffer text information.

DNODE cccccccc -- Destination node name.

USRFD TRACE RECORDS (VTAM -- TPIOS)

Input Trace Record

```
      TPIOS  IN  ANODE  cccccccc  FDB  hhhhhhhh  hhhhhhhh  hhhhhhhh  RSVD  hhhh  LNG2  hhhh
       LOCAL     DNODE  cccccccc  FSB  hhhhhhhh  hhhhhhhh  hhhhhhhh  hhhhhhhh  hhhhhhhh  hhhhhhhh  hhhhhhhh  hhhhhhhh
                                   FDB  hhhhhhhh  hhhhhhhh  hhhhhhhh  *.................
                                   TEXT hhhhhhhh  hhhhhhhh  *.............
```

Output Trace Record

```
          USRFD  FEF  ASCB  hhhhhhhh  JOBN  cccccccc

          TPIOS OUT ANODE  cccccccc  FDB  hhhhhhhh  hhhhhhhh  hhhhhhhh  RSVD  hhhh  LNG2  hhhh
           LOCAL    DNODE  cccccccc  TEXT hhhhhhhh  hhhhhhhh  hhhhhhhh  hhhhhhhh  hhhhhhhh  *.................*
                                      hhhhhhhh  hhhhhhhh  hhhhhhhh  hhhhhhhh  hhhhhhhh  *.................*
```

A TPIOS record contains the contents of message buffers during TPIOS processing.

USRFD -- Identifies VTAM's trace-record formatting routine (AMDUSRFD).

FEF -- Event identifier specified by the GTRACE macro instruction.

ASCB hhhhhhhh -- Address of the ASCB of the VTAM application.

JOBN cccccccc -- Name of the job associated with the address space.

TPIOS IN -- Indicates in-bound TPIOS buffer information.

TPIOS OUT -- Indicates out-bound TPIOS buffer information.

LOCAL -- Indicates data being transferred to or from a locally attached device.

ANODE cccccccc -- Application (source) node name.

FDB hhhhhhhh -- Feedback data block.

FSB hhhhhhhh -- Feedback status block.

RSVD hhhh -- Reserved. Also used as a VTAM workarea.

LNG2 hhhh -- Length of buffer.

DNODE cccccccc -- Destination node name.

TEXT hhhhhhhh -- Buffer text information.

USRFD TRACE RECORDS (VTAM -- TPIOS REMOTE)

Input Trace Record

```
      USRFD   FEF   ASCB   hhhhhhhh   JOBN   cccccccc

   TPIOS IN ANODE   cccccccc   FDB      hhhhhhhh  hhhhhhhh  hhhhhhhh RSVD  hhhh  LNG2   hhhh
    REMOTE  DNODE   cccccccc   FSB      hhhhhhhh  hhhhhhhh  hhhhhhhh hhhhhhhh  hhhhhhhh  hhhhhhhh  hhhhhhhh  hhhhhhhh
                               THRH[1]  hhhhhhhh  hhhhhhhh  hhhhhhhh hh(hhhhhh  hhhhhhhh)
                               TEXT     hhhhhhhh  hhhhhhhh  hhhhhhhh hhhhhhhh  hhhhhhhh  *....................*
                                        hhhhhhhh  hhhhhhhh  hhhhhhhh hhhhhhhh  hhhhhhhh  *....................*
```

Output Trace Record

```
       USRFD   FEF   ASCB   hhhhhhhh   JOBN   cccccccc

    TPIOS OUT ANODE   cccccccc   FDB      hhhhhhhh  hhhhhhhh  hhhhhhhh RSVD hhhh   LNG2   hhhh
     REMOTE   DNODE cccccccc    THRH     hhhhhhhh  hhhhhhhh  hhhhhhhh hh(hhhhhh  hhhhhhhh)
                                TEXT     hhhhhhhh  hhhhhhhh  hhhhhhhh hhhhhhhh  hhhhhhhh  *....................*
                                         hhhhhhhh  hhhhhhhh  hhhhhhhh hhhhhhhh  hhhhhhhh  *....................*
```

A TPIOS record contains the contents of message buffers during TPIOS processing.

USRFD -- Identifies VTAM's trace-record formatting routine (AMDUSRFD).

FEF -- Event identifier specified by the GTRACE macro instruction.

ASCB hhhhhhhh -- Address of the ASCB of the VTAM application.

JOBN cccccccc -- Name of the job associated with the address space.

TPIOS IN -- Indicates in-bound TPIOS buffer information.

TPIOS OUT -- Indicates out-bound TPIOS buffer information.

REMOTE -- Indicates data being transferred to/from a device attached to a 370x.

ANODE cccccccc -- Application (source) node name.

FDB hhhhhhhh -- Feedback data block.

RSVD hhhh -- Reserved.

LNG2 hhhh -- Length of buffer.

RSVD2 hhhhhhhh -- Reserved. Also used as a workarea.

DNODE cccccccc -- Destination node name.

FSB hhhhhhhh -- Feedback status block.

THRH hhhhhhhh -- Transmission and request or response headers.

TEXT hhhhhhhh -- Buffer text information.

---

[1]The SAF/DAF fields are reserved for the FID0 PIUS in the THRH fields.

USRFE Trace Records (BSAM/QSAM/BPAM/BDAM)

```
USRFE  hhh  ASCB hhhhhhhh  JOBN cccccccc

            BSAM/QSAM/BPAM/BDAM  TRACE RECORD  DDNAME cccccccc  ABEND CODE hhh  RETURN CODE hh  TIME=dd.dd.dd

            cccc... [AT LOCATION hhhhhhhh]
            hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh ...
            hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh ...
```

A USRFE trace record indicates abnormal termination of a data management access method routine.

USRFE -- Identifies the data management trace-record formatting routine (AMDUSRFE).

hhh -- Event identifier specified in the GTRACE macro instruction. The event identifier (EID) corresponds to the abnormal completion code (CC) as follows: EID(cc): FF3(002), FF4(008), FF6(112), FF7(215), FF8(119), FF9(235), FFA(239), FFB(145), FFC(251), FFD(451), FFE(169)

ASCB hhhhhhhh -- Address of the ASCB for the address space in which the abnormal termination occurred.

JOBN cccccccc -- Name of the job associated with the address space.

BSAM/QSAM/BPAM/BDAM TRACE RECORD -- Record identification provided by the AMDUSRFE formatting routine.

DDNAME cccccccc -- Name of the DD statement for the data set being processed.

ABEND CODE hhh -- Task abnormal completion code.

RETURN CODE hh -- Return code from the module that detected the error condition.

TIME=dd.dd.dd -- Time (hour.minute.second) when the GTRACE macro instruction was issued (or blank, if the time is not available).

cccc... [AT LOCATION hhhhhhhh]
hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh ... -- Data area name, or name and address, followed by data area contents. For descriptions of the DCB, DEB, ECB, and IOB, refer to Volume 2; for description of other data areas, refer to OS/VS2 SAM Logic, or to OS/VS2 BDAM Logic.

USRFF Trace Records (OPEN/CLOSE/EOV)

```
                                    USRFF  FFF  ASCB hhhhhhhh  JOBN cccccccc
                                                 xxxx...
```

A USRFF trace record indicates either an abnormal termination during OPEN/CLOSE/EOV processing, or a user request for a work area trace.

USRFF -- Identifies the OPEN/CLOSE/EOV trace record formatting routine (IMDUSRFF).

FFF -- Event identifier specified by the GTRACE macro instruction.

xxxx ... -- Data recorded by OPEN/CLOSE/EOV. For abnormal termination trace records, this data consists of unformatted RRCBSAs (recovery routine control block save areas). For user-requested work area trace records, the data consists of formatted control blocks and OPEN/CLOSE/EOV work areas.

For a complete description of the recorded data, refer to OS/VS2 OPEN/CLOSE/EOV Logic.

ERROR
INDICATORS

Section 4: Error Indicators

This section summarizes the following system error indicators:

- System Error Messages
- Machine-Check Interruption Code
- PSW Error Indicators
  (program interruption codes, condition codes, and wait state codes)
- ECB Indicators
  (error completion codes and abnormal completion codes)
- I/O Error Indicators
  (channel status word, limited channel logout, and UCB sense bytes)

System Error Messages

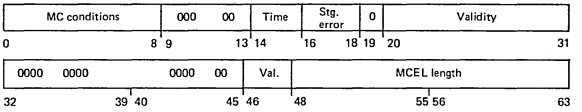
System error messages are identified by seven or eight-character message numbers. The first three characters are the same as the first three characters in the name of a module associated with the message.

For each three-character message number (module name) prefix, the Module Summary in Section 5 identifies the corresponding system component, program logic manual, and message library publication.

Machine-Check Interruption Code

The machine-check interruption code is stored in location E8 (hex) of the prefixed storage area (PSA) for each processor. The format is as follows:

insert 15 (MCH logout diagram)

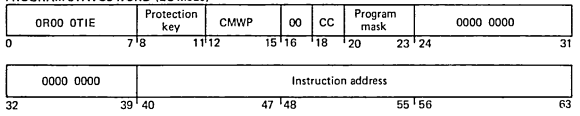


0	System damage	15	Delayed	25	Region code
1	Instr. proc'g damage	16	Uncorrected	27	Floating-pt registers
2	System recovery	17	Corrected	28	General registers
3	Timer damage	18	Key uncorrected	29	Control registers
4	Timing facil. damage	20	PSW bits 12-15	30	CPU ext'd logout
5	External damage	21	PSW masks and key	31	Storage logical
7	Degradation	22	Prog. mask and CC	46	CPU timer
8	Warning	23	Instruction address	47	Clock comparator
14	Backed-up	24	Failing stg. address		

PSW Error Indicators

PSW error indicators include program interruption codes, condition codes, and wait state codes. The general format of the PSW is as follows:

PROGRAM STATUS WORD (EC Mode)



1	(R) Program event recording mask	15	(P=1) Problem state
5	(T=1) Translation mode	18-19	(CC) Condition code
6	(I) Input/output mask	20	Fixed-point overflow mask
7	(E) External mask	21	Decimal overflow mask
12	(C=1) Extended control mode	22	Exponent underflow mask
13	(M) Machine-check mask	23	Significance mask
14	(W=1) Wait state		

PROGRAM INTERRUPTION CODES

0001	Operation exception	000C	Exponent overflow excp
0002	Privileged operation excp	000D	Exponent underflow excp
0003	Execute exception	000E	Significance exception
0004	Protection exception	000F	Floating-point divide excp
0005	Addressing exception	0010	Segment translation excp
0006	Specification exception	0011	Page translation exception
0007	Data exception	0012	Translation specification excp
0008	Fixed-point overflow excp	0013	Special operation exception
0009	Fixed-point divide excp	0040	Monitor event
000A	Decimal overflow exception	0080	Program event (code may be combined with another code)
000B	Decimal divide exception		

CONDITION CODES

Condition Code Setting	0	1	2	3
Mask Bit Position	8	4	2	1
General Instructions				
ADD (and ADD HALFWORD)	zero	< zero	> zero	overflow
ADD LOGICAL	zero, no carry	not zero, no carry	zero, carry	not zero, carry
AND	zero	not zero	– –	– –
COMPARE (and COMPARE HALFWORD)	equal	low	high	– –
COMPARE AND SWAP	equal	not equal	– –	– –
COMPARE DOUBLE AND SWAP	equal	not equal	– –	– –
COMPARE LOGICAL	equal	low	high	– –
COMPARE LOGICAL CHARACTERS UNDER MASK	equal	low	high	—
COMPARE LOGICAL LONG	equal	low	high	– –
EXCLUSIVE OR	zero	not zero	– –	– –
INSERT CHARACTERS UNDER MASK	zero	1st bit one	1st bit zero	– –
LOAD AND TEST	zero	< zero	> zero	– –
LOAD COMPLEMENT	zero	< zero	> zero	overflow
LOAD NEGATIVE	zero	< zero	– –	– –
LOAD POSITIVE	zero	– –	> zero	overflow
MOVE LONG	count equal	count low	count high	destr. overlap
OR	zero	not zero	– –	– –
SHIFT LEFT DOUBLE	zero	< zero	> zero	overflow
SHIFT LEFT SINGLE	zero	< zero	> zero	overflow
SHIFT RIGHT DOUBLE	zero	< zero	> zero	– –
SHIFT RIGHT SINGLE	zero	< zero	> zero	– –
STORE CLOCK	set	not set	error	not operational
SUBTRACT (and SUBTRACT HALFWORD)	zero	< zero	> zero	overflow
SUBTRACT LOGICAL	– –	not zero, no carry	zero, carry	not zero, carry
TEST AND SET	zero	one	– –	– –
TEST UNDER MASK	zero	mixed	– –	ones
TRANSLATE AND TEST	zero	incomplete	complete	– –
Decimal Instructions				
ADD DECIMAL	zero	< zero	> zero	overflow
COMPARE DECIMAL	equal	low	high	– –
EDIT	zero	< zero	> zero	– –
EDIT AND MARK	zero	< zero	> zero	– –
SHIFT AND ROUND DECIMAL	zero	< zero	> zero	overflow
SUBTRACT DECIMAL	zero	< zero	> zero	overflow
ZERO AND ADD	zero	< zero	> zero	overflow
Floating-Point Instructions				
ADD NORMALIZED	zero	< zero	> zero	– –
ADD UNNORMALIZED	zero	< zero	> zero	– –
COMPARE	equal	low	high	– –
LOAD AND TEST	zero	< zero	> zero	– –
LOAD COMPLEMENT	zero	< zero	> zero	– –
LOAD NEGATIVE	zero	< zero	– –	– –
LOAD POSITIVE	zero	– –	> zero	– –
SUBTRACT NORMALIZED	zero	< zero	> zero	– –
SUBTRACT UNNORMALIZED	zero	< zero	> zero	– –

Condition Code Setting	0	1	2	3
Mask Bit Position	8	4	2	1
Input/Output Instructions				
CLEAR CHANNEL	I/O system reset performed	– –	channel busy	not operational
CLEAR I/O	no operation in progress	CSW stored	channel busy	not operational
HALT DEVICE	interruption pending, or busy	CSW stored	channel working	not operational
HALT I/O	interruption pending	CSW stored	burst op. stopped	not operational
START I/O	successful	CSW stored	busy	not operational
START I/O FAST RELEASE	successful	CSW stored	busy	not operational
STORE CHANNEL ID	ID stored	CSW stored	busy	not operational
TEST CHANNEL	available	interruption pending	burst mode	not operational
TEST I/O	available	CSW stored	busy	not operational
System Control Instructions				
LOAD REAL ADDRESS	translation available	ST entry invalid	PT entry invalid	length violation
RESET REFERENCE BIT	R bit zero, C bit zero	R bit zero, C bit one	R bit one, C bit zero	R bit one, C bit one
SET CLOCK	set	secure	– –	not operational
SIGNAL PROCESSOR	order code accepted	status stored	busy	not operational

Explanation:

> zero — Result is greater than zero

high — First operand compares high

< zero — Result is less than zero

low — First operand compares low

NOTE: The condition code may also be changed by LOAD PSW, SET PROGRAM MASK, and DIAGNOSE, and by an interruption.

Wait State Codes

Wait state codes are found in the 12 low-order bits of the PSW when the PSW wait bit (bit 14) is set to one. The PSW format is:

yyyyyyyy xxxxxzzz

yyyyyyyy

This is the left half of the program status word. This half has two forms:

070s0000 If the wait state code (zzz in the right half of the program status word) is 000, no tasks are ready in the system. Otherwise, an error condition, indicated by zzz has occurred.

000s0000 System wait state caused by an error condition.

In each of the above forms, s represents bits 12-15 (the CMWP bits).

xxxxxzzz

This is the right half of the program status word. The wait state code, zzz, indicates the error condition.

Wait state codes are described briefly in the following table. For complete explanations and problem determination procedures, refer to *OS/VS Message Library: VS2 System Codes*.

Code	Explanation
001	IPL: "Not operational" is the status of the volume containing the nucleus being loaded.
002	IPL: An I/O operation was not initiated; CSW was stored; channel was not busy.
003	IPL: An I/O operation was not initiated because a "not operational" response was received from a SIO instruction; CSW was not stored; channel was not busy.
004	IPL: An I/O operation was not initiated; CSW was not stored; channel was not busy after execution of a Test I/O instruction.
005	IPL: A unit check caused an I/O interruption; the address of the CCW is placed in location X'4C'; the first four unit-check sense bytes are placed in location X'54'.
006	IPL: A interface control check, or channel control check, or channel data check, or program check, or channel chaining check occurred.
007	NIP: A console was not available.
00A	NIP: A catalog entry for the SYS1.LINKLIB data set was not found on or not retrievable from the system catalog.
00B	The master scheduler terminated abnormally during initialization; a dump was taken to the SYS1.DUMPxx data set.
00C	IPL: An active nucleus member is not edited in the scatter format; the nucleus cannot be loaded.
00D	The master scheduler terminated abnormally during initialization; no dump was taken.
00E	IPL: The SYS1.NUCLEUS data set or active member, IEANUCOx, of SYS1.NUCLEUS is not found on the IPL volume; the system nucleus cannot be located.
00F	IPL: An IPLed volume does not contain the IPL text.
013	NIP: An error occurred from which recovery was impossible.
014	Recursive translation exceptions occurred due to incorrect initialization, or overlaying of the master address space segment tables or common page tables occurred.
017	IPL: A unit check occurred following a successfully started I/O operation.
018	IPL: The nucleus is too big for the machine size; the space available for relocation dictionary records was exceeded.
019	IPL: An unexpected program interruption occurred.
01A	An attempt was made to execute multiprocessing modules that were not included in the nucleus when the system was generated.
01B	The error condition specified on a SLIP command has been met and the system is placed in a "wait state", as requested.
01C	Supervisor control suffered a double recursive abnormal termination in supervisor control super FRR.
021	NIP: An I/O error occurred on the main console following an EXCP operation.
022	A page fault occurred for a duplexed page but primary and secondary devices are either in a quiesced or "not ready" state.
024	The system termination facility failed to receive the expected interruption while attempting to write record to the SYS1.LOGREC data set.
026	Either a power interruption occurred or the operator replied "REST" to the message, ICFTIM21.

Code	Explanation
027	A power interruption occurred, or the operator replied "REST" to the message, ICFTIM21, or the operator replied "STOP" to one of the power warning initialization messages.
029	The time-of-day clock is in error.
02A	The hardware could not signal the other processor because the other processor did not appear to exist.
02B	The hardware could not signal the other processor, because the other processor had experienced a similar problem and was the first processor to enter the 02A wait state.
02D	NIP: An unconditional mount was issued requesting a Mass Storage System device (3850), which is not supported during NIP.
02E	ASM can no longer access a required page data set due to an unusual number of permanent I/O errors.
02F	A direct access device containing a page data set is unavailable for an I/O operation.
030	NIP: An unanticipated request for abnormal task termination (ABEND) has occurred.
031	NIP: The IPL volume resides on a unit for which there is no UCB.
032	NIP: A required module is not found in the SYS1.NUCLEUS data set by BLDL.
033	NIP: An I/O error occurred during BLDL's processing for a required module.
034	The operating system is incompatible with the processor. Dynamic address translation, the CS, CDS, IPK, and SPKA instructions or the clock comparator feature are not supported on the processor.
036	An attempt was made to expand storage initially allocated to SQA before initializing the paging subsystem.
037	The DSCB for SYS1.LOGREC, SYS1.SVCLIB, SYS1.PARMLIB, or SYS1.LINKLIB could not be successfully read from the volume where the data set resides.
038	NIP: Insufficient real storage space is available for initializing the VS2 system.
039	NIP: The required direct access volume could not be successfully mounted due to a conflict with another online volume.
03A	NIP: An error occurred during an attempt to build the LPA (cold start).
03B	NIP: A module required for continued system operation cannot be found in the system LPA.
03C	ASM detected insufficient auxiliary storage resources for system operations.
03D	NIP: The subroutine, IEAVCSEG, returned a nonzero code indicating that an error occurred in creating a segment table entry, the page table, or the external page table.
03F	NIP: A NIP function was improperly invoked.
040	NIP: An abend request occurred during NIP processing.
041	The ACR call to the I/O supervisor detected devices reserved from the failing processor which have to be re-reserved on the operative processor (restartable).
042	The I/O supervisor found devices reserved from failing processor that have no logically online paths from the operative processor.
043	The I/O supervisor found it impossible to reserve a device that was previously reserved to the failing processor.
044	NIP: A machine check interruption occurred on a processor in the configuration.
045	NIP: A BLDCPOOL request for a cellpool in the SQA failed.
046	NIP: An unexpected program check interruption occurred.
047	NIP (for a multiprocessor configuration): A malfunction alert interruption was received by a processor in the configuration.
048	NIP: The subroutine, IEAVPCB, returned an undefined return code.
049	A device was released that was attached to an unavailable channel, and this device was found to be reserved by the processor receiving the machine check interruption.
04A	NIP: The TOD clock is in the error state.
04B	Channels have become temporarily unusable without reserves outstanding (restartable).
04C	IOS: A machine check interruption has occurred indicating one or more channels are at least temporarily unavailable (restartable).
04D	IOS: An I/O interruption has occurred indicating that a channel is unusable (restartable).
04E	IOS: An internal error has occurred while attempting to recover from an I/O related hardware problem (not restartable).

Code	Explanation
04F	More channels were lost after processing was initiated for temporarily unusable channels (not restartable).
050	ACR was recursively invoked.
051	ACR encountered an unrecoverable error while recovery was in progress for another processor (probable software error).
052	ACR encountered an unrecoverable error while recovery was in progress for another processor (probable hardware error).
058	NIP: The subroutine, IEAVSPSA, returned a nonzero code indicating either that it was unable to obtain or free the SALLOC lock, or that a GETMAIN or FREEMAIN for space in the SQA failed.
059	NIP: A BLDL request returned an undefined return code.
05A	NIP: The IPL processor failed to set a synchronization flag for a processor that it had signaled.
05B	NIP: In a multiprocessor configuration, a non-IPLed processor was signaled by the IPLed processor to begin initializing itself. The non-IPLed processor is a possible "run-away".
05C	NIP: Could not obtain a pointer to the master catalog from the SYS1.NUCLEUS data set.
05D	NIP: A DSCB for the master catalog was not found or could not be read from the VTOC of the volume pointed to by the SYS1.NUCLEUS data set.
05E	NIP: An I/O error occurred while attempting to read the master catalog's self-describing records.
05F	NIP: The master catalog's self-describing records are damaged.
060	NIP/ASM: An error is detected while processing external page table entries or (XPTE) while saving or rebuilding the PLPA. The message, IEA943W, precedes this wait state code.
061	NIP/ASM: An error is detected in the TOD (time of day) clock while executing a store clock instruction.
063	NIP: A conditional GETMAIN failed.
064	NIP: A system error is detected and RTM is entered.
065	NIP: Attempted to execute a type 3 or 4 SVC before it was loaded.
066	IOS: Hot I/O has been detected on a non-DASD device. No reserved devices are on the same channel (restartable).
067	IOS: Hot I/O has been detected on a non-DASD device. Reserved devices are on the same channel (restartable).
068	IOS: Hot I/O has been detected on a channel or a DASD device. No reserved devices are on the same channel (restartable).
069	IOS: Hot I/O has been detected on a channel or a DASD device. Reserved devices are on the same channel (restartable).
06A	IOS: Complete recovery for hot I/O on a channel or DASD device cannot be accomplished (restartable).
06B	IOS: An attempt was made to change channel sets; however, the hardware was unable to connect any channel set, leaving the system with no I/O capability (not restartable).
06C	A machine check interruption occurred indicating a serious error in the channels.
06F	An inoperative 3830 control unit, or a channel error, or an undetermined error was detected along a path used by this system. The device on the failing path is a paging device (restartable).
101	A GETMAIN macro, executed by a program executing in supervisor mode, requested more bytes from the SQA than was available.
102	A GETMAIN macro, executed by a program executing in supervisor mode, requested more pages of real storage from SQA than was available.
A01	Either a predefined number of hard machine check interruptions occurred on the only online processor with I/O capability, or a hard machine check interruption occurred from which recovery was not possible.
A18	IOS received an unsolicited device end interruption for a volume containing a paging data set.
A22	A processor was put into a stopped condition because of disabled console communication processing (restartable).
A23	A program check interruption occurred while the machine check handler was executing on the only available online processor; recovery was impossible.
A24	A loop was detected while the machine check handler was executing on the only available online processor; recovery was impossible.
A25	A machine check interruption occurred on the only available online processor; the interruption code indicates system damage was sustained.
A26	A machine check interruption occurred on the only available online processor; the interruption code is invalid.
A27	A machine check interruption or a program interruption has occurred in a routine externally linked to the machine check handler.

B01	The 3203/3211 utility completed normally.
B02	The 3203/3211 utility detected a control card missing or out of order.
B03	The JOB statement for the 3203/3211 utility is incorrect.
B04	The DFN statement for the 3203/3211 utility is incorrect.
B05	The UCS statement for the 3203/3211 utility is incorrect.
B06	The FCB statement for the 3203/3211 utility is incorrect.
B07	The END statement for the 3203/3211 utility is incorrect.
B0A	An external interruption occurred during the execution of the 3203/3211 utility.
B0B	A program check interruption occurred during the execution of the 3203/3211 utility.
B0C	A machine check interruption occurred during the execution of the 3203/3211 utility.

Code	Explanation
B11	The reader required by the 3203/3211 utility is not online.
B12	The reader required by the 3203/3211 utility is not ready.
B13	The reader required by the 3203/3211 utility is not ready.
B14	A reader channel error occurred during the execution of the 3203/3211 utility.
B15	No device end is indicated on the reader being used by the 3203/3211 utility.
B19	A printer required by the 3203/3211 utility is not online.
B1B	A unit check occurred on the printer being used by the 3203/3211 utility.
B1C	A printer channel error occurred during the execution of the 3203/3211 utility.
B1D	No device end is indicated on the printer being used by the 3203/3211 utility.
CCC	During execution of the QUIESCE function on a uniprocessor, the system is placed in a disabled wait state as part of its normal operation (restartable).
E02	A permanent I/O error occurred while the IBM 2250 display unit was being used as a primary console; no alternate console was available at the time of the error.

ECB Indicators

An ECB (event control block) can be posted with a two-part completion code:

- Bits 1 through 7 are posted by data management and teleprocessing functions. This part of the completion code is described under "ECB" in the "Data Area Descriptions" portion of Volume 2.
- Bits 8 through 31 are posted by all system components and by user-written programs. When a task is abnormally terminated, the ECB for the task is posted with an abnormal system completion code in bits 8 through 19, or with an abnormal user completion code in bits 20 through 31.

Abnormal Completion Codes (001-Fhh HEX)

The following table provides a brief explanation of the system abnormal completion codes. For complete explanations, associated messages, reason codes, and problem determination procedures, refer to *OS/VS Message Library: VS2 System Codes*.

Code	Explanation
00n	*DATA MANAGEMENT CODES*
001	CHECK/GET/PUT: An I/O error was encountered (BDAM/BPAM/BISAM/BSAM/QISAM/QSAM).
002	An error occurred while processing, creating, or opening a data set (BSAM/QSAM/QISAM load mode).
003	READ/PUNCH/PRINT: A 3525-associated data set I/O macro sequence error occurred (BSAM/QSAM).
004	OPEN: Conflicting or invalid DCB parameters (BSAM/QSAM) exist.
005	READ: An invalid DECB (BSAM) exists.
006	The program is not authorized to run a 1419.
008	CHECK: A save area was destroyed during the execution of a SYNAD routine (BSAM).
01h	*DATA MANAGEMENT CODES*
013	OPEN: The OPEN macro could not be executed (BDAM/BSAM/QSAM).
014	CLOSE: The CLOSE macro could not be executed (BDAM).
02h	*BDAM AND RSM CODES*
020	OPEN: The DCBMACRF field of the DCB was invalid; or the data set contained zero extents (BDAM).
028	An I/O error occurred while attempting a page-in operation; or an indeterminate error occurred in a RSM service routine, or a translation error occurred in an ASM routine.
03h	*ISAM CODES*
030	OPEN: The DCBMACRF field of the DCB indicated an invalid mode of operation (BISAM/QISAM).
031	An I/O error occurred during processing. The control program attempted to pass control to a SYNAD routine but either the DCBSYNAD field contained an invalid address for a SYNAD routine (QISAM), or the error occurred after the CLOSE macro instruction was executed.
032	OPEN: The DCBMACRF field of the DCB contained invalid ISAM information (BISAM/QISAM).
033	OPEN: An I/O error occurred while reading the highest level index; or while searching for end-of-file mask; or the DCBMSHI field of the DCB contained an invalid address; or an I/O error occurred while reading last prime data block; or all volumes of a multivolume ISAM data set were not mounted.
034	OPEN: An invalid address exists in the DCBMSWA; or the DCBNCBHI field of the DCB indicated that a too small virtual storage area (BISAM) exists.
035	OPEN: The DCBMSWA and DCBSMSW fields of the DCB indicated too small a virtual storage area (BISAM).
036	OPEN: No space is allocated as the prime area for a data set (BISAM/QISAM).
037	OPEN: Supplied buffers were inadequate for records to be processed (BISAM/QISAM).
038	OPEN: Space for an index area was either exhausted or occupied more than one volume (QISAM).
039	The DCBEODAD field of the DCB did not contain an EOD exit routine address (QISAM).
03A	CLOSE: An I/O error occurred while writing an updated data set control block back to a direct access device (BISAM/QISAM).
03B	OPEN: No records from an indexed sequential data set were processed (QISAM).
03D	OPEN: An error in the VOLUME parameter of DD statement (QISAM/BISAM/BDAM) exists.
03E	OPEN: No space was available for additional records (QISAM).

Code	Explanation
04h	*TCAM CODES*
040	OPEN: An error occurred during the execution of an OPEN macro instruction for a TCAM line group.
041	OPEN: An error occurred during the execution of an OPEN macro instruction for a TCAM message queue data set.
042	OPEN: An error occurred while running with the telecommunications online test executive (TOTE).
043	OPEN: An error occurred during the execution of an OPEN macro instruction for a TCAM application program data set.
044	An error occurred during the execution of the FE common write (COMWRITE) task.
045	An error occurred during the execution of a TCAM message control program (MCP).
046	A TCAM application program data set was left open when the message control program terminated.
047	An unauthorized program requested a restricted SVC.
05h	*GAM CODES*
056	The graphic attention service routine referred to an invalid UCB.
057	The graphic attention service routine referred to an UCB associated with a nongraphic device.
06h	*GAM CODES*
061	CLOSE: The DAR macro was issued for a graphic attention control block which was not specified by the closing task.
062	A routine from the graphics subroutine package for FORTRAN IV, COBOL, and PL/I was not executed.
063	The 2250 operator canceled the graphics program.
07h	*SUPERVISOR CODES*
070	An error occurred during the execution of the Suspend, Resume, or TCTL function.
071	The operator pressed the RESTART key.
072	A task terminated because a required processor was logically offline.
073	SETLOCK: An invalid request was made.
074	An invalid lockword address was specified; or the SETLOCK function was unable to complete.
075	A SRB specified an invalid ASCB.
076	An error occurred in address space creation or initialization.
077	A start or stop MONITOR JOBNAMES, STATUS, or SESSIONS was made with an invalid console ID or ASID for the terminal user.
078	RCT processing received an unacceptable return code from an invoked routine.
079	RCT attention exit processing encountered an unacceptable return code from an invoked routine.
07A	RCT recovery processing was invoked as a result of an unrecoverable error. RCT issued a CALLRTM macro to terminate the address space.
07B	A remote immediate, or remote pending, or direct signal interprocessor communications service request was supplied with an invalid PCCA address.
07C	An error was detected by the supervisor control recovery function.
07D	SETFRR:A FRR is not added to the recovery stack because the stack is full.
07E	IEEVDEV received an invalid parameter.
07F	Execution of the address space recovery verification function resulted in an empty TCB queue.

Code	Explanation
08h	*STORAGE MANAGEMENT (ASM and RSM) CODES*
081	GETMAIN or IEAVITAS passed an incorrect virtual address to IEAVSQA.
082	The lockword ID does not match the caller's ID.
083	ILRGMA: No storage cell was available for a workarea, for an ACE, for an AIE, for an IOE, for a TME, or for a SRB.
084	ILRGMA: The storage cell used for a workarea, an ACE, an AIE, an IOE, a TME, or a SRB was not freed.
085	ASPCT: Virtual storage is unavailable for initialization or expansion.
086	ILRACT01 received an error return code from VSAM record-management macro instruction or from the ILRAPR00 routine.
087	ILRAFS00 received an error return code from FREEMAIN.
08F	IEAVCKEY: An error occurred during the execution of the CHANGKEY macro instruction.
09h	*BTAM CODES*
090	OPEN: A non-communications device was allocated to the DCB.
091	OPEN: An invalid or unsupported transmission control unit was specified in the UCB.
092	OPEN: An invalid or unsupported terminal control or terminal adapter was specified in the UCB.
093	OPEN: An invalid or unsupported terminal was specified in the UCB.
094	OPEN: An invalid or unsupported optional feature or mode of operation was specified in the UCB.
095	OPEN: Lines allocated to a line group did not have the identical terminal types and/or optional features.
096	OPEN: Buffers could not be dynamically allocated.
097	OPEN: An additional entry could not be placed in device I/O directory.
098	OPEN: The transmission control unit is not a 2701, or not a dual communication interface; or a dual code feature was not specified in the UCB.
099	OPEN: Dynamic buffering is not supported for V = V.
0Ah	*VTAM CODES*
0A7	Either VTAM was unable to schedule a user's TPEND exit routine or, a TPEND exit routine was not specified.
0A8	An invalid RPL or ECB exists.
0A9	A bad DVT was supplied by the user; or a VTAM lock was held on TPEXIT; or a request for storage was not granted; or storage was not released; or the local address space lock was not obtained; or a request for a device/logical unit was invalid; or control was not passed to proper module, or a HALTNET, CANCEL command was processed.
0AA	VTAM's FRRs were unable to associate an ABEND condition to a particular TCB.
0AB	TSO/VTAM: An error occurred while a VTIOC module was executing a VTAM macro.
0AC	TSO/VTAM: A TCAS error occurred.
0AD	TSO/VTAM: An error occurred while the VTIOC queue manager was executing a GETCELL or FREECELL macro.
0Bh	*SCHEDULER CODES*
0B0	The SWA manager detected an uncorrectable error.
0B1	Invalid parameters were passed to the master subsystem module.
0B2	An invalid card exists in the MSTRJCL data set.
0B3	An invalid or non-servicable request was sent to a pseudo access method via the RPL/ACB interface.
0B4	The master subsystem was unable to use the SYS1.PROCLIB data set defined by the IEFPDSI card.
0B5	The master subsystem was unable to ATTACH the converter.
0B7	A reconfiguration command processing error occurred.
0B8	An error situation is encountered while processing a START, MOUNT, or LOGON command.
0B9	Started task control received an invalid return code from the master subsystem.
0BA	Started task control or the initiator received an invalid return code from the subsystem interface.
0BB	The initiator was unable to ATTACH the job step task, started task, or logon task.

Code	Explanation
0Ch	*PROGRAM INTERRUPTIONS (0-17) CODES*
0C1	An operation exception occurred.
0C2	A privileged-operation exception occurred.
0C3	An execute exception occurred.
0C4	An protection exception occurred, or a virtual address could not be translated into a real address.
0C5	An addressing exception occurred.
0C6	A specification exception occurred.
0C7	A data exception occurred.
0C8	A fixed-point-overflow exception occurred.
0C9	A fixed-point-divide exception occurred.
0CA	A decimal-overflow exception occurred.
0CB	A decimal-divide exception occurred.
0CC	An exponent-overflow exception occurred.
0CD	An exponent-underflow exception occurred.
0CE	A significance exception occurred.
0CF	A floating-point-divide exception occurred.
0Dh	*PROGRAM INTERRUPTIONS (18-31) CODES*
0D2	A translation specification error occurred.
0D3	A program check interruption (special operator error) occurred.
0Eh	*VIO CODES*
0E1	EXCP: An error occurred while processing the first EXCP against a VIO data set.
0E2	EXCP: An error occurred while processing the first EXCP for a VIO data set after a restart.
0E3	A VIO data set processing error occurred.
0E4	Unable to scratch a VIO data set.
0E5	Unable to disconnect the virtual track buffer from a VIO data set.
0E6	Unable to process with a scratched data set.
0E7	A journaling error occurred at step termination or at a checkpoint.
0E8	A processing error occurred for VIO data sets after an abnormal system or job step task failure.
0Fh	*PROLOGUE AND INTERRUPTION HANDLER CODES*
0F0	An error occurred while processing a machine check interruption with the machine check handler's FRR stack active. The task that took the machine check interruption was abnormally terminated.
0F3	Unrecoverable machine check interruption.
0F6	A SETEVENT service processing error occurred.
0F8	The issuer of a SVC was not in TCB mode, or held a lock, or was disabled.
0F9	Space for a SVRB was not available to honor a SVC request in an address space.
0FA	Space for a SVRB was not available to honor a SVC request in an address space while an ABEND is in process.
0FB	A translation exception error occurred.
0FC	A translation exception error occurred during a TRAS function.
0FD	A type 6 SVC made a cross-address space SRB request.
0FE	An error was detected while RMF was sampling data about the state of the system.
100 to Ehh (See "SVC and Special-Category Abnormal Completion Codes.")	
Fhh	An invalid SVC instruction was issued (0Ahh) or an error occurred during the execution of a SVC routine or an OPEN/CLOSE/EOV problem occurred, or the determination routine failed to recognize the passed abnormal termination code (indicated by message IEC900I).

SVC and Special-Category Abnormal Completion Codes (100 - Ehh)

Codes are listed by the last two digits (SVC number), and include:

- SVC codes (h00 - h7B)
- System restart codes (hF3)
- JES2 codes (hFA)
- JES3 codes (hFB)
- Supervisor codes (hFC)
- MF/1 codes (hFD)
- Miscellaneous codes (hFE)

Code	Explanation
h00	*SVC 0 (0A00) CODES*
200	EXCP processing encountered an error when checking the validity of an I/O request.
300	A DEB failed the validity check; or on IOBM value is greater than the value in the DEBEXSCL field of the DEB.
400	The DCB address in the DEB differs from the DCB address in the IOB.
500	An invalid UCB is specified in the DEB.
700	A program check interruption occurred in a supervisor service routine.
800	An error occurred while fixing or unfixing a page during EXCP processing, or the IDA bit is ON in a virtual channel program.
A00	A program check interruption occurred while processing an I/O request.
B00	A program check interruption occurred in an EXCP procedure during EXCP processing of an I/O request.
E00	An IOSB is returned with a completion code of 045 due to a program check interruption in IOS.
h01	*SVC 1 (0A01) CODES*
101	The problem program specified more WAIT events than ECBs.
201	The WAIT macro expansion contained an invalid ECB address.
301	The WAIT macro specified an ECB whose WAIT flag was already "ON".
h02	*SVC 2 (0A02) CODES*
102	The POST macro encountered an invalid ECB address.
202	The POST macro encountered an invalid RB address in the specified ECB.
402	An invalid EVENTS table address exists in the specified ECB.
502	The EVENTS table is already full.
702	The POST failed attempting to process a POST exit request.
h03	*SVC 3 (0A03) CODES*
A03	When the task attempted to terminate, one or more subtasks were not yet terminated.
C03	A data set could not be closed by end-of-task processing.
E03	A "must complete" was not reset before the task attempted termination.
h04	*SVC 4 (0A04) CODES*
104	The GETMAIN macro requested more bytes of virtual storage from the SQA than were available in real storage.
304	An asynchronous cross-address space POST failed. (Not applicable to SVC 4.)
504	A GETMAIN macro requested storage where the length list and the address list would occupy overlapping storage.
604	A GETMAIN macro had an invalid address in the A or LA operand; or a GETMAIN macro had an invalid parameter list address.
704	An uncorrectable machine, system, or indeterminate error occurred while processing a GETMAIN macro.
804	A GETMAIN was issued for which there was not enough virtual storage available; or a negative length was specified.
B04	A GETMAIN macro contained a subpool specification error.

Code	Explanation
h05	*SVC 5 (0A05) CODES*
305	A FREEMAIN macro contained a subpool specification error.
505	A FREEMAIN macro requested the release of storage where the length list and the address list would occupy overlapping storage locations.
605	A FREEMAIN macro had an invalid address in A of LA operand; or the FREEMAIN macro had an invalid parameter list address.
705	An uncorrectable machine, system, or indeterminate error occurred while processing the FREEMAIN macro.
905	The address of the storage area specified in the FREEMAIN macro was not on a doubleword boundary.
A05	The area to be freed by the FREEMAIN macro overlapped a free area in virtual storage, or part of the area was still fixed in real storage.
B05	A FREEMAIN macro contained a subpool specification error.
D05	FREEMAIN macro attempted to free an unowned storage area.
h06	*SVC 6 (0A06) CODES -- Applicable to SVC 6 through SVC 9*
106	LINK, LOAD, ATTACH, or the XCTL macro requested a program which could not be brought into virtual storage.
206	LINK, LOAD, XCTL, or the DELETE macro had an error in its parameter list.
306	LINK, LOAD, ATTACH, or the XCTL macro requested a module that was not found on the authorized library but was on an unauthorized library, or the module name was in a CDE entry whose major CDE was unauthorized.
406	LINK, ATTACH, or the XCTL macro requested that a program be marked "only loadable".
506	LINK, LOAD, ATTACH, or the XCTL macro in an overlay program requested a program that could not be loaded. (NOT applicable to SVC 6, 7, 8, or 9.)
706	LINK, ATTACH, LOAD, or the XCTL macro requested that a module be marked "not executable".
806	LINK, XCTL, ATTACH, or the LOAD macro requested a program or a SVC routine that could not be found; or an I/O error occurred while processing, or an invalid DEB was detected.
906	The maximum use count was exceeded for the LINK, LOAD, ATTACH, or XCTL macro, or the maximum use count and the maximum responsibility count are both exceeded.
A06	A prior request for a module already queued for this task is now being requested by the LINK, LOAD, ATTACH, or XCTL macro.
B06	The system error task, operating on behalf of a user task, abnormally terminated. (NOT applicable to SVC 6, 7, 8, or 9.)
h0A	*SVC 10 (0A0A) CODES*
10A	A R-Type GETMAIN macro requested more bytes of virtual storage from the SQA than were available in real storage.
30A	A R-Type FREEMAIN macro contained an error in its subpool specification.
40A	A R-Type FREEMAIN macro specified the release of all of subpool 0, SQA, CSA, LSQA, or an invalid subpool.
70A	An uncorrectable machine, system, or indeterminate error occurred while processing a R-type GETMAIN or FREEMAIN macro.
80A	A R-Type GETMAIN macro specified too much virtual storage or a negative length.
90A	A R-Type FREEMAIN macro instruction specified a storage area not on a doubleword boundary.
A0A	A R-Type FREEMAIN macro specified an area that overlapped a free area in virtual storage, or part of the area was still fixed in real storage.
B0A	The subpool specified by a R-type GETMAIN or FREEMAIN macro is erroneous.
D0A	A R-Type FREEMAIN macro attempted to free an unowned storage area.
h0B	*SVC 11 (0A0B) CODES*
10B	Input parameter errors exist for the TIME service routine.
20B	No usable time-of-day clock is available in the system.

Code	Explanation
h0D	*SVC 13 (0A0D) CODES*
10D	The jobstep task was terminated with a subtask in "step-must-complete" status.
20D	An error occurred while a subtask was operating in "step-must-complete" status; or the subtask issued an ABEND macro with the "STEP" option.
40D	Recursive errors during the termination of a task caused address space termination to take place.
50D	The program tried to use the OPEN, OPEN-J, CLOSE, CLOSE-T, EOV, or FEOV function while holding exclusive control of a resource defined in the TIOT.
C0D	A routine found a condition that should not occur.
D0D	RTM2 was invalidly reentered.
h0E	*SVC 14 (0A0E) CODES*
10E	The SPIE macro encountered an invalid PICA address.
20E	The SPIE macro encountered an invalid PIE address.
30E	The SPIE macro in an unauthorized program specified interruption code 17; or the SPIE caller was in supervisor state; or the SPIE caller's PSW key does not match the caller's TCB key.
h0F	*SVC 15 (0A0F) CODE*
20F	A non-supervisor state module requested the use of SVC 15 in IOS.
h12	*SVC 18 (0A12) CODES*
112	The BLDL or FIND macros passed invalid input to SVC 18.
212	SVC 18 encountered an error not caused by invalid user input.
h13	*SVC 19 (0A13) CODES*
113	No JFCB extension block was found, or no JFCB exit was found for the OPEN TYPE=J macro instruction.
213	The OPEN macro could not be executed for a data set on a direct access device.
313	The OPEN macro could not be executed for a data set on a direct access device.
413	The OPEN macro could not be executed for a data set on a magnetic tape or direct access device.
513	The tape device specified by the OPEN macro already has an open data set on it.
613	An I/O error occurred while positioning, reading, or writing on a magnetic tape.
713	The expiration data of the data set specified by the OPEN macro had not yet been reached.
813	The data set name in the header label did not match that in the JFCB (occurs only when using the OPEN macro for tape).
913	While processing an OPEN macro, an error occurred involving a password-protected data set or a checkpointed data set.
A13	The data set requested by the OPEN macro is not found (tape only).
B13	The OPEN macro could not be executed for a data set on a 1403 printer.
C13	The OPEN macro could not be executed for a concatenated partitioned data set or a graphics device.
D13	The OPEN macro could not be executed for a graphics DCB when the DCB is associated with a non-graphics device.
E13	The OPEN macro could not be executed for a graphics DCB when the DCBGNCP field contains an invalid value.
h14	*SVC 20 (0A14) CODES*
214	The CLOSE macro could not be executed for a data set on a magnetic tape(I/O error).
314	The CLOSE macro could not be executed for a data set on a direct access device(I/O error).
414	The CLOSE macro encountered an I/O error while writing a DSCB.
614	The CLOSE macro detected an error while attempting to write a file mark.
714	The CLOSE macro detected an I/O error while writing a trailer label or a tape mark.
A14	The CLOSE macro encountered an I/O error while attempting a partial release of space on a direct access device.
B14	The CLOSE macro could not be executed for a partitioned data set opened for output to a member.
D14	The CLOSE macro for a graphics DCB could not be executed.

Code	Explanation
h15	*SVC 21 (0A15) CODES*
115	A DCB did not point to a valid DEB; or a valid DEB did not point back to the correct DCB.
215	An error exists in the contents of the DCB or the parameter list being used.
315	A STOW error occurred that was not caused by invalid user input.
h17	*SVC 23 (0A17) CODES*
117	An I/O error occurred while executing a TYPE=T, CLOSE macro (BSAM).
317	The TYPE=T, CLOSE macro encountered an I/O error reading the DSCB, on a direct access device.
417	The TYPE=T, CLOSE macro encountered an I/O error while writing an updated DSCB (BSAM).
717	The TYPE=T, CLOSE macro detected an I/O error while writing a trailer label or a tape mark.
h18	*SVC 24 (0A18) CODES*
118	Invalid parameters were passed to DEVTYPE.
218	An error occurred that was not caused by invalid user input.
h19	*SVC 25 (0A19) CODES*
119	Invalid input was passed to SVC 25.
219	An error occurred that was not caused by invalid user input.
h1A	*SVC 26 (0A1A) CODES*
11A	A protection check occurred attempting to store into a user-supplied OS/VS CAMLST work area.
h22	*SVC 34 (0A22) CODES*
122	The operator canceled the job and requested a dump.
222	The operator canceled the job.
322	The job, job step, or cataloged procedure exceeded the specified time.
522	An ECB specified in a WAIT request was never posted.
622	A task activated by a TSO terminal was not initiated.
722	The specified output limit was exceeded.
822	The region required to run the step could not be obtained.
922	ABEND, program check interruption, machine check interruption, or depression of the RESTART key occurred when the initiator was executing.
A22	The operator entered the FORCE command for the job.
h23	*SVC 35 (0A23) CODES*
A23	A cross-address space POST using an invalid ECB address was specified by the communications task.
B23	A WTOR went unanswered.
D23	An invalid parameter list exists, or subpool space is unavailable.
E23	An ECB address, RB address, or virtual storage area address is invalid.
h28	*SVC 40 (0A28) CODES*
128	The EXTRACT macro received an invalid address at which to store the requested fields.
228	The EXTRACT macro received an invalid parameter list address.
328	The TCB specified in the second operand of EXTRACT macro is not a subtask of the task issuing the EXTRACT.
h2A	*SVC 42 (0A2A) CODES*
12A	A task attempted to give a shared subpool to a subtask.
22A	An invalid subpool was specified on the ATTACH macro.
42A	The ECB operand of the ATTACH macro is specified with an invalid ECB address.
52A	Insufficient LSQA storage exists to copy the necessary STAI information.
72A	An invalid parameter address is used in the ATTACH macro.
82A	The ATTACH macro specified an invalid subpool ID.
92A	An environmental error occurred in SVC 60, in subpool 255, or in SETLOCK.

Code	Explanation
h2C	*SVC 44 (0A2C) CODES*
12C	An invalid address for a subtask TCB was supplied in the CHAP macro.
22C	An invalid TCB address was specified in the CHAP macro.
h2D	*SVC 45 (0A2D) CODES*
12D	Words 3 and 4 of the segment table are incorrect.
22D	An invalid address exists in the segment table or the entry table.
32D	A record of incorrect length was found; or an uncorrectable I/O error occurred while loading a segment from a library.
42D	An error occurred in SEGLD processing.
C2D	An invalid scatter record exists.
D2D	An invalid record type exists.
E2D	An invalid address was found.
h2E	*SVC 46 (0A2E) CODES*
12E	Erroneous input parameters exist for the TTIMER service routine.
22E	No usable time-of-day clock was available for TTIMER service routine.
h2F	*SVC 47 (0A2F) CODES*
12F	Erroneous input parameters exist for the STIMER service routine.
22F	No useable combination of the time-of-day clock and clock comparator or no usable processor timer is available for the STIMER service routine.
h30	*SVC 48 (0A30) CODES*
130	A resource was specified for DEQ that had not been previously specified by ENQ under the same task.
230	An invalid length was specified for the name representing a resource being dequeued.
330	An unauthorized task attempted to use the authorized options of the DEQ macro.
430	An invalid parameter list was used for the DEQ macro.
530	A task issued an ENQ macro for a resource; Before the ENQ request was filled, this same task issued a DEQ macro for the same resource.
630	An out-of-storage condition occurred during the execution of the DEQ macro.
730	An unexpected error occurred during the processing of a DEQ request.
h33	*SVC 51 (0A33) CODES*
133	An unauthorized caller invoked an SVC dump via the SDUMP macro.
233	Invalid parameters were passed to SVC dump.
h35	*SVC 53 (0A35) CODES*
135	Exclusive control of a resource could not be obtained or released.
235	A control block required as input to SVC 53 was not in the user's region and/or did not have the user's storage key.
335	The data set does not contain the block for which acquisition or release of exclusive control is requested.
435	An input DCB points to an invalid DEB, or the IOB has been altered by other than a system routine.
535	No storage was available for the GETMAIN issued by SVC 53.
h37	*SVC 55 (0A37) CODES*
137	An I/O error occurred during the end-of-volume processing for a magnetic tape; or an invalid header or a trailer label was read.
237	An invalid block count or dsname exists.
337	No EODAD routine was specified in the DCB when an EOF occurred.
437	An invalid DCB pointer exists.
637	An I/O error occurred during end-of-volume processing for a data set on magnetic tape, or an error occurred during concatenation.
737	An I/O error occurred during end-of-volume processing, or during the allocation of a secondary quantity of direct access storage.
837	An error occurred during end-of-volume processing for a sequential data set.
937	An error occurred during end-of-volume processing for a checkpoint data set, or a RACF protected data set.

Code	Explanation
A37	The DCB was not open when end-of-volume processing took place.
B37	A volume cannot be demounted; or an invalid VTOC could not be converted; or a direct access volume contained the maximum number of users.
C37	An end-of-volume condition occurred on a data set opened for input, but the next volume of the data set contained more than 16 extents. Or a type of split cylinder allocation exists that cannot be processed under OS/VS.
D37	No secondary space was requested.
E37	No more volumes were available; or a DSCB with a duplicate data set name on the next volume was found.
h38	*SVC 56 (0A38) CODES*
138	Two ENQ macro instructions were issued for the same resource in the same task without an intervening DEQ macro instruction.
238	An invalid length was specified on an ENQ macro for the name representing the resource.
338	An unauthorized task attempted to use an authorized ENQ option.
438	An invalid parameter list was created for the ENQ macro.
638	An out-of-storage condition occurred during the execution of an ENQ macro.
738	An unexpected error occurred during the execution of an ENQ macro.
838	Serious damage occurred to the ENQ/DEQ control blocks.
h39	*SVC 57 (0A39) CODES*
139	An error occurred while freeing a buffer or extending the unscheduled test.
239	A control block required as input to SVC 57 was not in the user's region and/or did not have the user's storage key.
339	The DEB pointed to by the DCBDEBAD field of the DCB was not on a DEB chain for that job step TCB, or did not point back to that DCB. Or the request was issued for a data set which was neither BDAM or ISAM.
h3C	*SVC 60 (0A3C) CODES*
13C	An invalid request was detected.
h3E	*SVC 62 (0A3E) CODES*
13E	The calling task issued a DETACH macro specifying STAE=NO before a subtask had terminated.
23E	The DETACH macro detected a storage key error or an invalid parameter.
33E	The DETACH macro specifying STAE=YES was issued by the originating task before a subtask had completed execution.
43E	An invalid ECB address was specified for the ATTACH macro.
53E	An error occurred during end-of-task processing.
h3F	*SVC 63 (0A3F) CODES*
13F	A checkpoint restart error occurred.
23F	An unauthorized user has attempted to read or write a checkpoint data set.
h40	*SVC 64 (0A40) CODES*
240	The RDJFCB macro detected an error in a DCB.
h44	*SVC 68 (0A44) CODES*
144	An invalid access method code was specified in the high order byte of register 15.
244	An invalid register save area was provided by the user.
344	An invalid DCB address was used.
444	An invalid DECB address was used.
544	An invalid IOB address was used.
644	The message buffer was modified or freed.
744	A program check interruption occurred that was not caused by invalid user input.

Code	Explanation
h45	*SVC 69 (0A45) CODES*
145	An invalid DCB or ICB is used.
245	An error occurred that was not caused by invalid user input.
h4F	*SVC 79 (0A4F) CODES*
14F	The STATUS macro was used for a function other than STOP, STOP SYNCH, or START, and the user was not in supervisor key (0-7).
h51	*SVC 81 (0A51) CODES*
151	An invalid parameter list address was passed to the SETPRT macro.
251	The SETPRT macro detected an invalid DCB exit list.
351	The SETPRT macro could not be executed.
451	The SETPRT macro detected an invalid DCB, invalid IOBs, an invalid DEB address, or a DCB exit list address.
h53	*SVC 83 (0A53) CODES*
153	An error occurred in the cross-address space communications for SMF record processing.
h57	*SVC 87 (0A57) CODES*
157	The issuer of a DOM request is not in the same address space and is not a privileged task.
h5C	*SVC 92 (0A5C) CODES*
15C	The program issuing SVC 92 was not in supervisor state.
h5D	*SVC 93 (0A5D) CODES*
15D	An invalid data area is specified in the TGET or TPUT macro and detected by SVC 93.
h5F	*SVC 95 (0A5F) CODES*
15F	The system resources manager is invalidly invoked.
25F	A system failure destroyed the data used by the system resources manager.
35F	An address space was non-swappable when a SYSEVENT, TRANSWAP (X'0E'), was issued.
45F	A quiese or swap-out failed while a transition swap (TRANSWAP) or requested swap (REQSWAP) was pending.
55F	An error was encountered during the execution of the XMPOST SRB for an issuer of SYSEVENT, TRANSWAP or REQSWAP.
h60	*SVC 96 (0A60) CODES*
260	The STAX SVC detected an invalid user parameter.
h69	*SVC 105 (0A69) CODES*
169	The CLOSE macro passed an invalid DCB address to IMGLIB.
269	An error occurred during the execution of the IMGLIB macro.
h6B	*SVC 107 (0A6B) CODES*
16B	An invalid parameter list exists.
h6D	*SVC 109 (0A6D) CODES*
16D	An invalid ESR code for SVC 109, 116, or 122 was detected.
26D	MF/1 could not initialize, collect, or reinitialize the workload activity measurements.
36D	A communications error occurred between RMF and the system resources manager.
h6E	*SVC 110 (0A6E) CODES*
16E	The DEBCHK function was not completed. (Applies to SVC 117, not SVC 110.)

Code	Explanation
h6F	*SVC 111 (0A6F) CODES*
36F	An invalid entry code was passed to SVC 111, or DEBCHK processing found that the DEB was invalid, or the DEB was not a subsystem DEB.
56F	An error occurred during end-of-job processing for a JES2 job that was being processed by an execution batch monitor.
h71	*SVC 113 (0A71) CODES*
171	An illegal or invalid request is made for the PGFIX, PGFREE, PGLOAD, or PGOUT macro.
271	An unauthorized call is made for the PGFIX or PGFREE macro.
h72	*SVC 114 (0A72) CODE*
172	An SVC 114 was issued with one of the following conditions: the protect key was not correct (not 0 thru 7), the request was not issued in the supervisor state, or the authorization bit was not set in the JSCB.
h75	*SVC 117 (0A75) CODES*
—	(See completion code 16E)
h77	*SVC 119 (0A77) CODES*
177	Invalid input parameters were detected by TESTAUTH.
h78	*SVC 120 (0A78) CODES*
178	The RC or RU type GETMAIN macro requested more bytes of virtual storage from the SQA than were available in real storage.
378	The RC or RU type FREEMAIN macro contained an error in its subpool specification.
478	The RC or RU type FREEMAIN macro contained an error in its subpool specification.
778	An uncorrectable machine, system, or indeterminate error occurred while processing the RC or RU type GETMAIN or FREEMAIN macro.
878	Not enough virtual storage is available for the GETMAIN issuer, or the RC or RU type GETMAIN issuer specified a negative length.
978	The address of the storage area specified by the RC or RU type FREEMAIN macro was not on a doubleword boundary.
A78	An error occurred in defining the area to be freed by the RC or RU type FREEMAIN macro.
B78	The RC or RU type GETMAIN or FREEMAIN macro contained a subpool specification error.
D78	The RC OR RU type FREEMAIN macro attempted to free an unowned storage area.
h79	*SVC 121 (0A79) CODES*
179	Any one of the following conditions: • PGFIX or SETLOCK fail. • The conversion to a real address (honoring a LRA instruction) fails. • The block size for track overflow is not 4k.
279	An invalid BUFC or protection check is indicated in the IOSB by IOS.
h7A	*SVC 122 (0A7A) CODES*
17A	An invalid EVENTS table address was specified.
37A	A request was issued for an EVENTS table being waited on by another program.
47A	The ENTRIES specification is incorrect (not from 1 to 32,767).
h7B	*SVC 123 (0A7B) CODES*
17B	An invalid ASID was specified when the PURGEDQ SVC was issued.
27B	A PURGEDQ received a non-zero return code from FESTAE.

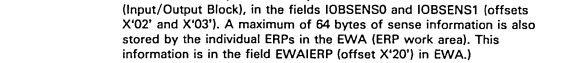
Code	Explanation
h7D	*SVC 125 (0A7D) CODES*
17D	An invalid EVENTS table address was specified.
37D	A request was issued for an EVENTS table being waited on by another program.
47D	An invalid LAST parameter was specified.
57D	An invalid ECB address was specified.
67D	A "WAITED ON" ECB is trying to be initialized.
77D	Invalid input options were specified.
87D	The event table is full.
h82	*SVC 130 (0A82) CODES*
E82	An error occurred while disabled. An inactive SVC was invoked.
h83	*SVC 131 (0A83) CODES*
E83	An error occurred while disabled. An inactive SVC was invoked.
h84	*SVC 132 (0A84) CODES*
E84	An error occurred while disabled. An inactive SVC was invoked.
h85	*SVC 133 (0A85) CODES*
E85	An error occurred while disabled. An inactive SVC was invoked.
hF3	*SYSTEM RESTART CODES*
2F3	A system failure occurred.
hFA	*JES2 CODES*
1FA	Invalid data management control blocks were found by JES2.
hFB	*JES3 CODES*
1FB	An error condition exists that was caused by either module, IATDMEB, or module, IATSIDM.
2FB	The JES3 address space abnormally terminated.
3FB	Sufficient storage could not be obtained in subpool 241 to build a DSB or to create an RAB.
4FB	Sufficient storage could not be obtained in subpool 241 for the SSISERV; or sufficient storage could not be obtained in subpool 230 for user address space buffers; or an error occurred while processing TQE.
5FB	An error occurred during the execution of an MVS-JES3 allocation subsystem interface routine.
6FB	An error condition exists. The reason code is in the high-order byte of register 15; the return code is in the low-order byte of register 15.
7FB	An error occurred during the execution of the MVS JES3 dynamic device reconfiguration (DDR) subsystem interface routines.
8FB	An error occurred in one of the interpreter service non-JES3 support areas.
hFC	*SUPERVISOR CODES*
1FC	An error occurred in executing a SVC first level interruption handler.
2FC	An error occurred while executing the I/O first level interruption handler.
3FC	An error occurred while executing the external first level interruption handler.
4FC	An error occurred while executing the program check first level interruption handler.
5FC	An error occurred while executing the restart first level interruption handler.
6FC	An error occurred while a SPIE exit routine was being set up by the program-check first level interruption handler.
hFD	*MF/1 CODES*
1FD	An error was detected by MF/1 while sampling data about the state of the system.

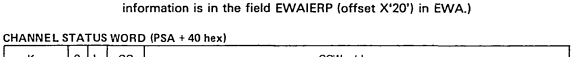
I/O Error Indicators

I/O errors are indicated in:

- The channel status word (CSW)
- The limited channel logout area
- The UCB sense bytes (Sense bytes 0 and 1 are also stored in the IOB (Input/Output Block), in the fields IOBSENS0 and IOBSENS1 (offsets X'02' and X'03'). A maximum of 64 bytes of sense information is also stored by the individual ERPs in the EWA (ERP work area). This information is in the field EWAIERP (offset X'20') in EWA.)

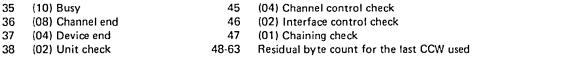
CHANNEL STATUS WORD (PSA + 40 hex)





5	Logout pending	40	(80) Program-controlled interruption
6-7	Deferred condition code	41	(40) Incorrect length
32	(80) Attention	42	(20) Program check
33	(40) Status modifier	43	(10) Protection check
34	(20) Control unit end	44	(08) Channel data check
35	(10) Busy	45	(04) Channel control check
36	(08) Channel end	46	(02) Interface control check
37	(04) Device end	47	(01) Chaining check
38	(02) Unit check	48-63	Residual byte count for the last CCW used
39	(01) Unit exception		

LIMITED CHANNEL LOGOUT (PSA + B0)



4	Processor	12	Control unit	22	Channel address
5	Channel	15	Limited channel logout is valid (not used on all models)	23	Device address
6	Storage control unit			24-25	Type of termination
7	Storage unit	16	Interface address	00	Interface disconnect
8	Processor	17-18	Reserved (00)	01	Stop, stack or normal
9	Channel	19	Sequence code	10	Selective reset
10	Main storage control	20	Unit status	11	System reset
11	Main storage	21	Command address and key	27	Interface operative
				28(A)	I/O error alert
				29-31	Sequence code

Sense Bytes

The sense bytes are found in the EWA data area at offset X'02'. The first two sense bytes can also be found in the IOB data area at offset X'20', and in the IOSB data area at offset X'2A'. Also, error message IEA000I, when issued, contains sense byte data.

The following abbreviations are used in the chart on the next pages.

ACCT	Account
ADDR	Address
ADV	Advance
ALU	Arithmetic logic unit
AMT	Amount
ATT	Attention
BK	Back
BKWD	Backward
BLK	Block
BOC	Branch on condition
BPI	Bits per inch
BR	Branch
BUF	Buffer
CARR	Carriage
CH	Channel
CHAR	Character
CHK	Check
CMD	Command
CNT	Count
CNTRL	Control
CNVTT	Converter
COMP	Compare
COMPT	Compatible
CORR	Correction
CRC	Cyclic redundancy check
CU	Control unit
CYL	Cylinder
DEN	Density
DET	Detected
DIAG	Diagnostic
DOC	Document
DUCT	Dueling and uncomparable character table
EC	Engineering change
ENG	Engineering
ENV	Envelope; Environmental
EOD	End of data
EOT	End of tape
EQ	Equipment
ERR	Error
EXP	Expected
FCB	Forms control buffer
FEA	Feature
FLT	Fault
FRU	Field replacable unit
FUNC	Function/Functional
FUT	Future
HAM	Hammer
HRDWR	Hardware
IBG	Interblock gap
IC	Instruction counter
ID	Identification
INC	Incomplete
IND	Indicator
INIT	Initialized
INSTR	Instruction
INT	Intervention
INTRR	Interrupt
INVAL	Invalid
KYBD	Keyboard
LRC	Longitudinal redundancy check
LRCR	Longitudinal redundancy check register
LSR	Local storage register
LWR	Loop write to read
MBSB	Missing basic status bit
MP	Multiprocessor
MTE	Multi-track error
NON-XST	Non-existent
NRZI	Non-return to zero indiscrete
OP	Operation
OPR	Operator
OVRN	Overrun
PAR	Parity
PCU	Primary control unit
PE	Phase encoded
PERM	Permanent
PGS	Progress
PLB	Print line buffer
POSN	Position
POSNG	Positioning
PRED	Predictor
PRO	Process
PROC	Processor
PROT	Protection
RCP	Recognition and control processor
RCVY	Recovery
RD BK	Read back
RDY	Ready
REC	Record
REG	Register
REJ	Reject
REQ	Required
REST	Restore
RESVD	Reserved
REV	Reversal
RIB	Ribbon
ROS	Read-only storage
RPQ	Request for price quotation
RPS	Rotational position sensing
RQST	Request
RWD	Rewind
RWDNG	Rewinding
R/W	Read/write
SCU	Secondary control unit
SEL	Selected
SEP	Separator
SEQ	Sequence
SPUR	Spurious
SRT	Special record transferred
STAT	Status
STKD	Stacked
STKR	Stacker
SUP	Suppression
SW	Switch
SYNC	Synchronous
TGR	Trigger
TI	Tape indicate
TM	Tape mark
TRK	Track
TU	Tape unit
UCSAR	Universal character set address register
UCSB	Universal character set buffer

VEL	Velocity	WRT	Write
VRC	Vertical redundancy check	WTM	Write tape mark

BYTE 0

BIT / DEVICE	0	1	2	3	4	5	6	7
33, 35, 1030, 1050	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	LOST DATA	TIME OUT
1052, 2150	CMD REJ	INT REQ	BUS OUT	EQ CHK				
1287	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	NON RCVY	KYBD CORR
1288	CMD REJ	INT REQ	BUS BUS	EQ CHK	DATA CHK	OVER-RUN	NON RCVY	SHOULD NOT OCCUR
1403, 1443	CMD REJ	INT REQ	BUS OUT	EQ CHK	TYPE BAR	TYPE BAR		CH 9
1419/1275 PCU	CMD REJ	INT REQ	BUS OUT		DATA CHK	OVER-RUN	AUTO SELECT	NOT USED
1419/1275 SCU	CMD REJ	INT REQ	BUS OUT CHK			LATE STKR SELECT	AUTO SELECT	OP/ ATT
2021, 2540	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK		UN-USUAL CMD	
2250	CMD REJ	SHOULD NOT OCCUR	BUS OUT	EQ CHK	DATA CHK	SHOULD NOT OCCUR	BUFFER RUN-NING	SHOULD NOT OCCUR
2260	CMD REJ	INT REQ	BUS OUT	EQ CHK	SHOULD NOT OCCUR	SHOULD NOT OCCUR	SHOULD NOT OCCUR	SHOULD NOT OCCUR
2305	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN		
2314, 2319	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN		INVAL ADDR
2400	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	WORD CNT ZERO	DATA CNVTT CHK
2495	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	SHOULD NOT OCCUR	POSN CHK	SHOULD NOT OCCUR
2501, 2520	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN		
2671	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK			
2701, 2702	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	LOST DATA	TIME OUT
2715	CMD REJ		BUS OUT	EQ CHK			LOST DATA	TIME OUT
2803, 2804	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	WORD CNT ZERO	DATA CNVTT CHK
3203, 3211	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	BUFFER PARITY CHK	LOAD CHK	CH 9
3210, 3215	CMD REJ	INT REQ		EQ CHK				
3270	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	UNIT SPEC	CNTL CHK	OP CHK
3330/3333 Mod I, Mod II	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN		

BYTE 0 (Continued)

BIT / DEVICE	0	1	2	3	4	5	6	7
3340/3344	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	TRK COND CHK	SEEK CHK
3350	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	TRK COND CHK	SEEK CHK
3410/3411	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	WORD CNT ZERO	DATA CNVTT CHK
3420/3803	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	WORD CNT ZERO	DATA CNVTT CHK
3505, 3525	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK		ABNOR-MAL FORMAT RESET	PER-MANENT ERROR (BYPASS KEY)
3540	CMD REJ	INT REQ	BUS OUT CHK	EQ CHK	DATA CHK			
3800	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK		LOAD CHK	CH9
3886	CMD REJ	INT REQ	BUS OUT	EQ CHK			NON INIT	RCP ERR
3890	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK		NON INIT	RUN-NING
3895	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK		NON INIT	RUN-NING
7770	CMD REJ		BUS OUT	EQ CHK	DATA CHK	OVER-RUN		

BYTE 1

BIT / DEVICE	0	1	2	3	4	5	6	7
33, 35, 1030, 1050								
1052, 2150								
1287	TAPE MODE	LATE STKR SELECT	NO DOC FOUND	SHOULD NOT OCCUR	INVAL OP	SHOULD NOT OCCUR	SHOULD NOT OCCUR	SHOULD NOT OCCUR
1288	SHOULD NOT OCCUR	END OF PAGE	NO DOC FOUND	SHOULD NOT OCCUR	INVAL OP	SHOULD NOT OCCUR	SHOULD NOT OCCUR	SHOULD NOT OCCUR
1403, 1443								
1419/1275 PCU			DOC UNDER READ HEAD	AMT FIELD VALID	PRO-CESS CNTRL FIELD VALID	ACCT # FIELD VALID	TRANSIT FIELD VALID	SERIAL# FIELD VALID
1419/1275 SCU								
2021, 2540								
2250	LIGHT PEN DETECT	END ORDER SEQ	CHAR MODE			2840 OUTPUT CHECK	2840 INPUT CHECK	
2260								

BYTE 1 (Continued)

BIT / DEVICE	0	1	2	3	4	5	6	7
2305	PERM ERR	INVAL TRK FORMAT	END OF CYL		NO REC FOUND	FILE PROT		OPER-ATION INL
2314, 2319	DATA CHK IN COUNT	TRK OVER-RUN	END OF CYL	INVAL SEQ	NO REC-FOUND	FILE PROT	SERVICE OVER-RUN	OVER-FLOW INL
2400	NOISE	00–NON-XST TU 01–NOT READY 10–RDY AND NO RWD 11–RDY AND RWDNG		7 TRK	AT LOAD POINT	WRT STATUS	FILE PROT-ECT	NOT CAP-ABLE
2495								
2501, 2520								
2671								
2701, 2702								
2715								
2803, 2804	NOISE	TU STATUS A	TU STATUS B	7 TRK	LOAD POINT	SEL & WRT STATUS	FILE PROT	NOT COMP
3203	CMD RETRY	PRT CHK		LINE POS CHK	FORMS CHK	CMD SUP	CNTRLR CHK	
3210, 3215								
3211	CMD RETRY	PRINT CHK	PRINT QUAL-ITY	LINE POSN	FORMS CHK	CMD SUP	MECHA-NICAL MOTION	
3270								
3330/3333 Mod I, Mod II	PERM ERR	INVAL TRK FORMAT	END OF CYL	STATE VAR PRES	NO REC FOUND	FILE PROT	WRITE INHIBIT	OPER-ATION INC
3340/3344	PERM ERR	VAL TRL FORMAT	END OF CYL		NO REC FOUND	FILE PROT	WRITE INHIBIT	OPER-ATION INC
3350	PERM ERR	VAL TRL FORMAT	END OF CYL		NO REC FOUND	FILE PROT	WRITE INHIBIT	OPER-ATION INC
3410/3411	NOISE	TU STATUS A	TU STATUS B		AT LOAD POINT	WRT STATUS	FILE PROT-ECT	NOT CAP-ABLE
3420/3803	NOISE	TU STATUS A	TU STATUS B	7 TRK TU	LOAD POINT	WRT STATUS	FILE PROT-ECT	NOT CAP-ABLE
3505, 3525	PERM-ANENT ERROR	AUTO-MATIC RETRY	MOTION MAL-FUNC-TION	RETRY AFTER INT REQ COM-PLETE				
3540	PERM ERR	AUTO-MATIC RETRY		RETRY AFTER INT COM-PLETE	SRT			
3800	BIT MEANINGS DEPENDENT UPON BYTE 0. SEE TABLE OF "3800 SENSE BYTES 0-2"							

BYTE 1 (Continued)

BIT / DEVICE	0	1	2	3	4	5	6	7
3886		MARK CHK	INVAL FORMAT		INCOMP SCAN		NON RECOV-VERY	OUT-BOARD
3890								

BYTE 1 (Continued)

BIT / DEVICE	0	1	2	3	4	5	6	7
3895	PRED REQ	FEA CHK	FEED CMD REQ			INC PRO	MERGE FEED REQ	PROC NOT FUNC
7770								

BYTE 2

BIT / DEVICE	0	1	2	3	4	5	6	7
2250		BUFFER ADDRESS REGISTER BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9
2305	BUF LOG FULL	CORRECTABLE						
2314, 2319	UNSAFE	SHIFT REG CHK	SKEW FAIL	CTR CHK	COMP CHK			
2400	BITS 0 - 7 INDICATE A TRACK IS IN ERROR						6 & 7 INDICATE NO ERROR OR MULTI - ERROR	
3203	CARR FAILED TO MOVE	CARR MOTION CHK				FORMS JAM		TRAIN VEL CHK
2803, 2804	TRACK IN ERROR OR DEAD TRACK REGISTER							
3211	CARR FAILED TO MOVE	CARR SEQ CHK	CARR STOP CHK	PLATEN FAILED TO ADV	PLATEN FAILED TO RETRACT	FORMS JAM	RIBBON MOTION	TRAIN OVERLOAD
3330/3333 Mod I, Mod II		CORRECTABLE		ENV DATA PRESENT	COMP CHK			
3340/3344	RPS FEATURE	CORRECTABLE		ENV DATA	COMP CHK		MODULE SIZE	MODULE SIZE
3350	RPS FEATURE	CORRECTABLE		ENV DATA	COMP CHK		MODULE SIZE	MODULE SIZE
3410/3411	TRACK IN ERROR							
3420/3803	TRACK IN ERROR							
3505, 3525	USED FOR DIAGNOSTIC PURPOSES ONLY							
3540	USED FOR DIAGNOSTIC PURPOSES ONLY							
3800	BIT MEANINGS DEPENDENT UPON BYTE 0 SEE TABLE OF "3800 SENSE BYTES 0-2"							

BYTE 3

BIT / DEVICE	0	1	2	3	4	5	6	7
2250	BUFFER ADDRESS REGISTER BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1
2305	RESTART COMMAND							
2314, 2319	LRC BIT 0	LRC BIT 1	LRC BIT 2	LRC BIT 3				
2400	R/W VRC	LRC	SKEW	CRC	SKEW REQ VRC	0 - 800 1 - 1600	BKWD STATUS	COM-PARE
2803, 2804	R/W VRC	LRC/ MULTI-TASK	SKEW	CRC/ EOD CHK	SKEW REG VRC/ENV CHK		BKWD	C COMP
3203	UCSB PAR	PLB PAR		COIL PROT	HAM FIRE		SYNC CHK	
3211	UCSB PARITY	PLB PARITY	FCB PARITY	COIL PROT CHK	HAM-MER FIRE	FIELD END	UCSAR SYNC CHK	SEP SYNC
3330/3333 Mod I, Mod II	RESTART COMMAND							
3340	RESTART COMMAND							
3350	RESTART COMMAND							
3410/3411	VRC	MTE/ LRC	SKEW	END DATA CHK/ CRCR	ENV CHK	1600 BPI SET IN TU	BKWD	
3420/3803	R/W VRC	MTE/ LRC	SKEW	END DATA CHK/ CRC	VRC ENV CHK	1600 BPI SET IN TU	BKWD	C/P COM-PARE
3505, 3525	USED FOR DIAGNOSTIC PURPOSES ONLY							
3540	TRACK IN ERROR							
3800	PRINTER RDY	PAGE BUFFER EMPTY	BK DATA CHK	PAPER THREAD-ED FOR SHEET STACKER	SYSTEM RESTART RE-QUIRED	PHOTO CON-DUCTOR ADVANCE ENABLED		

BYTE 4

BIT / DEVICE	0	1	2	3	4	5	6	7
2305								
2314, 2319	SEQ IND 0	SEQ IND 1			PHYSICAL DRIVE IDENTIFICATION			
2400	ECHO ERR	REJ TAPE UNIT	READ CLOCK ERR	WRITE CLOCK ERR	DELAY CNTR ERR	SEQ IND C	SEQ IND B	SEQ IND A
2803, 2804		REJ TU		WRT TRC VRC	START RD CHK	SEQ IND C	SEQ IND B	SEQ IND A
3203	ALWAYS X"84"							

Byte 4 (Continued)

BIT / DEVICE	0	1	2	3	4	5	6	7
3330/3333 Mod I, Mod II	PHYSICAL DRIVE IDENTIFICATION							
3340/3344	PHYSICAL DRIVE IDENTIFICATION							
3350	PHYSICAL DRIVE IDENTIFICATION							

Byte 4 (Continued)

3410/3411	TAPE UNIT POSN CHK	TAPE UNIT REJ	TI EOT			DIAG TRK CHK	TAPE UNIT CHK	SPARE
3420/3803	ALU/MP HARD-WARE ERROR	REJ TAPE UNIT	TAPE INDI-CATE	WRITE TGR VRC	MICRO-PRGM DETECT ERROR	LWR ERROR	TAPE UNIT CHK	
3540	BINARY ZERO							
3800	DIAGNOSTIC STATUS CODES							

BYTE 5

BIT / DEVICE	0	1	2	3	4	5	6	7
2305	CYLINDER ADDRESS							
2314, 2319								
2400	COMMAND IN PROGRESS WHEN OVERFLOW INCOMPLETE OCCURS OR ZERO							
2803, 2804	FUT DEN	CE FUNC					DIAG MADE	RPQ
3203	RIB REV	REST KEY	RESET LINE CNTR	SPACE KEY	IDLE STAT	START KEY	STOP KEY	
3330/3333 Mod 1, Mod 11	CYLINDER ADDRESS							
3340	CYLINDER ADDRESS							
3350	LOW ORDER LOGICAL CYLINDER ADDRESS							
3410/3411	NEW SUB-SYSTEM	NEW SUB-SYSTEM	WRT TAPE MARK CHK	ID BURST	PARITY COM-PARE	TACHO-METER CHK	FALSE END MARK	RPQ
3420/3803	NEW SUB-SYSTEM	NEW SUB-SYSTEM	WRT TAPE MARK CHK	ID BURST	START READ CHK	PARTIAL RECORD	EXCESS-IVE POST-AMPLE OR TM	RPQ
3540	SECTOR IN ERROR							
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

BYTE 6

BIT / DEVICE	0	1	2	3	4	5	6	7
2305	CURRENT HEAD ADDR							
3203	INVAL FUNC RQST	HI-SPD FLT	DATA CHK			INVAL I/O CMD	CNTRLR BUS OUT PAR	HALT
3330/3333 Mod I	RE-VERSE	CYL HIGH	DIFFER HIGH	HEAD ADDR				
3330/3333 Mod II		CYL 512	CYL 256	HEAD ADDR				
3340/3344	RE-VERSE	CYL HIGH	DIFFER HIGH	HEAD ADDR				
3350	RE-VERSE	CYL HIGH	DIFFER HIGH	HEAD ADDR				
3410/3411		SHORT GAP MODE	DUAL DEN-SITY	ALTER-NATE DENSITY	TU MODEL			
3420/3803	7 TRK	WRITE CUR-RENT FAILURE	DUAL DEN-SITY	DEN-SITY NOT 1600	TAPE UNIT MODEL DEFINED			
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

BYTE 7

BIT / DEVICE	0	1	2	3	4	5	6	7
2305	ENCODED ERROR MESSAGE							
3203	PRINT CHK	CAB 2 PAR	DUCT PAR	INTRR OVRN	SPUR INTRR	MBSB	FMR PAR	TIME OUT
3330/3333 Mod I	FORMAT OF REMAINING SENSE BYTES (8 - 23)				ENCODED ERROR MESSAGE			
3330/3333 Mod II	FORMAT OF REMAINING SENSE BYTES (8 - 23)				ENCODED ERROR MESSAGE			
3340/3344	FORMAT OF REMAINING SENSE BYTES (8 - 23)				ENCODED ERROR MESSAGE			
3350	FORMAT OF REMAINING SENSE BYTES (8 - 23)				ENCODED ERROR MESSAGE			
3410/3411	LAMP FAILURE CHK	TAPE BOTTOM LEFT COLUMN CHK	TAPE BOTTOM RIGHT COLUMN CHK	RESET KEY	DATA SECUR-ITY ERASE CHK			
3420/3803	LAMP FAILURE	TAPE BOTTOM LEFT	TAPE BOTTOM RIGHT	RESET KEY	DATA SECUR-ITY ERASE	ERASE HEAD	AIR BEARING PRES-SURE	LOAD FAILURE
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

BYTE 8

BIT / DEVICE	0	1	2	3	4	5	6	7
3203	FAIL TO MOVE	ACCEL TOO SLOW	MISSING FDBK PULSE	EXTRA FDBK PULSE	STEP 1 CHK	STEP 2 CHK	CARR OP INC	CARR CNTRLD STOP
3410/3411		WRT FEED-THROU-GH CHK		END VELO-CITY CHK	NO READ-BACK DATA	START VELO-CITY CHK		MARG-INAL VELO-CITY
3420/3803	IBG DET IN WRT	FEED THRU		EARLY BEGIN RD BK CHK	EARLY END RD BK CHK SAGC	SLOW BEGIN RD BK CHK	SLOW END RD BK CHK	VELO-CITY RETRY/ RESTART
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

BYTE 9

BIT / DEVICE	0	1	2	3	4	5	6	7
3203	FORMS CHK	OUT OF FORMS	STACKER FULL/JAM	FRONT INTRLK OPEN	ATTCHMT COIL PROT	PRNTR INDPND COIL PROT	TRAIN VEL	SYNC CHK 0
3410/3411								
3420/3803	6250 COR	VELOC-ITY CHGE WHILE WRTNG	CH BUF CHK	CRC III	3803-2			CU RESVD
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

BYTE 10

BIT / DEVICE	0	1	2	3	4	5	6	7
3203	FUNC GO REJ	MISSING NPL DEV RDY	BLK DATA CHK	FOLDING			SYNC CHK 1	SYNC CHK 2
3420/3803	CMD STATUS REJ		CNTRL STATUS REJ	NO BLK ON RCD RD BK CHK	WTM NOT DETECT	TACHO-METER START FAIL		VELO-CITY
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

BYTE 11

BIT / DEVICE	0	1	2	3	4	5	6	7
3203	ATTCH RDY	CARR OP	PRINT OP	INTRR EXP	TIME OUT EXP	INIT HAM SETLNG	TRAIN DRIVE OFF	FEED ROLL IDLE
3420/3803	BUS PARITY LSR ADDR ERR MP1		LOW ROS PAR/PAR ERR ON BR INSTR MP1	INSTR DECODE MP1	MICRO PRGM DETECT-ED ERR MP1	D BUS PARITY MP1		BOC MP1
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

BYTE 12

BIT / DEVICE	0	1	2	3	4	5	6	7
3203	BUSY	ERROR PENDING	GO NOT RDY	ATTN PNDNG	PLB-ERLOG IN PRGRS	CARR CNTRLD STOP	FUNC REQ STKD	HI-SPD CARR
3420/3803	BUS PARITY LSR ADDR ERR MP2		LOW ROS PAR/PAR ERR ON BR INSTR MP2	INSTR DECODE MP2	MICRO PRGM DETECT-ED ERR MP2	D BUS PARITY MP2		BOC MP2
3800	MODULO 256 TRANSFER 2 PPI COUNT							

BYTE 13

BIT / DEVICE	0	1	2	3	4	5	6	7
3203	FUNCTION REQUEST BYTE							
3420/3803	CONTROL UNIT FEATURES		CONTROL UNIT UNIQUE ID HIGH					
3800	MODULO 256 FUSER 8-16 PPI COUNT							

BYTE 14

BIT / DEVICE	0	1	2	3	4	5	6	7
3203	PRINT COMPLT INTRR	NOT RDY INTRR	TIMED STATUS INTRR	ADPTR ERR INTRR	OP INTRR	MACH CHK INTRR	ENABL INTRR	INTRR RQST
3420/3803	CONTROL UNIQUE ID LOW							
3800	FUSER PAGE COUNT							

BYTE 15

BIT / DEVICE	0	1	2	3	4	5	6	7
3203	FIRST FAILING HAMMER POSITION							
3420/3803	TAPE UNIT UNIQUE ID							
3800	FUSER PAGE COUNT							

BYTE 16

BIT / DEVICE	0	1	2	3	4	5	6	7
3203	ENABLE	COMPARE	FIRED	HANDLED	DATA CHK	HAM FIRE	HAM MISFIRE	PLB PAR CHK
3420/3803	TAPE UNIT UNIQUE ID							
3800	PAPER COUNT							

BYTE 17

BIT / DEVICE	0	1	2	3	4	5	6	7
3203	NUMBER OF FAILING HAMMERS							
3420/3803	TWO CHAN-NEL SW (MIS)	CONTROL UNIT DEVICE SWITCH FEATURES			EC LEVEL OF CONTROL UNIT			
3800	PAPER COUNT							

BYTE 18

BIT / DEVICE	0	1	2	3	4	5	6	7
3420/3803	POWER CHK/ OVER-TEMPER-ATURE				EC LEVEL OF TAPE UNIT			
3800	SERIAL NUMBER							

BYTE 19

BIT / DEVICE	0	1	2	3	4	5	6	7
3420/3803	PRIMED FOR DEVICE END							
	TU 7	TU 6	TU 5	TU 4	TU 3	TU 2	TU 1	TU 0
3800	SERIAL NUMBER							

BYTE 20

BIT / DEVICE	0	1	2	3	4	5	6	7
3420/3803	PRIMED FOR DEVICE END							
	TU F	TU E	TU D	TU C	TU B	TU A	TU 9	TU 8
3800	PAGE BACK-UP COUNT							

BYTE 21

BIT / DEVICE	0	1	2	3	4	5	6	7
3420/3803	LOAD BUTTON DEPRESS	LEFT REEL TURN-ING	RIGHT REEL TURN-ING	TAPE PRE-SENT	REELS LOADED	LOAD REWIND	LOAD COM-PLETE	LOAD CHECK
3800	PAGE BACK-UP COUNT							

BYTE 22

BIT / DEVICE	0	1	2	3	4	5	6	7
3203	MESSAGE FORMAT 00 = PRINTER SUBSYSTEM 01 = CONTROLLER 02 = STATISTICS							
3420/3803	FRU IDENTIFIERS FOR CONTROL UNIT							
3800	PHOTO CONDUCTOR GAP LOCATION OR LOAD CHECK OFFSET							

BYTE 23

BIT / DEVICE	0	1	2	3	4	5	6	7
3420/3803	FRU IDENTIFIERS FOR CONTROL UNIT							
3800	PHOTO CONDUCTOR GAP LOCATION OR LOAD CHECK OFFSET							

3800 Sense Bytes 0-2

BYTE 0									
HEX →			80	40	20	10	08	04	02	01
BIT →			0	1	2	3	4	5	6	7
BIT HEX ↓	↓		COM-MAND REJECT	INTER-VEN-TION RE-QUIRED	BUS OUT PARITY	EQUIP-MENT CHECK	DATA CHECK	(RE-SERVED) 0	LOAD CHECK	CHAN-NEL 9
BYTE 1	0	80	INVALID COM-MAND	NOT READY	COM-MAND CODE	HARD-WARE ERROR	UN-PRINT-ABLE CHAR-ACTER		IN-COR-RECT LENGTH	(RE-SERVED)
	1	40	(RE-SERVED)	OPERA-TION CHECK	DATA BYTE	PER-MANENT ERROR	(RE-SERVED)		IN-CORRECT MUL-TIPLE OF 6 OR 8 LINES	
	2	20		TONER COLL-ECTOR FULL	(RE-SERVED)	ERROR LOG FULL	NO TRANS-LATE TABLE		FCB 1/2 INCH ERROR	
	3	10		TONER EMPTY		CANCEL KEY	NO FCB CHAN-NEL CODE MATCH		INVALID FCB CHAN-NEL CODES	
	4	08		DEVEL-OPER REPLACE-MENT RE-QUIRED		(RE-SERVED)	MUL-TIPLE CHAR-ACTERS		FCB VS FORM LENGTH ERROR	
	5	04		END OF FORMS			(RE-SERVED)		WCGM NOT LOADED	
	6	02		OUTPUT FULL					UN-ASSI-GNED GRAPHIC CHAR-ACTER	
	7	01		(RE-SERVED)					(RE-SERVED)	
BYTE 2	0	80	(RE-SERVED)	FORMS HOLDER INTER-LOCK	(RE-SERVED)	(RE-SERVED)	(RE-SERVED)		INVALID WCGM ID	(RE-SERVED)
	1	40		TRANS-FER CHECK					NO ID FOR WCGM 00	
	2	20		FUSER CHECK					INVALID COPY MODIFY SE-QUENCE	
	3	10		CPS CHECK					INVALID FORMS OVER-LAY SE-QUENCE	
	4	08		PROCESS CHECK					INVALID GRAPHIC MODIFY SE-QUENCE	
	5	04		BURSTER TRIM-MER STACK-ER CHECK					WCGM DATA PARITY ERROR	

3800 Sense Bytes 0-2 (continued)

BYTE 0									
HEX →			80	40	20	10	08	04	02	01
BIT →			0	1	2	3	4	5	6	7
BIT HEX ↓ ↓			COMMAND REJECT	INTERVENTION REQUIRED	BUS OUT PARITY	EQUIPMENT CHECK	DATA CHECK	(RESERVED) 0	LOAD CHECK	CHANNEL 9
BYTE 2	6	02	(RESERVED)	(RESERVED)	(RESERVED)	(RESERVED)	(RESERVED)		(RESERVED)	(RESERVED)
	7	01		LINE OVERRUN					(RESERVED)	

3838 Sense Information FIXED Format

BIT / BYTE	0	1	2	3	4	5	6	7
0	COM-MAND REJECT	INT REG	BUS OUT PARITY	EQUIP CHECK	DATA CHECK	OVERRUN	NOT INITIAL-IZED	PREVIOUS I/O EXCEPT-TION
1	RESTART-ABLE	UNRE-LATED ERROR	DATA-DEPEN-DENT ERROR	INVALID PASS-WORD	UNDE-FINED	UNDE-FINED	UNDE-FINED	UNDE-FINED
2	CP (MACHINE CHECK)	I/O	AE	DTC	CP EXECU-TIVE	PERFOR-MANCE MONI-TOR		
3	• FORMAT OF BYTES 8-31 • FOR EACH VALUE OF THIS FIELD, LOOK UP CORRESPONDING SENSE FORMAT. VALUE = 0/FORMAT 0, VALUE = 1/FORMAT 1, VALUE = 2/FORMAT 2, VALUE = 4/FORMAT 4.							
4	BYTES 4-5 ARE THE CIT OP CODE.							
5								
6	BYTES 6 7 IS THE CIT OFFSET.							
7								

3838 Sense Information Extended Format 0 (Equipment – Related)

BIT / BYTE	0	1	2	3	4	5	6	7
8	BULK STORAGE	CP PROGRAM STORAGE	CP CONTROL STORAGE	AE WORKING STORAGE	AE CONTROL STORAGE			
9	BULK STORAGE	CP PROGRAM STORAGE						
10	ILLEGAL MICRO-INSTRUC-TION	EDB TIMEOUT	EDB PARITY ERROR (READ)	EDB PARITY ERROR (OUT-BOUND)				
11	MACHINE CHECK	WS CONFLICT						
12	INVALID INSTRUC-TION	DIVIDE BY ZERO	I/O INITIATE FAIL	INVALID IOCW	DTC INITIATE FAIL	INVALID TCB	AE INITIATE FAIL	INVALID AFCW
13	INVALID ADDRESS REFER-ENCE	DEVICE INITIALI-ZATION ERROR	INSUFFI-CIENT BUFFER SPACE					
14	BYTES 14-15 STORAGE PAGE NUMBER							
15								
16	BYTES 16-17 CP PROGRAM STORAGE ADDRESS							
17								
18	BYTES 18-19 CONTROL STORAGE ADDRESS REGISTER							
19								
20-27	IOCW BYTES 20-27							
28-31	BYTES 28-31 UNUSED							

3838 Sense Information Extended Format 1 (Hardware Verification)

BIT / BYTE	0	1	2	3	4	5	6	7
8	BYTE 8 CONTAINS PERFORMANCE MONITOR EXECUTION COUNT							
9	BYTE 9 CONTAINS PERFORMANCE MONITOR ERROR COUNT							
10	PERFORMANCE MONITOR ERROR STATUS 1-10 (BYTES 10-19) EACH OF THESE FIELDS MAY CONTAIN AN ERROR INDICATION GENERATED BY THE PERFORMANCE MONITOR FUNCTIONAL VERIFICATION PACKAGE. THE NUMBER OF VALID FIELDS IS EQUAL TO THE ERROR COUNT (SENSE BYTE 9). THE FORMAT OF EACH FIELD IS AS FOLLOWS: BIT — CONDITION 0 — CP FAILURE 1 — AE FALIURE 2 — BULK STORAGE FAILURE 3 — DTC FAILURE 4-7 — STATUS CODE DEPENDING ON SETTING OF BITS 0-3							
11								
12								
13								
14								
15								
16								
17								
18								
19								
20-31	UNUSED							

3838 Sense Information Extended Format 2 (Data Related)

BIT / BYTE	0	1	2	3	4	5	6	7
8	I/O LIMIT CHECK	I/O LENGTH ERROR	CIT CHANNEL PROGRAM CORRE-SPON-DENCE	CIT LIMIT CHECK	INVALID CIT OPERA-TION	INVALID CIT PARA-METER	INVALID CIT BRANCH	CIT BRANCH COUNT EX-CEEDED
9	CIT INDEX OVER-FLOW	CIT INDEX DIVIDE CHECK	BULK STORAGE LIMIT CHECK	DTC CONVER-SION CHECK	SELECTED ERROR STOP			
10				RECIPRO-CAL ESTIMA-TOR EXPONENT OVER-FLOW	MULTIPLY EXPO-NENT OVER-FLOW	MULTIPLY EXPO-NENT UNDER-FLOW	INTEGER MULTIPLY OVER-FLOW	UNNOR-MALIZED RECIPRO-CAL ESTIMA-TOR INPUT
11	ADDER A EXPO-NENT OVER-FLOW	ADDER A EXPO-NENT UNDER-FLOW	ADDER A INTEGER OVER-FLOW	ADDER A RIGHT EXPO-NENT LARGER	ADDER B EXPO-NENT OVER-FLOW	ADDER B EXPO-NENT UNDER-FLOW	ADDER B INTEGER OVER-FLOW	ADDER B RIGHT EXPO-NENT LARGER
12-15	BYTES 12-15 CONTAIN THE CIT SUCCESSFUL BRANCH COUNT							
16-31	UNASSIGNED							

3838 Sense Information Extended Format 4 (Not Initialized)

BIT / BYTE	0	1	2	3	4	5	6	7
8	CONTAINS THE SUBCHANNEL I. D.							
9				SENSE ID	CHANNEL END	UNIT CHECK	STATUS MODIFIER	BUSY
10	READ	WRITE	SENSE	CONTROL PROCESS	CONTROL SYSTEM	DISCON-NECT IN	SELEC-TIVE RESET	INTER-FACE DISCON-NECT
11	COM-MAND REJECT	BUS OUT PARITY	SUPPRESS OUT	STOP	DATA CHECK	ADDRESS CHECK	STATUS TAKEN	ONLINE
12		ILLEGAL MICRO-BRANCH	EQUIP-MENT CHECK	DIVIDE BY ZERO	INVALID ADDRESS	INVALID INSTRUC-TION		
13	CP MICRO STATE -- VALID ONLY WHEN BYTE 0, BIT 3 IS ON							
14						EXTER-NAL BUS PARITY ERROR (READ)	EXTER-NAL BUS PARITY ERROR (WRITE)	CONTROL STORE PARITY ERROR
15	PROGRAM STORE PARITY ERROR (INST. FETCH)	PROGRAM STORE PARITY ERROR (DATA FETCH)		ILLEGAL ADDRESS (INSTRUC-TION FETCH)	ILLEGAL ADDRESS (DATA FETCH)	EXTER-NAL BUS TIMEOUT		
16-17	CP PROGRAM STORAGE DATA ADDRESS							
18-19	CP CONTROL STORAGE ADDRESS REGISTER							
20-21	UNASSIGNED							
22-23	CP INSTRUCTION COUNTER							
24-31	UNASSIGNED							

GENERAL
REFERENCE

This section provides general reference information useful for debugging purposes. There are eight subsections:

- **SVC Summary**

  defines the five types of SVC routines, briefly describes the SVC table, summarizes system-defined SVC instructions, and lists the SVC's and associated macros.
- **SYSEVENT Summary**

  describes system events causing entry to the system resources manager (SRM) by a direct branch or SVC 95.
- **Module Summary**

  identifies, for each three-character module-name prefix, the corresponding system component and program logic manual. (Where the module-name prefix is also a system message number, the appropriate message manual is also identified.)
- **Component Summary**

  identifies the component microfiche, the modules that belong to the component, and the program logic manual that provides primary documentation.
- **Storage Summary**

  briefly describes the layout of real and virtual storage, the use of storage protection keys, and the meaning of virtual storage subpools.
- **Serialization Summary**

  describes the use of locks and ENQ/DEQ names.
- **General System Flow**

  describes generalized control flow among system components.
- **Acronyms**

  defines commonly used acronyms. For each acronym that identifies a data area, the definition includes a reference to the mapping macro and to the publications that describe the data area.

SVC Summary

This summary defines the five types of SVC routines, briefly describes the SVC table, and summarizes each system-defined SVC instruction.

SVC Routines

There are five types of SVC routines, which are distinguished as follows:

Residence: Types 1, 2 and 6 SVC routines are part of the nucleus; types 3 and 4 SVC routines reside in the link pack area. A type 3 routine is a single load module, while a type 4 routine consists of two or more load modules.

Naming Conventions: Types 1, 2 and 6 SVC routines are named IGCxxx, where xxx is the SVC number (decimal). Types 3 and 4 SVC routines are named IGCyyxxx, where xxx is the SVC number and yy is the number of the load, beginning with 00 for the first (or only) load.

For types 3 and 4 SVC routines, the internal format of the SVC number (xxx) is a zoned decimal with a four-bit sign code (1100) in the low-order position. Because of the sign code, a low-order digit between 1 and 9 corresponds to an EBCDIC character between A and I; a low order zero corresponds to an EBCDIC opening brace '{'. Since an opening brace is not included in most character sets, a low-order zero is usually entered as a 12-0 punch and printed as a blank.

Register Conventions: SVC routines are entered with the following data in the general purpose registers:

Registers 0, 1, 13, and 15 – Contents when the SVC instruction was executed.
Register 3 – Address of the CVT.
Register 4 – Address of the TCB.
Register 5 – Address of the current RB (for type 1 SVC), or address of the SVRB for SVC routine (for type 2, 3, 4, or 6 SVC).
Register 6 – Address of the SVC routine entry point.
Register 7 – Address of the ASCB.
Register 14 – Return address.
Other registers – Unpredictable

Locks: Each SVC routine is entered with the locks specified for the routine in the SVC table. In addition, each type 1 SVC routine is entered with the LOCAL lock, whether or not it is specified in the SVC table; this lock must not be released. An SVC routine can acquire any lock, and runs enabled or disabled depending on the lock held. To avoid disabled page faults, a type 3 or 4 SVC routine must fix its pages in real storage before acquiring a disabled lock (any lock other than LOCAL, CMS, or CMSEQDQ). A type 6 SVC may not be suspended for a lock request (applicable to MVS/System Extensions).

Page Faults: An SVC routine can be restarted after a page fault, provided that the routine does not hold a disabled lock.

SVC Instructions: An SVC routine can issue SVC instructions, provided that it does not hold any lock. (***Note:*** A type 1 SVC routine cannot issue SVC instructions, because it always holds at least the LOCAL lock.)

Other Characteristics: All SVC routines are entered in supervisor state with a zero storage protect key (other keys may be used during execution). Use of an SVC routine is authorized or unauthorized, as specified in the SVC table. A type 6 SVC executes disabled and must not enable (applicable to MVS/System Extensions).

SVC Table

The SVC table is a system data area that contains one eight-byte entry for each system-defined or user-defined SVC instruction. Each entry contains the following information.

- The entry-point address of the SVC routine.
- The SVC type (1, 2, 3, 4, or 6).
- The function code (authorized or unauthorized).
- The locks to be acquired by the SVC first level interruption handler before the SVC routine is executed.

The format of an SVC table entry is described under "SVCTABLE" in the "Data Area Descriptions" portion of Volume 3.

System SVC Instructions

For each SVC, this summary provides the following information:

- The SVC instruction in assembler language and machine language (hexadecimal).
  Example: SVC 16 (0A10)
- The macro instructions that generate the SVC instruction.
- The SVC type (1, 2, 3, 4, or 6).
- Locks acquired by the SVC routine or by the SVC first level interruption handler.
- The object module containing the entry point of the SVC routine.
- The applicable PLM which describes the SVC's operation.
- APF protected, if applicable. (Unless otherwise noted, the SVC in question is *not* APF protected.)
- GTF trace data:
  - Information passed to the SVC routine in general registers 15, 0, and This includes the extended SVC routing codes for SVC 109, SVC 116, and SVC 122.
  - Additional information displayed in GTF comprehensive trace records (but omitted in GTF minimal trace records). For complete GTF record formats, refer to "SVC Trace Records" in Section 3.

SVC - Macro List

The following is a list of SVCs and associated macros.

SVC	Macro	SVC	Macro
0	EXCP/XDAP	70	GSERV
1	WAIT/WAITR	71	ASGNBFR/BUFINQ/
2	POST/PRTOV		RLSEBFR
3	EXIT	72	'NO MACRO'
4	GETMAIN	73	SPAR
5	FREEMAIN	74	DAR
6	LINK	75	DQUEUE
7	XCTL	76	IFBSTAT
8	LOAD	77	'RESERVED'
9	DELETE	78	LSPACE
10	GETMAIN/FREEMAIN	79	STATUS
	(with R operand)	80	'RESERVED'
11	TIME	81	SETPRT
12	SYNCH	82	DASDR
13	ABEND	83	SMFWTM
14	SPIE	84	GRAPHICS
15	ERREXCP	85	DDRSWAP
16	PURGE	86	ATLAS
17	RESTORE	87	DOM
18	BLDL/FIND (TYPE D)	88	MOD88
19	OPEN	89	'RESERVED'
20	CLOSE	90	'RESERVED'
21	STOW	91	VOLSTAT
22	OPEN (TYPE = J)	92	TCPEXCP
23	CLOSE (TYPE = T)	93	TGET/TPUT
24	DEVTYPE	94	'SEE SVC 94'
25	TRKBAL	95	SYSEVENT
26	CATALOG/INDEX/	96	STAX
	LOCATE	97	IKJEGS9G
27	OBTAIN	98	PROTECT
28	'RESERVED'	99	DYNALLOC
29	SCRATCH	100	IKJEFFIB
30	RENAME	101	QTIP
31	FEOV	102	AQCTL
32	ALLOC	103	XLATE
33	IOHALT	104	TOPCTL
34	MGCR/QEDIT	105	IMGLIB
35	WTO/WTOR	106	'RESERVED'
36	WTL	107	MODESET
37	SEGLD/SEGWT	108	'RESERVED'
38	'RESERVED'	109	'SEE SVC109'
39	LABEL	110	'NO MACRO'
40	EXTRACT	111	'NO MACRO'
41	IDENTIFY	112	PGRLSE
42	ATTACH	113	PGFIX/PGFREE/
43	CIRB		PGLOAD/PGOUT
44	CHAP	114	EXCPVR
45	OVLYBRCH	115	'RESERVED'
46	TTIMER	116	'SEE SVC116'
47	STIMER	117	DEBCHK
48	DEQ	118	'RESERVED'
49	'RESERVED'	119	TESTAUTH
50	'RESERVED'	120	GETMAIN/FREEMAIN
51	SNAP/SDUMP	121	VSAM
52	RESTART	122	'SEE SVC122'
53	RELEX	123	PURGEDQ
54	DISABLE	124	TPIO
55	EOV	125	EVENTS
56	ENQ/RESERVE	126	MSS(ICB2SVC)
57	FREEDBUF	127	'RESERVED'
58	RELBUF/REQBUF	128	'RESERVED'
59	OLTEP	129	'RESERVED'
60	STAE/STAI-ESTAE/ESTAI	130	RACHECK
61	IKJEGS6A	131	RACINIT
62	DETACH	132	RACLIST
63	CHKPT	133	RACDEF
64	RDJFCB	134	'RESERVED'
65	'RESERVED'	135	'RESERVED'
66	BTAMTEST	136	'RESERVED'
67	'RESERVED'	137	'RESERVED'
68	SYNADAF/SYNADRLS	138	'RESERVED'
69	BSP		

SVC 0 (0A00): EXCP/XDAP macro - is type 1, gets LOCAL lock, calls module IECVEXCP (E.P. = IGC000). PLM is *OS/VS2 1/O Supervisor Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the IOB associated with this request.

DDNAME cccccccc Name of the associated DD statement.

DCB xxxxxxxx Address of the DCB associated with this I/O request.

DEB xxxxxxxx Address of the DEB associated with this I/O request.

SVC 1 (0A01) WAIT/WAITR macro - is type 1, gets LOCAL lock, calls module IEAVSY50 (E.P. = IGC001). PLM is *OS/VS2 System Logic Library* GTF data is:

R15 No applicable data.

R0 Count of the number of events being waited for. If the count is zero, the wait is treated as a NOP. Bit 0 equals one indicates a long wait.

R1 If positive, the address of the ECB being used. If complemented, the address of a list of ECB addresses.

PLIST - Four to 40 bytes of the WAIT/WAITR parameter list, which has a maximum length of 1020 bytes. The list is a series of fullwords, each containing the address of an ECB.

SVC 2 (0A02) - POST/PRTOV macro - is type 1, gets LOCAL and SALLOC locks, calls module IEAVSY50 (E.P. = IGC002) for POST, module IGG019CL for PRTOV. PLM is: *OS/VS2 System Logic Library* for POST, *OS/VS2 SAM Logic* for PRTOV. GTF data is:

R15 No applicable data.

R0 For POST: The completion code to be placed in the ECB.

R1 For POST: The address of the ECB to be posted or (if the high-order bit is 1), the address of a parameter list as follows:

Bytes:

0-3 Address of the ECB.

4-7 Address of the ASCB for the address space that contains the ECB

8-11 Address of the ERRET routine.

12 Contains the storage protection key of the ECB if the high-order list of R0 is on and the high-order bit of R1 is on.

SVC 3 (0A03) - EXIT macro - is type 1, gets LOCAL lock, calls module IEAVEOR (E.P. = IGC003). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15, R0, and R1 - No applicable data.

SVC 4 (0A04) GETMAIN macro - is type 1, gets LOCAL lock, calls module IEAVGM00 (E.P. = IGC004). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list passed when the SVC was called.

PLIST - 10 bytes in length:

Bytes

0-3
- a. Single area request - length requested.
- b. variable request - address of a doubleword containing the minimum maximum length requested. Format is:

  Bytes

  0 Zero.

  1-3 Minimum length.

  4 Zero.

  5-7 Maximum length.
- c. List request - address of a list of lengths requested (one word per request); last word contains X'80' in byte 0.

4 Zero

5-7
- a. Single area request - address of a word GETMAIN initializes with the address of the area acquired.
- b. Variable area request - address of a doubleword GETMAIN initializes with the address of the area acquired and the actual length allocated.
- c. List request - address of a list of areas that GETMAIN initializes with the addresses of the areas allocated for each requested length in the length list.

8 Flag byte, format is:

- 10 Request is for storage aligned on a page boundary.
- 00 Unconditional single area request.
- 20 Conditional single area request.
- 80 Unconditional list request.
- A0 Conditional list request.
- C0 Unconditional variable request.
- E0 Conditional variable request.

9 Subpool identification.

SVC 5 (0A05) FREEMAIN macro - is type 1, gets LOCAL lock, calls module IEAVGM00 (E.P. = IGC005). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the input parameter list.

PLIST - 10 bytes, contents are:

Bytes

0-3
- a. Single area request - length to be freed.
- b. List area request - address of a list of FREEMAIN length requests (1 word per request); last word contains X'80' in byte 0.
- c. Variable Request-zero

4-7
- a. Single area request - address of a word containing the address of the area to be freed.
- b. List area request - address of a list of addresses of areas to be freed.
- c. Variable request-address of a doubleword containing the address to be freed in the first word and the length to be freed in the second word.

8 Flag byte, format is:

- 00 Unconditional single area request.
- 20 Conditional single area request.
- 80 Unconditional list area request.
- A0 Conditional list area request.
- C0 Unconditional variable request.
- E0 Conditional variable request.

9 Subpool identification.

SVC 6 (0A06 LINK macro - is type 2, gets LOCAL and CMS locks, calls module IEAVLK00 (E.P. = IGC006). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 Address of the parameter list.

R0 No applicable data.

R1 Address of the user optional parameter list.

NAME cccccccc entry point/directory entry (EP/DE) name of the module to be linked to or given control.

PLIST - The parameter list is twelve bytes long; the format is:

Bytes	
0	Flag byte
	80 DE form of macro instruction.
	00 EP and EPLOC form of macro instruction.
1-3	Address of the directory entry list, if byte 0 is X'80'. Address of the entry point name, if byte 0 is X'00'.
4	Indicates an extended parameter list. If X'80'.
5	DCB address or zero.
8	Contains the address of routine to get control on error (ERRET parameter) if byte 4 is X'80'.

SVC 7 (0A07) XCTL macro - is type 2, gets LOCAL and CMS locks, calls module IEAVLK00 (E.P. = IGC007). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 Address of the parameter list.

R0 and R1 - No applicable data.

NAME cccccccc entry point/directory entry (EP/DE) name of the module to be linked to or given control.

PLIST - The parameter list is eight bytes long; the format is:

Bytes	
0	Flag byte
	80 DE form of macro instruction
	00 EP and EPLOC form of macro instruction
1-3	Address of the directory entry list if byte 0 is X'80'. Address of entry point name if byte 0 is X'00'.
4	No applicable data.
5	DCB address or zero.

SVC 8 (0A08) LOAD macro - is type 2, gets LOCAL and CMS locks, calls module IEAVLK00 (E.P. = IGC008). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Content:

If byte 0 contains X'00', bytes 1 - 3 contain the address of the entry point name.

If byte 0 contains X'80', bytes 1 - 3 contain the address of the directory entry list.

R1 Byte 0 - If X'80' specifies that a return is requested.

Bytes 1 - 3 is the DCB address, or zero if the default for DCB was specified.

NAME cccccccc entry point/directory entry name of the module to be loaded.

SVC 9 (0A09) DELETE macro - is type 2, gets LOCAL and CMS locks, calls module IEAVLK00 (E.P. = IGC009). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R1 - No applicable data.

R0 Address of the entry point name.

NAME cccccccc entry point name of the module to be deleted.

SVC 10 (0A0A) GETMAIN/FREEMAIN macro with R operand - is type 1, gets LOCAL lock, calls module IEAVGM00 (E.P. = IGC010). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Number of the subpool requested in the high-order byte, and the length of the area requested in bytes 1-3. (A zero length is required for a subpool FREEMAIN).

R1 Any negative value if the request is for a GETMAIN.
Address of the storage to be freed if the request is for a FREEMAIN.
Zero if the request is for a FREEMAIN of an entire subpool.

SVC 11 (0A0B) TIME macro - is type 3, gets no lock, calls module IEAVRT01 (E.P. = IGC0001A). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Address of the area into which the microsecond elapsed time or the current TOD clock value is to be placed.

R1 Low-order byte has flag bits that designate that the time will be returned in register 0, and in what format.

Bits	Register 0 Contents
.... 0000	32-bit unsigned binary number representing the number of elapsed timer units. (A timer unit is approximately 26.04 microseconds)
.... 0001	Elapsed time in hundredths of a second.
.... 0010	Packed decimal digits representing elapsed time in hours, minutes, seconds, tenths of a second, and hundredths of a second (HHMMSShh).
.... 0011	Elapsed time where bit 51 of doubleword is equivalent to one microsecond.
.... 0100	The current TOD clock value is to be returned.
.1..	The routine specified by the ERRET operand gets control on an environmental error
1...	GMT values are to be returned.

SVC 12 (0A0C) SYNCH macro - is type 2, gets LOCAL and CMS locks, calls module IEAVLK00 (E.P. = IGC012). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 Address of the entry point for the processing program that is to be given control.

R0 and R1 - Optional user parameters.

SVC 13 (0A0D) ABEND macro - is type 4, gets LOCAL lock, calls module IEAVTRT2 (E.P. = IGC013). PLM is *OS/VS2 System Logic LIbrary*. GTF data is:

R15 No applicable data.

R0 If the DUMPOPT parameter is specified, R0 contains the address of a parameter list valid for the SNAP macro.

R1 Applicable if SVC 13 was not called by the ABTERM routines; format is:

Bytes			
0	Flag byte		
	Bits		
	1...	....	DUMP option.
	.1..	....	STEP option.
	..1.	....	DUMPOPT specified.
	...x	xxxx	Reserved.
1-3	ABEND		Completion code.

CMP CODE - The ABEND completion code if SVC 13 was called by ABTERM routines. It is the content of the TCBRCMP field of the current TCB at the time the SVC interruption occurred. If ABEND recursion has occurred, this field contains the recursive completion code.

SVC 14 (0A0E) SPIE macro - is type 3, gets LOCAL lock, calls module IEAVTB00 (E.P. = IGC0001D). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the PICA.

PICA xxxxxxxx .... PICA from the associated SPIE macro instruction.

SVC 15 (0A0F) ERREXCP macro - is type 1, gets LOCAL, IOSUCB, and IOSCAT locks, calls module IECVPST (E.P. = IGC015). PLM is *OS/VS2 I/O Supervisor Logic*. APF protected via TESTAUTH. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the IOSB that was assigned to this I/O request by IOS.

DDNAME cccccccc
U/A
****

cccccccc Name of the DD statement associated with this I/O request.

U/A Indicates that the DDNAME was unavailable because the DEB address was verified as invalid.

******** A program check occurred trying to gather the DDNAME.

ERP flags

IOSFLA flags from IOSB assigned to this request by IOS;

Format is:

IOSERR	..1.	....	Error routine is in control of this SRB. If the ERP returns with this bit on a retry is assumed. If the ERP returns with this bit off, the error is considered to be permanent or corrected depending on the setting of bit IOSEX.
IOSEX	....	.1..	Exceptional condition is set by IOS. Upon return from the abnormal or normal exit with this bit on, ERP processing is initiated if this is an initial error condition. If this bit is off, it is assumed that the exit corrected the condition or did not consider it an error. When the error routine returns with this bit on and bit IOSERR is off, the error is considered permanent. When the ERP returns with both bits off, the error has been corrected.
	xx.x	x.xx	No meaning for ERREXCP.

TCB Address of the TCB associated with the SRB scheduled to handle this I/O request.

DCUU - Device address, in channel-unit form, of the device associated with this I/O request.

SVC 16 (0A10) PURGE macro - is type 3, gets LOCAL, IOSUCB, IOSLCH, and IOSYNCH locks, calls module IGC0001F (E.P. = IGC016). PLM is *OS/VS2 I/O Supervisor Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 If positive, contains the address of the purge parameter list. If negative (complemented), contains the address of the IPIB.

DDNAME cccccccc
U/A
****

cccccccc Name of the DD statement associated with the requests being purged.
U/A Indicates that the DDNAME was unavailable because the DEB address was unavailable.
******** Indicates that a program check interruption occurred while trying to gather the DCB address or DDNAME.

DCB xxxx
U/A
****

xxxx Address of the DCB associated with the purge request.
U/A Unavailable because PPLDSID was 0 or verified as an invalid DEB address.
**** Indicates that a program check interruption occurred while trying to gather the DCB address.

PLIST - Purge parameter list; format is:

Bytes		Bits	
0	PPLOPT1 Option byte 1; bit settings are:		
	PPLDS	1...	If DSID purge was requested (bit 6), purge a single DSID (see PPLDSID). If zero, purge the DSID list.
	PPLPOST	.1..	ECBs associated with I/O requests purged should be posted with X'48'.
	PPLHIO	..1.	Halt the I/O requests and do not build a PIRL.
	PPLREL	...1	Purge only the I/O requests marked related and associated with the argument.
		.... x...	Reserved and must be zero.
	PPLRB	.... .1..	Do not purge the RB chain for asynchronously scheduled routines.
	PPLTASK	.... ..1.	If ASID purge is not specified, purge a single TCB.
	PPLEXR	1	Option byte 2 is present and contains valid information.
1-3	PPLDSID		Address of the DEB, the argument used for DSID purge.
4	PPLCC		Completion code.

If bit 8 of option byte 1 is 0, the only completion code is X'7F'.

If bit 8 of option byte 1 is 1, the completion codes are as follows:

X'7F'-Successful completion of the purge request.

X'40'-Unsuccessful completion. Details in register 15.

5-7	PPLTCBA		Address of the TCB.
8	PPLDVRID		Driver ID for the DSID purge – X'00' is for EXCP.
9-11	PPLPIRL		This is the address of the anchor from which the purged I/O Request List (PIRL) will be chained. The anchor is a single word where the right 3 bytes are used for a pointer to the PIRL. If the address in the anchor is X'FFFFFF', there was no I/O request purged.
12	PPLOPT2		Option byte 2, present if PPLSRM1, bit 8 is 1.
	PPLASID	..1.	Address space purge is specified.
		..0.	Address space purge is not specified.
	PPLCV	...1	Perform the DSID validity check.
	PPLOTCB	.... 1...	Purge the I/O requests so that when they are restored they will be associated with the TCB that originated them.
		.... 0...	Purge the I/O requests so that when they are restored they will be associated with the restoring TCB.
		xx.x .xxx	Reserved and must be zero.
	PPLBSS	... ..1.	Purge was called by the RCT – bypass the status start SRB's.

SVC 17 (0A11) RESTORE macro - is type 3, gets no lock, calls module IGC0001G (E.P. = IGC017). PLM is *OS/VS2 I/O Supervisor Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the pointer to the PIRL created by PURGE or a pointer to the fullword of X'xxFFFFFF', which means there are no requests to RESTORE.

PLIST as follows:

Bytes:

1	PIROPT	Option byte, bits meaning:	
	PIROTCB	1...	Restore the I/O requests to the TCB'(s) that originally started them. If they were not purged with that possibility, restore them to the restoring TCB.
		0...	Restore the I/O requests to the restoring TCB.
	PIRSUPCK	.1..	Perform the RESTORE TCB validity check even though the caller may be in supervisor state.
		.0..	Perform the TCB validity check based on the state of the caller.
		..xx xxxx	Reserved and must be zero.
2	PIRCNT	Number of PIRRSTR entries in the PIRL.	
3-4	Reserved and should be zero.		
5-8	PIRRSTR	The pointer to the I/O request list in the form required by the appropriate driver.	
9-C	PIRDVRU	The pointer to additional data the driver maintains.	

Note: PIRRSTR and PIRDVRU are repeated the number of times specified in PIRCNT.

SVC 18 (0A12) BLDL/FIND (Type D) macro - is type 2, gets no lock, calls module IGC018. *OS/VS2 SAM Logic*. GTF data is:

R15 No applicable data.

R0 Address of the parameter list.

R1 DCB address. If the address is positive, this is a BLDL request. If negative, this is a FIND request. If zero, this is a BLDL request on TASKLIB, STEPLIB, or JOBLIB concatenated with SYS1.LINKLIB.

PLIST - 12 bytes of the parameter list are traced.

(The parameter list may be longer than 12 bytes.)

Bytes

0-3	BLDL	FF is the number of entries. LL is the length of each entry.
4-11	BLDL	Hexadecimal representation of the first member name for which the BLDL was issued.
0-8	FIND	hexadecimal representation of the member name.

SVC 19 (0A13) OPEN macro - is type 4, gets LOCAL lock, calls module IGC0001I. PLM is *OS/VS2 OPEN/CLOSE/EOV Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of parameter list.

PLIST - Four to 40 bytes of OPEN parameter list, which has a maximum length of 1020 bytes. The list is a series of 4-byte entries in the following format:

Bytes

0 Option byte; bit settings are:

Bits		
1...	....	Last entry indicator.
.000	....	DISP.
.011	....	LEAVE.
.001	....	REREAD.
....	0000	INPUT.
....	1111	OUTPUT.
....	0100	UPDAT.
....	0111	OUTIN.
....	0011	INOUT.
....	0001	RDBACK.
....	1110	EXTEND.
....	0110	OUTINX.

1-3 DCB address.

SVC 20 (0A14) CLOSE macro - is type 4, gets LOCAL lock, calls module IGC00020. PLM is *OS/VS2 OPEN/CLOSE/EOV Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list.

PLIST - Four to 40 bytes of the OPEN parameter list, which has a maximum length of 1020 bytes. The list is a series of 4-byte entries in the following format:

Bytes

0 Option byte; bit settings are:

Bits

1...	....	Last entry indicator.
.000	....	DISP.
.100	....	REWIND.
.010	....	FREE.
.011	....	LEAVE.
.001	....	REREAD.

1-3 DCB address.

SVC 21 (0A15) STOW macro - is type 3, gets no lock, calls module IGC0002A. PLM is *OS/VS2 SAM Logic*. GTF data is:

R15 No applicable data.

R0 Address of the parameter list.

R1 Address of the associated DCB.

The sign of R0 and R1 indicate the directory action STOW is to take:

R0	R1	Action.
+	+	ADD.
+	-	REPLACE.
-	+	DELETE.
-	-	CHANGE.

DDNAME cccccccc name of the associated DD statement.

PLIST - The parameter list is of variable length, depending on the directory action being performed:

For ADD or REPLACE — 12 bytes of the parameter list will be dumped. The first 8 bytes contain the member name; the next 3 bytes contain the member's TTR; and the next byte contains the alias bit, number of TTRNs in the user data area, and the length of the user data area in halfwords. (The user data area varies from 0-62 bytes in length and does not appear.)

For DELETE — 8 bytes long and contains the member name or alias of the PDS directory entry being acted upon.

For CHANGE — 16 bytes long; first 8 bytes contain the old member name or alias; second 8 bytes contain the new member name or alias.

SVC 22 (0A16) OPEN (TYPE=J) macro - is type 4, gets LOCAL lock, calls module IGC0002B. PLM is *OS/VS2 OPEN/CLOSE/EOV Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list.

PLIST - Four to 40 bytes of the OPEN parameter list, which has a maximum length of 1020 bytes. The list is a series of 4-byte entries in the following format:

Bytes

0 Option byte; bit settings are:

Bits

1...	....	Last entry indicator.
.000	....	DISP.
.011	....	LEAVE.
.001	....	REREAD.
....	0000	INPUT.
....	1111	OUTPUT.
....	0100	UPDAT.
....	0111	OUTIN.
....	0011	INOUT.
....	0001	RDBACK.
....	1110	EXTEND.
....	0110	OUTINX.

1-3 DCB address.

SVC 23 (0A17) CLOSE (TYPE=T) macro - is type 4, gets LOCAL lock, calls module IGC0002C. PLM is *OS/VS2 OPEN/CLOSE/EOV Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list.

PLIST - Four to 40 bytes of the CLOSE parameter list, which has a maximum length of 1020 bytes. The list is a series of 4-byte entries in the following format:

Bytes

0 Option byte; bit settings are:

Bits

1...	....	Last entry indicator.
.011	....	LEAVE.
.001	....	REREAD.

1-3 DCB address.

SVC 24 (0A18) DEVTYPE macro - is type 3, gets no lock, calls module IGC0002D. PLM is *OS/VS2 SAM Logic*. GTF data is:

R15 No applicable data.

R0 Address of the output area or the two's complement of the output area address.

R1 Address of the DDNAME or the two's complement of the address of the DDNAME.

DDNAME cccccccc DDNAME associated with this request

SVC 25 (0A19) TRKBAL macro - is type 3, gets no lock, calls module IGC0002E. PLM is *OS/VS2 SAM Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the associated DCB.

If R1 is negative, the address is in complement form and the DCBFDAD and DCBTRBAL fields of the DCB are meaningless.

DDNAME cccccccc name of the associated DD statement.

DCBFDAD xxxxxxxx full direct access address (MBBCCHHR) from the DCB that is pointed to by R1.

DCBTRBAL xxxx track balance — number of bytes remaining on the current track after a write; negative if no bytes remain.

SVC 26 (0A1A) CATALOG/INDEX/LOCATE macro - is type 4, gets no lock, calls module IGC0002F. PLM is *OS/VS2 CATALOG Management Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list when a SVC 26 is issued.

The parameter list is in the format of either an OS/VS CAMLST or a VSAM catalog management parameter list (CTGPL).

The CAMLST macro is used to generate the CAMLST when the CATALOG, INDEX, or LOCATE macro issues the SVC call.

CAMLST:

Bytes

0 First option byte:

Bits

1...	....	CVOL specified.
.1..	.1..	CAMLST macro generated - not used.
..1.	....	CATALOG or CATBX specified.
...1	....	RECAT (re-catalog) specified.
....	1...	UNCAT or UCATDX specified.
....	..1.	LOCATE by TTR specified.
....	...1	Reserved.

1 Second option byte:

Bits

1...	....	Do not allocate CVOL.
.1..	....	BLDX/CATBX specified.
..1.	....	BLDG specified.
...1	....	BLDA specified.
....	1...	LNKX specified.
....	.1..	DLTX/UCATDX specified.
....	..1.	DSCB TTR specified.
....	...1	DLTA specified.

2 Third option byte:

1...	....	DRPX specified.
.1..	....	Indicate DELETE option.
..1.	....	Reserved.
...1	....	SYSZTIOT is enqueued exclusively.
....	1...	Indicate the EMPTY option.
....	.11.	Reserved.
....	...0	The caller supplied a CAMLST.
....	...1	The caller supplied a CTGPL.

3 Maximum generation count for GDG or zeroes.

4-7 Address of the name or TTR if byte 0, bit 6 is "on".

8-11 Address of the CVOL volser or zeroes if byte 0, bit 0 is zero.

12-15
a. When cataloging, the address of the volume list.
b. Address of an 8-byte area that contains an alias for a high-level index. Note: CVOL must also be specified.
c. When performing LNKX, the address of a 10-byte area that contains a 4-byte device code followed by the 6-byte volume serial number of the CVOL to be connected.
d. When performing the LOCATE function, the address of a 256-byte work area that must be on a doubleword boundary. If the issuer of LOCATE has a non-zero protect key, then the work area must have a matching storage protect key.

16-19 Address of the DSCB TTR when cataloging it.

Entry from VSAM CATALOG (indicated by Byte 2, Bit 7 being on):

Offset	Bytes/Bits	Field	Description
0 (0)	1	CTGOPTN1	First option byte:
	1...	CTGBYPSS	Bypass the catalog management security verification processing.
	.1..	CTGMAST	Check the master password.
	..1.	CTGCI	Check the control interval password.
	...1	CTGUPD	Check the update password.
	.... 1...	CTGREAD	Check the read password.
	.... .1..	CTGNAME	The CTGENT field contains the address of a 44-byte DSNAME, or a 6-byte volume serial number (padded with binary 0s).
	.... .0..		The CTGENT field contains the address of a 3-byte control interval number.
	.... ..1.	CTGNAME	The CTGCAT field contains the address of a catalog's 44-byte DSNAME.
	.... ..0.		The CTGCAT field contains the address of a 4-byte field containing a VSAM catalog's ACB address.
	1	CTGGENLD	Generic locate request.
1 (1)	1	CTGOPTN2	Second option byte:
	1...	CTGEXT	Extend option (with UPDATE).
		CTGNSVS	Catalog cleanup request.
	.1..	CTGERASE	Erase option (with DELETE).
		CTGSMF	Write SMF record option (with LSPACE).
		CTGREL	Release number.
		CTGGTALL	Search all catalogs (with LISTCAT).
	..1.	CTGPURG	Purge option (with DELETE).
		CTGVMNT	The caller is VSAM Open/Close/EOV:
			Volume mount and verify routine (IDA0192V).
		CTGRCATN	Return the catalog name (with generic LOCATE).
	...1	CTGGTNXT	Get-next option (with LISTCAT).
	.... 1...	CTGDISC	Disconnect option (with EXPORT).
	.... .1..	CTGOVRID	Erase override option (with DELETE).
	.... ..1.	CTGSCR	Scratch space option (with DELETE NonVSAM). Force option (with DELETE GDG, CATALOG, or SPACE).
	x		Reserved.

2 (2)	1		CTGOPTN3	Third option byte:
	xxx.	....	CTGFUNC	Specifies the caller-requested function:
	001.	....	CTGLOC	LOCATE.
	010.	....	CTGLSP	LSPACE.
	011.	....	CTGUPDAT	UPDATE.
	100.	....	CTGCMS	A catalog management services function (see CTGOPTNS).
	...1	....	CTGSUPLT	SUPER-LOCATE function.
	....	1...	CTGGDGL	GDG locate request—the caller supplied the base generation level (CTGWAGB field in CTGWA).
	....	.1..	CTGSRH	Search the master catalog only.
	....	.0..		Search the user's catalog first (specified by CTGCAT or, if CTGCAT = 0, search the user's catalogs available to the caller via JOBCAT or STEPCAT DD statements, then search the master catalog).
	....	..x.		Reserved.
	....	...1	CTGAM0	The call is a CVOL catalog management request.
	....	...0		The call is an OS catalog management request; the caller supplied a CAMLST parameter list that was translated into this CTGPL and CTGFLs.
3 (3)	1		CTGOPTN4	Fourth option byte:
	1...	....	CTGLBASE	Locate the base level (with SUPERLOCATE-GDG only).
	.0..	....	CTGDOCAT	If the needed catalog is not open, dynamically allocate and open it.
	.1..	....		Do not dynamically open the needed catalog.
	...1	....		Controller intercept requested.
	....	1...		Bypass security prompting to the system operator.
	....	.1..		SYSZTIOT is enqueued.
	..x.	..xx		Reserved.
4 (4)	4		CTGENT	Address of the catalog record identifier, as defined in CTGOPTN1. When the request is a generic locate, byte 1 of CTGENT is a length byte, followed by a 1-to 43-character generic name.
			CTGFVT	Address of the caller's CTGFV.
8 (8)	4		CTGCAT	Address of the catalog's DSNAME or ACB, as specified in CTGOPTN1.
			CTGCVOL	Address of an OS/VS system-catalog catalog name area, if the request is SUPERLOCATE. The catalog name area contains the catalog's DSNAME and, if the catalog is identified with an alternate DSNAME, the catalog's alias. The OS/VS2 job scheduler uses this information to build the catalog's PCCB.
12 (C)	4		CTGWKA	Address of the caller's work area.
16 (10)	2		CTGDSORG	Data set organization, if the request is SUPERLOCATE.
16 (10)	1		CTGOPTNS	catalog management services request options:
	0000	1...	CTGDEFIN	DEFINE.
	0001	0...	CTGALTER	ALTER.
	0001	1...	CTGDELET	DELETE.
	0010	0...	CTGLTCAT	LISTCAT.
	0011	0...	CTGCNVTV	CONVERT.
	....	.xxx		Reserved.

17 (11)	1		Reserved.
18 (12)	1	CTGTYPE	Type of catalog record:
	C'A'	CTGTALIN	NonVSAM data set.
	C'B'	CTGTGBS	Generation data group (GDG) base.
	C'C'	CTGTCL	Cluster.
	C'D'	CTGTDATA	Data set.
	C'G'	CTGTAIX	Alternate index.
	C'I'	CTGTINDX	Index.
	C'M'	CTGTMCAT	Master catalog.
	C'P'	CTGTPGS	Page space.
	C'R'	CTGTPTH	Path.
	C'U'	CTGTUCAT	User catalog.
	C'V'	CTGTVOL	Volume.
	C'X'	CTGTANM	Alias name.
	C'Y'	CTGTUPG	Upgrade.
19 (13)	1	CTGNOFLD	Number of entries contained in CTGFIELD.
20 (14)	4	CTGDDNM	Address of the JCL DD statement, if one is associated with this request.
		CTGNEWNM	Address of the new DSNAME, if the request is ALTER and the object's name is being changed.
If the request is SUPERLOCATE:			
20 (14)	2	CTGFDBK	Feedback area.
22 (16)	1	CTGFBFLG	Flags:
	1...	CTGPAR	Parallel mount.
	.1..	CTGKEEP	Forced keep.
	..1.	CTGGDGB	GDG Base located.
	...1	CTGNGDSN	Generation data set name was generated (in the form 'dsname.gxxxxvyy').
	 xxxx		Reserved.
23 (17)	1		Reserved.
24 (18)	4	CTGJSCB	Address of the JSCB.
		CTGPSWD	Address of the caller-supplied password.
28 (1C)	VL	CTGFIELD	The 4-byte address of each CTGFL, to specify each catalog field to be processed. The length of CTGFIELD is the CTGNOFLD value times 4.

SVC 27 (0A1B) OBTAIN macro - is type 3, gets LOCAL lock, calls module IGC0002G. PLM is *OS/VS2 DADSM Logic*. GTF data is:

R15 and R0 - No applicable data.
R1 Address of the parameter list.
PLIST - Parameter list is sixteen bytes long; format is:

Bytes	
0-3	Operation code. X'C1000000' SEARCH for DSNAME. X'C0800000' SEEK for track address.
4-7	Address of the data set name or address of the track address of the DSCB (CCHHR) depending on the operation code.
8-11	Address of the volume serial number.
12-15	Address of a 140-byte workarea.

VOLSER ccccc volume number of an associated volume.
DSN/CCHHR ccccc... data set name (displayed when the operation code in word 1 of the parameter list indicates SEARCH), or track address (displayed when the operation code in word 1 of the parameter list indicates SEEK).

SVC 28 (0A1C) Reserved.

SVC 29 (0A1D) SCRATCH macro - is type 3, gets LOCAL lock, calls module IGC0002I. PLM is *OS/VS2 DADSM Logic*. GTF data is:

R15 No applicable data.
R0 Zeros or the address of a UCB for a device upon which volumes can be mounted.
PLIST - The parameter list is sixteen bytes long; format is:

Bytes	
0-3	Operation code as follows: If bit 1 of byte 1 is set to 1 and JSCBPASS=1 then the RACDEF macro is not issued. 4100 4000 Check purge date. 4100 5000 Override purge date. 4180 4000 Check purge date (job or step termination). 4180 5000 Override purge date (job or step termination). 4180 C000 Check purge date (dynamic unallocation). 4180 D000 Override purge date (dynamic unallocation).
4-7	Address of the data set name.
8-11	Not used.
12-15	Address of the volume list.

DSN ccccc.... data set name.
VOLIST the volume list is variable in length; format is:

Bytes	
0-1	Number of 12-byte volume list entries to follow.
2-5	UCB device code.
6-11	Volume serial number.
12	Reserved.
13	Scratch status byte.

Note: Each succeeding volume list entry (if any) has the same format as offset 2-13.

SVC 30 (0A1E) RENAME macro - is type 3, gets LOCAL lock, calls module IGC00030. PLM is *OS/VS2 DADSM Logic*. GTF data is:

R15 No applicable data.
R0 Address of the UCB for the device on which volumes can be mounted, or zeros.
R1 Address of the parameter list.
PLIST - The parameter list is sixteen bytes long; format is:

Bytes	
0-3	X'C1002000' If bit 1 of byte 1 is set to 1 and JSCBPASS=1 then the RACDEF macro is not issued.
4-7	Address of the old data set name.
8-11	Address of the new data set name.
12-15	Address of the volume list.

OLDDSN ccccc... fully qualified name of the data set to be renamed.
NEWDSN ccccc... new name of the data set being renamed.
VOLIST the volume list is variable in length; format is:

Bytes	
0-1	Number of 12-byte volume list entries to follow.
2-5	UCB device code.
6-11	Volume serial number.
12	Reserved.
13	Rename status byte.

Note: Each succeeding volume list entry (if any) has the same format as offset 2-13.

SVC 31 (0A1F) FEOV macro - is type 4, gets LOCAL lock, calls module IGC0003A. PLM is *OS/VS2 OPEN/CLOSE/EOV Logic*. GTF data is:

R15 and R0 - No applicable data.
R1 High-order byte—flags as follows:
00 No option specified.
20 REWIND specified.
30 LEAVE specified.
Three low-order bytes—address of DCB.
DDNAME cccccccc DDNAME associated with this request.

SVC 32 (0A20) No macro - is type 4, gets LOCAL lock, calls module IGC0003B. PLM is *OS/VS2 DADSM Logic*. APF protected. GTF data is:

R15 No applicable data.
R0 If positive, contains address of associated JFCB.
If negative (not complemented — high-order bit is set on), contains the address of the associated partial DSCB.
R1 Address of a fullword containing the UCB address; may be zero.
CUU cccc unit address from the UCB pointed to by R1.
DSN cccc... data set name from the DSN field of either the JFCB or DSCB pointed to by R0.

SVC 33 (0A21) IOHALT macro - is type 3, gets IOSUCB lock, calls module IGC0003C. PLM is *OS/VS2 I/O Supervisor Logic*. GTF data is:

R15 No applicable data.
R0 If byte 1 of R1 is X'80' then R0 contains the offset from the IOB to the virtual CCW which corresponds to the real CCW to be modified to a NOP.
R1 Contents:

Bytes		
0	Ignored.	
1	Input	Option.
	= '00'	use IOS HALT I/O subroutine.
	= '80'	use EXCP CCW modify subroutine.
2-3	Address of the UCB associated with the HALT request.	

CUU cccc The device address associated with the device being halted.

SVC 34 (0A22) MGCR/QEDIT macro - is type 4, gets LOCAL and CMS lock, calls module IEE0003D. PLM is *OS/VS2 System Logic Library*.

For a system task issuing SVC34:

R1	R0	FUNCTION or ACTION
POS	N/A	0, UCMID, TSO ASID, or command authority.
ZERO	POS	Free the CIB chain.
ZERO	ZERO	Error: return code 8, BR 14.
ZERO	NEG	Set CHCIBCT to zero.
NEG	POS	CIB chain ADD or DELETE.
NEG	ZERO	CSCB processing.
NEG	NEG	Store R1 in CHCIBCT.

For a problem program issuing SVC34:

R1	R0	FUNCTION or ACTION
POS	N/A	Return code 8, BR14.
ZERO	POS	Return code 8, BR14.
ZERO	ZERO	Return code 8, BR14.
ZERO	NEG	Set CHCIBCT to zero.
NEG	POS	Delete the CIB.
NEG	ZERO	CSCB processing.
NEG	NEG	Store R1 in CHCIBCT.

SVC 35 (0A23) WTO/WTOR macro - is type 4, gets LOCAL and CMS locks, calls module IEAVVWTO. PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.
R1 Address of the parameter list.
R0 Three high order bytes—a new line is to be connected to the message with this three byte message ID.
Low order byte-console ID.
Length of PLIST (1 byte); includes routing and descriptor code field, if present.
PLIST - 12 bytes long for WTO; 20 bytes long for WTOR. *Note:* If routing and descriptor codes are present, they will be appended to the parameter lists, increasing the lengths of the parameter lists by four bytes.

PLIST for WTO:

Bytes		
0	00	Indicates WTO parameter list.
1	Message length plus four.	
2	MCS flag byte, bit settings are:	
	1...	Routing and descriptor codes follow the message text.
	.1..	Message is to be queued to the console whose source ID is in Register 0.
	..1.	WTO is an immediate command response.
	...1	Message type field exists.
	.... 1...	WTO reply to a WTOR macro instruction.
	.... .1..	Message should be broadcast to all active consoles.
	.... ..1.	Message queued for hard copy only.
	1	Message queued unconditionally to the console whose source ID is in register 0.
3	Second MCS flag byte: bit settings are:	
	1...	Do not timestamp this record.
	.1..	Message is a multiline WTO.
	.... .1..	Message is not queued for hard copy.
	..11 1.11	Invalid entry.
4-11	First 8 bytes of the message text. Normally, the message ID.	
12-15	Routing and descriptor codes, if present.	

PLIST for WTOR:

Bytes	
0	Length of the reply.
1-3	Address of the reply buffer.
4-7	Address of the reply ECB.
8	Zeros.
9	Message length +4.
10-11	MCS flag bytes (see WTO PLIST).
12-19	First 8 bytes of the message text. Normally, the message ID.
20-23	Routing and descriptor codes, if present

SVC 36 (0A24) WTL macro - is type 4, gets no lock, calls module IEEMB804. PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.
R1 Pointer to the WTL parameter list.

SVC 37 (0A25) SEGLD/SEGWT macro - is type 2, gets no lock, calls module IEWSUOVR (E.P. = IGC037). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.
R0 Zero entry was from SEGLD.
Nonzero entry from SEGWT.
R1 Address of the parameter list.
PLIST - 12 bytes, format is:

Bytes	
0-3	Branch instruction to a SVC 45.
4-7	Address of the referred-to-symbol.
8	'TO' segment number.
9-11	Previous caller or zero.

SVC 38 (0A26) Reserved.

SVC 39 (0A27) LABEL macro - is type 3, gets no lock, calls module IGC0003I. PLM is *OS/VS Utilities*. APF protected. GTF data is:

R15 and R0 - No applicable data.
R1 Address of the parameter list.
PLIST - 20 bytes long, format is:

Bytes		
0-2	C00004	REWIND option.
	C00000	UNLOAD option.
3	Relative UCB in the TIOT to use for mounting purposes.	
4-7	Address of the 8-byte ddname for the DD card that allocates the device.	
8-11	Address of the volume label set.	
12-13	Length of one volume label.	
14	Number of labels in the volume label set.	
15	Command byte of the control CCW.	
16-19	Address of the first 10 bytes of the volume header label.	

SVC 40 (0A28) EXTRACT macro - is type 3, gets LOCAL lock, calls module IEAVTB00 (E.P. = IGC00040). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.
R1 Address of the parameter list.
PLIST - 12 bytes long; format is:

Bytes		
0	Reserved; should be zeros.	
1-3	Address of the list area in which the extracted information will be stored.	
4	Reserved; should be zeros.	
5-7	Address of the TCB from which the EXTRACT will get requested information.	
	Zeros indicate that the EXTRACT will get information from the current TCB and/or its related control blocks.	
8	Flag bytes that indicates the fields to be extracted:	
	Bits	
	1...	Address of the general register save area.
	.1..	Address of the floating point register save area.
	..0.	Reserved.
	...1	Address of the end-of-task exit routine.
	.... 1...	Limit priority and dispatching priority.
	.... .1..	Task completion code.
	.... ..1.	Address of the TIOT.
	1	Address of the command scheduler communication list in the CSCB.
9	Flag Byte 2	
	Bits	
	1...	Address of a byte. If the high order bit is 1, it indicates a TS address space.
	.1..	Address of the protected storage control block.
	..1.	ASID (only if a TS address space). Where AUTH ONLY is indicated, the parameter is valid only for an authorized task-authorized by system key, supervisor state, or APF authorized. If the attaching task isn't authorized, the parameter is ignored.
	...1	ASID.
	 xxxx	Reserved.
10-11	Reserved; should be zeros.	

SVC 41 (0A29) IDENTIFY macro - is type 3, gets LOCAL and CMS locks, calls module IEAVID00 (IGC041). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.
R0 Entry point name address or zero.
R1 Address of the entry point name being added or of the parameter list.
EPNAME cccccccc The entry point name being added.

SVC 42 (0A2A) ATTACH macro - is type 3, gets LOCAL lock, calls module IEAVEAT0 (IGC0004B). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 Address of the parameter list being passed to the SVC routine.

R0 No applicable data.

R1 Address of the parameter list being passed to the called program, or zeros (no parameter list being passed).

Length of SUPRVLIST (1 byte), =36 if byte 20 (below) = X'00', =60 if byte 20 X'20' = X'20'.

SUPRVLIST - The parameter list passed to the SVC routine is 36 or 60 bytes long (seę byte 20). Where AUTH ONLY is indicated, the parameter is valid only for a task operating in a system key or in supervisor state. If an attaching task is not a system task, the parameter is ignored. Format is:

Bytes

0 EP/DE flag byte:

00 EP or EPLOC specified.

80 DE specified.

1-3 Address of the EP name or directory entry (determined by byte 0).

4 No applicable data.

5-7 Address of the DCB; or zeros.

8 Flag bytes as follows:

0000	.0..	Reserved.
....	1...	Field ATTPLNG is present.
....	..1.	TASKLIB is provided.
....	...1	STAI or ESTAI exit is provided.

9-11 Address of the ECB.

12 GSP flag byte; bit settings are:

00 bytes 13-15 contain the subpool number or zero.

01 bytes 13-15 contain the address of a list of subpool numbers to be given to the subtask.

13-15 Subpool number or address of the subpool list (determined by byte 12), or zero.

16 SHSP flag byte; bit settings are:

00 bytes 17-19 contain a subpool number or zero.

01 bytes 17-19 contain the address of a list of subpool numbers to be shared.

17-19 Subpool number or address of a subpool list (determined by byte 16), or zero.

20 Flags

00.0	0000	Reserved.
..1.	....	the SUPRVLIST is a VS2 extended SUPRVLIST and is 60 bytes long (see bytes 8 and 54). Byte 36 contains the LSQA value and byte 40 contains the TASK ID.

21-23 Address of the end-of-task exit routine.

24-25 Dispatching priority number.

26 Limit priority number.

27 Key flags byte; bit settings are:

Bits

0...	....	Leave the task dispatchable (DISP=YES).
1...	....	Set TCBANDSP in the new TCB (DISP=NO)—AUTH ONLY.
.0..	....	Propagate the JSCB field from the originating task.
.1..	....	Move the specified JSCB address into the attached TCB—AUTH ONLY.
..0.	....	Reserved.
...1	....	Attached task is to have a protect key of 0—AUTH ONLY.
...0	....	Propagate the key of the originating task.
....	0...	Subpool zero will be shared with the subtask.
....	1...	Subpool zero will not be shared.
....	.0..	Save area of 72 bytes will be obtained for the task.
....	.1..	No save area will be obtained—AUTH ONLY.
....	..0.	Propagate the TCBJSTCB field from the originating task.
....	..1.	TCBJSTCB of the new task will point to the new task TCB—AUTH ONLY.
....	...0	New task will operate in problem program mode.
....	...1	New task will operate in supervisor mode—AUTH ONLY.

28-35 Entry point name for EP; blank or zeros for EPLOC or DE specification.

36 Reserved.

37-39 JSCB address.

40 Task ID—AUTH ONLY.

41-43 Address of the parameter list for the STAI or the ESTAI routine.

44 Flag bytes:

....	..00	Purge = QUIESCE specified.
....	..01	Purge = HALT specified.
....	..10	Purge = NONE specified.
....	.1..	ASYNCH = YES.
....	1...	ESTAI was specified.
...1	....	TERM = YES was specified.
000.	....	Reserved.

45-47 Address of the routine to get control if the subtask ABENDs.

48 Reserved.

49-51 Address of the opened TASKLIB DCB.

52 Indicators:

0...	....	Word 56 contains a subpool number (AUTH ONLY) or zero.
1...	....	Word 56 contains the address of a list of subpools—AUTH ONLY.
.1..	....	Reset APF.
..00	0000	Reserved.

53 Reserved.

54 Length of the parameter list in bytes.

56 A subpool number or a list of subpool numbers. See byte 52—AUTH ONLY.

Length of PLIST (1 byte)

PLIST - PLIST up to 40 bytes of parameter list passed to a program. The parameter list is a series of 4-byte entries. Each entry has its high-order byte reserved and an address in the low-order three bytes.

Note: Contents of register 1 at the time SVC 42 is issued are passed to the ttached program.

SVC 43 (0A2B) CIRB macro - is type 1, gets LOCAL lock, calls module EAVEF00 (E.P. = IGC043). PLM is *OS/VS2 System Logic Library*. GTF data s:

R15 No applicable data.

R0 Entry point address of the user's asynchronous exit routine. When the routine is dispatched it will get control at this entry point.

R1 The meanings of the bytes of the register are as follows:

Byte 1

0100	0...	A normal IRB is being created.
1000	0...	An SIRB is being created. This is used only by IOS to run ERP routines.
....	.1..	Problem program key.
....	.0..	Supervisor key.
....	..1.	Problem program state.
....	..0.	Supervisor state.
....	...1	Save area for registers requested.
....	...0	No save area requested.

Byte 2

0000	...0	Reserved - always zero.
....	1...	Indicates that the IQE's are going to schedule the routine.
....	0...	Indicates that the RQE's are going to schedule the routine
....	.1..	Return the IQEs at exit if the IRB has a work area and the RBUSIOE flag is not on.
....	.0..	Do not return the IQE's at exit.
....	..1.	Indicates that the RB will be freed when the exit issues an SVC 3.
....	..0.	Indicates that the RB will not be freed when the exit issues an SVC 3.

Byte 3

Reserved.

Byte 4

Indicates the size in doublewords of the work area to be acquired. CIRB will unconditionally request space from subpool 253. The maximum size is 255 doublewords.

SVC 44 (0A2C) CHAP macro - is type 2, gets LOCAL lock, calls module IEAVECH0 (IGC044). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Signed value to be added to the dispatching priority of the specified task; negative value will be in two's-complement form.

R1 Address of the area containing the address of TCB whose priority is to be changed; or zeros. Zeros indicates that the active task's priority is to be changed.

CHAP TCB hhhhhhhh Address of the TCB whose priority is to be changed. Must be a subtask of the current task.

SVC 45 (0A2D) OVLYBRCH macro - is type 2, gets no lock, calls module IEWSUOVR (E.P. = IGC045). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 Address of the entry-table entry that caused the SVC to be issued.

R0 and R1 - No applicable data.

PLIST - 12 bytes long; format is:

Bytes	
0-3	Branch instruction to SVC 45.
4-7	Address of the referred-to symbol.
8	'To' segment number.
9-11	Previous caller or zero.

SVC 46 (0A2E) TTIMER macro - is type 3, gets no locks, calls module IEAVRT00 (E.P. = IGC0004F). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Address of the doubleword where time in microseconds is to be returned.

R1 Low-order byte has code determining the type of request and the format of the returned value.

Code		
....	.000	Time remaining in the current task's time interval is to be in register 0 in timer units; the time interval is not to be canceled.
....	...1	Current task's time interval is to be canceled.
....	..1.	Same as .....000 except the interval remaining is returned to the specified address in the TOD clock format.
....	.1..	Routine specified by the ERRET parameter gets control on an environmental error.

SVC 47 (0A2F) STIMER macro - is type 3, gets no locks, calls module IEAVRT00 (E.P. = IGC0004G). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Contents:

Bytes			
0	STIMER option byte as follows:		
	0000	....	TUINTVL option.
	0001	....	BINTVL option.
	0010	....	MICVL option.
	0011	....	DINTVL option.
	0110	....	GMT option.
	0111	....	TOD option.
	....	.000	Interval to be decreased only when the task is active.
	....	.001	Decrease the interval continuously and put the task in the wait state until the interval expires.
	....	.011	Decrease the interval continuously.
	....	1...	ERRET bit: Control is returned on errors with register 15 set to 8.
1-3	Exit address.		

R1 Address of the time value.

PLIST - 4 or 8 bytes depending on option in force:

a. DINTVL,TOD,MICVL, and GMT - 8 bytes; represents the time value.

b. BINTVL and TUINTVL - 4 bytes; represents the time value.

SVC 48 (0A30) DEQ macro - is type 2, gets LOCAL and CMSEQDQ locks, calls module IEAVENQ1 (E.P. = IGC048). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list.

PLIST - 20 bytes of DEQ parameter list, representing a DEQ request for a single resource. The complete parameter list may include requests for up to 65,535 resources.

Bytes

-4 TCB address when TCB= is specified (see flag byte 0) otherwise, contents are unpredictable.

0 Flag bits as follows:

0000	0000	List request
1...	....	End-of-list indicator; if zero, the parameter list contains another request. Up to 65,535 requests may be included in one parameter list
.1xx	xxxx	Old options are in effect.
.0..	....	New options are in effect (bits 2-7 have meaning).
.011	11..	Reserved.
.0..	..1.	A generic DEQUEUE (by major name) was requested.
.0..	...1	'TCB=tcbaddr' was requested; parmlist prefix contains the TCB address.

1 Length of the minor name whose address is in bytes 8 - 11 of this element.

Zeros indicate that the length of the minor name is in the first byte of the minor name field whose address is in bytes 8 - 11 of this element (does not include length byte itself).

2 DEQ parameter byte; bit settings are:

Bits		
0...	....	Reserved.
.0..	0...	Scope of the minor name is STEP.
.0..	1...	Resource is known across systems, and UCB= was specified. (This combination means that the last word in the parmlist contains the UCB address.)
.1..	0...	Scope of the minor name is SYSTEM.
.1..	1...	Scope of the minor name is SYSTEMS.
..1.	....	Obsoleted.
...1	....	Reset 'must complete'.
....	.000	RET=NONE.
....	.001	RET=HAVE.

3 Return code field for codes returned to the issuer by DEQ.

4-7 Address of the major resource name (QNAME).

8-11 Address of the minor resource name (RNAME).

12-15 If bits 1 and 4 of the DEQ parameters byte are set to 0 and 1 respectively, this word contains the UCB address; otherwise, the content of this word is unpredictable.

SVC 49 (0A31) Reserved.

SVC 50 (0A32) Reserved.

SVC 51 (0A33) SNAP/SDUMP macro - is type 4, gets LOCAL, CMS, CMSEQDQ, and SALLOC locks, calls module IEAVAD00. PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list.

PLIST - (SNAP only) parameter list is 20 bytes long and is passed to SNAP by SVC 51; format is:

Bytes				
0	ID number to be printed in the identification heading of the dump.			
1	SNPFLAG, if 00 SNAP parameter list is a MVT type.			
	SNPSNAP	1...	....	Requests a SVC dump; otherwise, a SNAP dump will be taken.
	SNPVS2EN	..1.	....	Parameter list in MVS.
		..x.	....	Reserved.
	SNPABEND	...1	....	SNAP dump requested; otherwise, called by ABEND.
	SNPID	....	1...	ID specified.
	SNPTCV	....	.1..	TCB specified.
	SNPSLIST	....	..1.	Storage list specified.
		....	...x	Reserved.
	SNPHDR	....	...1	Storage header list specified.
2	Reserved.			
3	Reserved.			
4	SNPS data	SDATA options.		
	SNPNUC	1...	....	Dump the nucleus, PSA, SQA, LSQA.
	SNPSQA	.1..	....	Dump the SQA.
	SNPLSQA	..1.	....	Dump the LSQA.
	SNPSWA	...1	....	Dump the SWA.
	SNPTRT	....	1...	Include the trace table (supervisor or GTF).
	SNPCB	....	.1..	Format the control blocks for the task.
	SNPQCB	....	..1.	Format the enqueue control blocks for the task.
		....	...x	Reserved.
	SNPDM	....	...1	Format the data management control blocks.
5	SDATA options.			
	SNPDM	....	...1	Format the IOS control blocks.
	SNPERR	.1..	....	Format the RTM control blocks.
6	SNPPDATA	PDATA options.		
	SNPSAVE	1...	....	Display the save area trace.
	SNPSAVE2	.1..	....	Display the save area; otherwise, display the entire save area.
	SNPREGS	..1.	....	Display the registers at entry to ABEND or SNAP.
	SNPLPA	...1	....	Display the active LPA modules.
	SNPJPA	....	1...	Display the job pack area modules.
	SNPPSW	....	.1..	Display the PSW, ILC, and interruption code.
	SNPSPLS	....	..1.	Display the user subpools, 0 through 127.
		....	...x	Reserved.
7	Reserved.			
8-11	Address of the DCB for the dump data set.			
12-15	Address of the TCB to be displayed.			
16-19	Address of the storage parameter list containing the starting and ending address of the areas to be dumped.			
20-23	Address of the header parameter list containing the address of headers to be used when dumping storage areas.			

MODN cccccccc name of the module issuing the SVC call.

PLIST - (SDUMP) parameter list for SVC dump requests is 40 bytes long and is passed to SVC dump by SVC 51; format is:

Bytes				
0	SDUFLAG0	First byte of SVC dump flags.		
	SDUDCB	1...	....	Caller supplied the data set.
	SDUBUF	.1..	....	Dump the 4K SQA SDUMP buffer.
	SDUSTOR	..1.	....	Storage range list specified.
	SDUHDR	...1	....	Caller supplied the title information.
	SDUECB	....	1...	ECB address is supplied by the caller.

	SDUASID	.... .1..	ASID specified by the caller.
	SDUQUIET	.... ..1.	Set the system non-dispatchable while performing the SQA/CSP dump.
	SDUBRANH	1	Branch entry to SVC dump.
1	SDUFLAG1	Second byte of the SVC dump flags.	
	DUMPTYPE	1...	SVC dump request.
	SDUABEND	.1..	SYSDUMP request type.
	SDUNEW	..1.	Enhanced SVC dump request.
	SDUASLST	...1	ASIDLST parameter specified.
	SDUSULST	.... 1...	SUMLIST parameter specified.
	SDUIGNCD	.... .1..	Ignore the specified CHNGDUMP parameter.
		xx	Reserved.
2	SDUSDAT1	SDATA dump options.	
	SDUALPSA	1...	Dump all PSA's.
	SDUPSA	.1..	Dump only the current PSA.
	SDUNUC	..1.	Dump the resident nucleus.
	SDUSQA	...1	Dump the SQA.
	SDULSQA	.... 1...	Dump the LSQA.
	SDURGN	.... .1..	Dump the private area including the LSQA.
	SDULPA	.... ..1.	Dump the active LPA.
	SDUTRT	1	Dump the system trace table.
3	SDUSDAT2	SDATA dump options.	
	SDUCSA	1...	Dump the CSA.
	SDUSWA	.1..	Dump the SWA.
	SDUSMDMP	..1.	Dump the summary dump data.
	SDUNSMDP	...1	Do not dump the summary dump data.
	SDUNAPSA	.... 1...	Do not dump all PSAs.
	SDUNSQA	.... .1..	Do not dump the SQA.
		xx	Reserved.
4-7	SDUDCBAD	Address of the caller supplied DCB.	
8-11	SDUSTORA	Address of the storage range list.	
12-15	SDUHDRAD	Address of the dump title.	
16-19	SDUECBAD	Address of the caller supplied ECB.	
20-21	SDUCASID	Caller's ASID.	
22-23	SDUTASID	ASID where the SVC dump is taken.	
24-27	SDUASIDP	Address of the caller supplied ASID list.	
28-31	SDUSUMLP	Address of the caller supplied SUM list.	
32-35	SDUSYSMS	Address of the SYSMDUMP SQA work area.	
36-39	SDUSYSMC	Address of the SYSMDUMP CSA work area.	

SVC 52 (0A34) RESTART macro - is type 4, gets LOCAL, CMS, and SALLOC locks, calls module IEFRSTRT. PLM is *OS/VS2 Checkpoint Restart Logic*. APF protected. GTF data is:

R15 and R0 - No applicable data.

R1 Address of parameter list, SVC 52 is issued from module IEFRSTRT to initiate a checkpoint restart.

Parameter list contains:

REPLNGTH	H	Length of the parameter list.
REPCIRAD	F	TTR of CIR records in the checkpoint data set entry.
REPCOUNT	H	Number of checkpoints taken.
REPCKIDL	H	Length of check ID.
REPCHKID	4F	Checkid.
REPDDNM	2F	DD Name of the checkpoint data set.
REPPPM	F	Low order address of the P/P area.
REPPPE	F	Size of the P/P area.
REPBLKSI	H	Checkpoint data set blocksize.
REPTIOTL	H	Length of the TIOT
REPFLAGS	CL1	Checkpoint flag byte 1.
REPWACL	CL3	Checkpoint work area length.
REPFLAG2	CL1	Checkpoint flag byte 2.
RSCKPPML	CL4	V=R Tests.

SVC 53 (0A35) RELEX macro - is type 3, gets no lock, calls module IGC0005C. PLM is *OS/VS2 BDAM Logic*. GTF data is:

R15 No applicable data.

R0 If R1 is negative, no applicable data.
If R1 is positive, the address of a parameter list that contains:
HHHHHHHH Relative block or TTR
or
MBBCCHHR Actual address.

R1 If positive, SVC was part of a RELEX macro call and R1 contains the DCB address.
If negative, SVC was issued as part of some BDAM exclusive control processing and R1 contains the two's complement of the IOB address.

DDNAME cccccccc DDNAME associated with this request.

SVC 54 (0A36) DISABLE macro - is type 2, gets LOCAL lock, calls module IGC0005D. PLM is *OS/VS2 ISAM Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the associated DCB.

DDNAME cccccccc name of the associated DD statement.
DCB hhhhhhhh address of the associated DCB.
DEB hhhhhhhh address of the associated DEB.

SVC 55 (0A37) EOV macro - is type 4, gets LOCAL lock, calls module IGC0005E. PLM is *OS/VS2 OPEN/CLOSE/EOV Logic*. GTF data is:

R15 No applicable data.

R0 IOB address if:
DCBOFLAGS = ...1 ....
DCBMACRF = 0... ....
or
Internal code for problem determination if the high order byte of R1 is X'FF', indicating DMABCOND was issued with the SVC = YES parameter,
or
X'0000 1000', indicating entry from access method routines to request a 001 ABEND.

R1 DCB address. If high-order byte of R1 is X'FF', R0 is expected to contain an internal code for problem determination.

DDNAME cccccccc DDNAME associated with this request.

SVC 56 (0A38) ENQ/RESERVE macro - is type 2, gets LOCAL and CMSEQDQ locks, calls module IEAVENQ1 (E.P. = IGC056). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list.

PLIST - 20 bytes long of the ENQ/RESERVE parameter list, representing a request for a single resource. The complete parameter list may include requests for additional resources; the last request is identified by a flag bit described below.

Bytes

-8 TCB address if both TCB and ECB were specified.

-4 TCB or ECB address depending on whether TCB= or ECB= was specified. (See flag bytes).

0 Flag bits as follows:

1...	....	End-of-list indicator; if zero, the parameter list includes another resource request.
.1xx	xxxx	Old options are in effect.
.0..	....	New options are in effect (bits 2-7 have meaning).
.011	111.	Reserved.
.0..	...1	'TCB=tcbaddr' was requested. TCB address is contained in the parameter list prefix.

1 Length of the minor name whose address is in bytes 8 - 11 of this element.
Zeros indicates the length of minor name is in the first byte of the minor name field whose address is in bytes 8 - 11 of this element (does not include the length byte itself).

2 ENQ parameters byte; bit settings are:

Bits		
0...	....	Exclusive request.
1...	....	Shared request.
.0..	0...	Scope of the minor name is STEP.
.0..	1...	RESERVE type. The resource is known across systems + UCB= was specified. The last word of the parameter list is the address of a word containing the UCB address.
.1..	0...	Scope of the minor name is SYSTEM.
.1..	1...	Scope of the minor name is SYSTEMS.
..1.	....	Obsoleted.
...1	....	Set "must complete" equal to STEP.
....	.000	RET=NONE.
....	.001	RET=HAVE.
....	.010	RET=CHNG.
....	.011	RET=USE.
....	.100	'ECB=addr'. The ECB address is contained in the parameter list.
....	.111	RET=TEST.

3 Field for codes returned to the issuer by ENQ.

4-7 Address of the major resource name (QNAME).

8-11 Address of the minor resource name (RNAME).

12-15 If bit 4 of the ENQ parameters byte (RESERVE) is set on, this word contains the address of a word containing the UCB address; otherwise, the content of this word is unpredictable.

Note: RESERVE is basically an ENQ with UCB= specified. See flag byte 2.

SVC 57 (0A39) FREEDBUF macro - is type 3, gets no lock, calls module IGC0005G. PLM is *OS/VS2 BDAM Logic*. GTF data is:

R15 No applicable data.

R0 DECB address. The address is in two's complement form and indicates an extended function.

R1 DCB address.

DDNAME cccccccc DDNAME associated with this request.

SVC 58 (0A3A) RELBUF/REQBUF macro - is type 1, gets local lock, calls module IGC058. PLM is *OS/VS2 BTAM Logic*. GTF data is:

R15 No applicable data.

R0 Request count or release address.

R1 DCB address.

DDNAME cccccccc DDNAME associated with this request.

SVC 59 (0A3B) OLTEP macro - is type 3, gets LOCAL and CMS locks, calls module IGC0005I. PLM is *OS/VS2 OLTEP Logic*. APF protected via TESTAUTH. GTF data is:

R15 No applicable data unless specified

- R1=00 To remove an outstanding WTOR from the RQE chain.
  R0= RSRM address:
  Word 0= pointer to the ECB.
- R1=04 UCB lookup for the control unit test.
  R0= RSRM address:
  Word 0 = Base address of the control unit.
  Word 1 = bytes 0,1 - number of devices on the control unit.
  bytes 2,3 - a code (0 or 1)
  R1=08 To determine if OLTEP is in a MP environment.
  R1=0C To vary offline a 3830 attached to a 3850 mass storage system.
  R1=10 To put a 3330 SSID (when attached to a 3850 mass storage system) into a list for cleanup.
- R1=14 To cleanup the UCBs and DEB chains and zero the CVTOLTEP word.
  R1=18 No function performed.
  R1=1C No function performed.
  R1=20 No function performed.
  R1=24 No function performed.
  R1=28 No function performed.
  R1=2C No function performed.
  R1=30 No function performed.
  R1=34 No function performed.
  R1=38 No function performed.
- R1=3C To check online or offline status.
  R0= RSRM address:
  Word 0 = pointer to the UCB.
  Word 1 = pointer to the 8-byte workarea.
- R1=40 UCB lookup for each DEVTAB entry.
  R0= RSRM address:
  Word 0 = pointer to the DEVTAB.
  Word 1 = number of entries in DEVTAB.
  Word 2 = pointer to the save area.
- R1=44 No function performed.
  R1=4C To translate a real address to a virtual address.
  R1=50 OLTEP will purge an I/O event and free the necessary control blocks and areas.
- R1=48 Move the OLTEP pseudo DEB (2 DEBs if R1=8048) to a protected subpool and initialize or update the CVTOLTEP DEB pointers.
  R0= RSRM address:
  Word 0 = pointer to the TDEB.
  Word 1 = pointer to the subpool.
- R1=54 Test UCB not ready bit.
  R0= RSRM address:
  Word 0 = pointer to the UCB.
- R1=58 Initialization (MVS).
  R0= RSRM address:
  Word 0 = DIE address.
  Word 1 = DIEPTR address.
  Word 2 = TESTDEB address.
  R15= 00 OK
  04 Second OLTEP.
  08 PGFIX for subpool 245 failed.
- R1=5C STARTIO - Move the IOSB.
  R0= RSRM address:
  Word 0 = Model the IOSB address.
  R15= 00 IOSB moved and the STARTIO issued.
  04 SRB/IOSB set not available, all are in use.
  XX CCW translator failure. XX is the return code from the translator.
- R1=60 Verify and set the processor affinity
  R0= RSRM address:
  Word 0 = pointer to the requested affinity.
  R15= 00 OK.
  04 Requested affinity cannot be set.

SVC 60 (0A3C) STAE/STAI-ESTAE/ESTAI macro - is type 3, gets LOCAL lock, calls module IEAVSTA0 (E.P. = IGC00060). PLM is *OS/VS2 System Logic Library*. GTF data is:

FOR STAE/STAI REQUESTS

R15 No applicable data.

R0 Contents:

- 00 Create.
- 04 Cancel.
- 08 Overlay.

R1 Address of the parameter list. The high-order bit is set to 1 if the XCTL=YES parameter was coded.

PLIST 12 bytes long; format is:

Bytes			
0	Flag byte:		
	1...	....	TCB address is supplied.
	....	.1..	Allow asynchronous exit scheduling.
	....	..10	Do not purge I/O operations.
	....	..01	Purge I/O operations with the halt option.
	....	..00	Purge I/O operations with the quiesce option.
	.xx1	x...	Reserved and set to zero.
1-3	If zero, the CANCEL operand is in effect; otherwise, the address of the STAE/STAI exit routine.		
4-7	Address of the exit routine parameter list; if zero, no exit routine parameter list exists.		
8-11	TCB address for a STAI request.		

FOR ESTAE/ESTAI REQUESTS

R15 No applicable data.

R0 Contents:

- 00 A new ESTAE parameter list is to be created.
- 02 Previous STAI/ESTAI exits are to be propagated from the originating task.
- 04 Cancel the most recent STAE request.
- 84 Cancel the most recent ESTAE request.
- 94 Branch enter to cancel the most recent ESTAE request.
- 08 Overlay the previous ESTAE parameter list with the parameters passed in this request.

R1 Address of the parameter list. The high-order bit is set to 1 if the ESTAE macro is not to be canceled when an XCTL is issued, and to 0 if the ESTAE macro is to be canceled when an XCTL is issued.

PLIST 12 bytes long; format is:

Bytes			
0	Flag byte:		
	1...	....	TCB address is supplied.
	.1..	....	ESTAR indicator.
	...1	....	ESTAE/ESTAI/ESTAR parameters.
	....	.1..	Allow asynchronous exit scheduling.
	....	..10	Do not purge I/O operations.
	....	..01	Purge I/O operations with the halt option.
	....	..00	Purge I/O operations with the quiesce option.
	..0.	0...	Reserved and set to zero.
1-3	Address of the user exit routine.		
4-7	Address of the user parameter list.		
8-11	TCB address if an ESTAI request, otherwise zero.		
12	Flag byte:		
	..1..	....	Request for termination processing.
	..1.	....	Request for error recording.
	....	.1..	Request branch entry to SVC 60.
13-15	Reserved.		

SVC 61 (0A3D) IKJEGS6A macro - is type 3, gets LOCAL lock, calls module IGC0006A. PLM is *OS/VS2 TSO Command Processor Logic: Volume III (TEST)*. GTF data is:

R15 and R0 - No applicable information.

R1 Contains:

Zeros if the routine is being entered from the overlay supervisor.

Address of the DCB used to fetch the module if the routine is being entered from the contents supervisor.

SVC 62 (0A3E) DETACH macro - is type 2, gets LOCAL lock, calls module IEAVEED0 (E.P. = IGC062). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the fullword containing the address of the subtask TCB to be detached. If bit 0 = 1, STAE=YES was specified. This affects the abend code with which an incomplete subtask is abended; If STAE=YES the code is 33E, otherwise it is 13E.

DETACH TCB hhhhhhhh Address of the subtask TCB to be detached.

Note: If R1 contains zeros, the DETACH TCB field is meaningless, and the issuer of SVC 62 will be abended with code 23E.

SVC 63 (0A3F) CHKPT macro - is type 4, gets LOCAL and CMS locks, calls module IHJACP00 (E.P. = IGC0006C). PLM is *OS/VS2 Checkpoint Restart Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Contents:

a. Address of the parameter list.

b. Zero if for a CANCEL request.

PLIST

8 bytes long; format is:

Bytes		
0	00	Check the ID address provided in the second parameter of CHKPT macro instruction.
	80	No check ID address is provided.
1-3	Address of the checkpoint DCB.	
4	00	Check ID address is provided.
	01 to 10	Check ID length is provided via the third parameter of the CHKPT macro instruction.
	FF	'S' specified as the third parameter of the CHKPT macro instruction; the system-generated check ID is to be placed at the address specified in bytes 5-7.
5-7	Address for storing the system-generated check ID or the address of the user provided check ID.	

SVC 64 (0A40) RDJFCB macro - is type 3, gets LOCAL lock, calls module IGC0006D. PLM is *OS/VS2 OPEN/CLOSE/EOV Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list.

PLIST four to 40 bytes of the RDJFCB parameter list, which has a maximum of 1020 bytes. The list is a series of 4-byte entries, each containing a DCB address. The high-order byte has bit 0 set to one to indicate the last entry.

SVC 65 (0A41) Reserved.

SVC 66 (0A42) BTAMTEST macro - is type 4, gets no lock, calls module IGC0006F. PLM is *OS/VS2 BTAM Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the IOB when the SVC was issued.

IOBERINF Address of the RFT message, inserted by the channel end appendage (IGG019MB).

IOBERNIF+4 Address of the parameter list, inserted by the terminal test control (IGG019MR).

SVC 67 (0A43) Reserved.

SVC 68 (0A44) SYNADAF macro - is type 4, gets no lock, calls module IGC0006H. PLM is *OS/VS2 SAM Logic*. GTF data is:

R15 High-order position is a flag byte; three low-order bytes of user data or the address of the entry point to the SYNAD routine. Flag byte codes are:

Code

X'00' EXCP request.
X'01' BPAM request.
X'02' BSAM request.
X'03' QSAM request.
X'04' BDAM request.
X'05' BISAM request.
X'06' QISAM request.
X'07' BTAM request.
X'09' GAM request.

R0 Three low order bytes:

Address of the DECB for BSAM, BPAM, BDAM, or BISAM.

Address of the IOB for QISAM or EXCP.

Address of the status indicators for QSAM.

High order byte:

QSAM Offset of the first CCW in the IOB.

Not applicable for other access methods.

R1 High-order byte has a flag byte; three low-order bytes have the address of the DCB. Flag byte bit settings are:

Bits-reserved for EXCP, BISAM, QISAM, BDAM, BPAM, BSAM, and QSAM as follows:

1...	....	Error caused by an input operation.
.1..	....	Error caused by an output operation.
..1.	....	Error caused by a BSP, CNTRL, or POINT.
...1	....	Record has been successfully read.
....	1...	Invalid request.
....	.1..	PT conversion - invalid character.
....	..1.	BDAM only - hardware error.
....	...1	BDAM only - no space for the record.

SVC 68 (0A44) SYNADRLS macro - is type 4, gets no lock, calls module IGC0006H. PLM is *OS/VS2 SAM Logic*. GTF data is:

R0 and R1 - No applicable data.

R15 High-order byte has X'FF' and three low-order bytes are user data.

X'FF' Indicates that the SVC routine is being entered from the SYNADRLS macro instruction.

R13 Save area address.

SVC 69 (0A45) BSP macro - is type 3, gets no lock, calls module IGC0006I. PLM is *OS/VS2 SAM Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the DCB.

SVC 70 (0A46) GSERV macro - is type 2, gets LOCAL lock, calls module IGC070. PLM is *OS/VS2 Graphics Access Method Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Contents:

Bytes

0 Mask indicating which bits in the graphic control byte (GCB) should be reset.

1-3 Address of a fullword field that identifies the DCB related to the GCB in which bits are to be reset.

PLIST 4 bytes displays the fullword pointed to by register 1. Byte 0 is a unit index factor used to locate the UCB address in the DEB associated with the DCB. (The GCB to be reset is in the UCB).

SVC 71 (0A47) ASGNBFR/BUFINQ/RLSEBFR macro - is type 3, gets LOCAL lock, calls module IGC0007A. PLM is *OS/VS2 Graphics Access Method Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list.

DDNAME cccccccc name of the DD statement associated with the DCB specified by the macro instruction.

PLIST parameter list up to 12 bytes long pointed to by R1. The content varies according to the macro instruction calling the SVC; contents are:

Entry from ASGNBFR:

Bytes

0 Request byte; settings are:
04 Indicates ASGNBFR.

1-3 DCB address.

4-7 Address of the halfword field containing the number of bytes of buffer to be assigned.

Entry from RLSEBFR:

Bytes

0 Request byte; settings are:
08 RLSEBFR.
0C RLSEBFR ALL.

1-3 DCB address.

4-7 Address of the halfword field containing the number of bytes of buffer to be released.

Entry from BUFINQ:

Bytes

0 Request byte; settings are:
10 Indicates BUFINQ.

1-3 DCB address.

4-7 Address of the table of buffer addresses (must be on a fullword boundary).

8-11 The number of bytes specified to be available for the table of buffer addresses.

SVC 72 (0A48) No macro - is type 4, gets LOCAL and CMS lock, calls module IEAVVCTR. PLM is *OS/VS2 System Logic LIbrary*. APF protected. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list that contains:

Offset

X'00' Address of the parameter list +8

X'04' Address of the DCB.

X'08' Module name for the XCTL.

X'10' Code for the OPEN/CLOSE (1 byte); address of the UCM entry (3 bytes).

X'14' Address of the UCM.

X'18' Address to return to.

SVC 73 (0A49) SPAR macro - is type 3, gets LOCAL lock, calls module IGC0007C. PLM is *OS/VS2 Graphics Access Method Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list.
PLIST up to 40 bytes. It is a series of 4-byte entries. First entry has format:

Bytes

0 Priority specified for the attention routine by the SPAR macro instruction.

1 Reserved.

2-3 Number of words in the parameter list.

Each additional entry contains the GACB address specified by the SPAR macro instruction.

SVC 74 (0A4A) DAR macro - is type 3, gets LOCAL lock, calls module IGC0007D. PLM is *OS/VS2 Graphics Access Method Logic*. GTF data is:

R15 and R0 - No applicable information.

R1 Address of the parameter list.

PLIST up to 40 bytes. It is a series of 4-byte entries. First entry has the format:

Bytes

0-1 Reserved.

2-3 Number of words in the parameter list.

Each additional entry contains the GACB address specified by the DAR macro instruction.

SVC 75 (0A4B) DQUEUE macro - is type 3, gets LOCAL lock, calls module IGC0007E. PLM is *OS/VS2 Graphics Access Method Logic*. GTF data is:

R15 No applicable data.

R0 Address of next the IQE on the IRB active list for the attention routine when ATTNINQ has specified the clear mode; otherwise, contains zeros.

R1 content:

Bytes

0 Unit index to identify a particular 2260 display station; or 00 for a 2250 station.

1-3 GACB address.

IQE When ATTNINQ specifies clear mode this field contains the first 3 words of the IQE pointed to by R0:

Bytes

0-3 Address of the next IQE in the chain, or zeros.

4-7 No applicable data.

8-11 Address of the IRB associated with the IQE. N/A will appear in this field whenever the ATTNINQ macro instruction did not specify the clear mode.

SVC 76 (0A4C) IFBSTAT macro - is type 3, gets no lock, calls module IFBSVC76 (E.P. = IGC0007F). PLM is *OS/VS2 SYS1.LOGREC Error Recording Logic*. APF protected. GTF data is:

R15 No applicable data.

R0 If positive, contains the function indicator in byte 3:

00 Indicates that the EOD recording is requested.

04 Indicates that the EREP entry to record statistical information in SYS1.LOGREC is requested.

08 Indicates that an IPL recording is requested.

0C Indicates entry to update date and time values in the SYS1.LOGREC time-stamp record.

If negative (complemented), contains the length in bytes of a record to be placed in the SYS1.LOGREC data set.

R1 If R0 is positive, R1 contains no applicable data. If R0 is negative, R1 contains the address of the record to be written.

SVC 77 (0A4D) Reserved.

SVC 78 (0A4E) LSPACE macro - is type 3, gets LOCAL lock, calls module IGC0007H. PLM is *OS/VS2 DADSM Logic*. GTF data is:

R15 No applicable data.

R0 Address of the associated UCB.

R1 SMF indicator and/or the message buffer address as follows:

Bytes

0 SMF indicator (caller must be in protect key 0 or authorized to specify either SMF indicator).

X'80' - Build SMF record type 19.

X'40' - LSPACE should test if the SMF volume information is requested before building the SMF record type 19.

1-3 zero or the address of a 30-byte message buffer.

CUU cccc unit address in channel-unit format.

SVC 79 (0A4F) STATUS macro - is type 1, gets LOCAL, CMS, CMSEQDQ, SALLOC, and DISP locks, plus the local and global intersect, calls module IEAVSETS (E.P. = IGC079). PLM is *OS/VS2 System Logic LIbrary*. GTF data is:

The two low-order bytes of register 0 contain a STATUS function code. Depending on the code, registers 15 and 1 contain other information as shown.

Register 0		Function	Register 1	Register 15
0-1	2-3			
0000	0001	MCSTEP	N/A	N/A
MASK	0003	NDSTEP	N/A	ASID(XM status only for reset/start)
MASK	0004	NDSYS	N/A	N/A
N/A	0005	NDTCB	†TCB	ASID
0000	0006	STOP	0 or †TCB	N/A
0000	0007	START	0 or †TCB	N/A
ASID	0008	SDSTEP	N/A	MASK
N/A	0009	SDSYS	N/A	N/A
ASID	000A	SDTCB	†TCB	MASK
ASID	000B	SDETCB	†TCB	MASK
MASK	000C	NDETCB	†TCB	ASID(XM status only for reset or start)
0000	000D	SRBS	N/A	ASID(XM status only for reset or start)
0000	000E	SYNCH	N/A	N/A
0000	000F	Caller, SD	†TCB	MASK
0000	000F	Caller, SD	N/A	N/A (for MVS/System Extensions)
0000	0010	Caller, ND	†TCB	MASK
0000	0010	Caller, ND	N/A	N/A(for MVS/System Extensions)
0000	0011	SRBs only	N/A	ASID(XM status only for reset/start - for MVS/System Extensions)

Note: The sign bit of register one indicates:
0 = set(stop).
1 = reset(start).
not applicable to codes 6, 7, 14, 15, 16.

SVC 80 (0A50) Reserved.

SVC 81 (0A51) SETPRT macro - is type 4, gets no lock, calls module IGC0008A. PLM is *OS/VS2 SAM Logic*. GTF data is:

R15 and R0 - No applicable data.
R1 Address of the parameter list.
DDNAME cccccccc Name of the DD statement associated with the data set being printed.
PLIST parameter list of four or twelve words being passed to SVC 81; The 12-word parameter list is collected if the extended list bit is on in the parameter list.

Bytes
0-3 address of the DCB.
4-7 EBCDIC character-set image ID.
8 LOAD MODE indicator; bit settings are:

Bits		
.0..	....	No fold.
.1..	....	Fold.
x.xx	xxxx	Reserved.

9 Verification indicator; bit settings are:

Bits		
...1	....	Display the image on the printer for verification.
...0	....	Do not display the image on the printer for verification.
xxx.	xxxx	Reserved.

10 Data check indicator; bit settings are:

Bits		
1...	....	Block.
.1..	....	Unblock.
00..	....	As the DCB specifies.
....	1...	Unfold the UCS 3211.
....	.1..	Fold the UCS 3211.
..xx	...x.	Reserved.
....	...1	Extended parameter list.

11-14 EBCDIC FCB image ID.

15 FCB parameter options; bit settings are:

Bits		
1...	....	Verify the FCB.
....	...1	Align.
.xxx	xxx.	Reserved.

16 SPPFLAG1 Flag indicators; bit settings are:

Bits		
1...	....	BURST=Y, thread through the burster-trimmer-stacker.
.1..	....	REXMIT=Y, retransmit.
..1.	....	INIT=Y initialize the printer.
....	1...	Bypass the "load forms overlay" message (JES2 and JES3 only).
....	.1..	Bypass the stacker setup message (JES2 and JES3 only).
...x	..xx	Reserved.

17 SPPFLAG2 Flag indicators; bit settings are:

Bits		
1...	....	MODIFY is specified as an address.
0...	....	MODIFY is not specified or is specified as a name.
.1..	....	First character arrangement table is specified as an address.
.0..	....	First character arrangement table is specified as a name or is not specified.
..1.	....	Second character arrangement table is specified as an address.
..0.	....	Second character arrangement table is specified as a name or is not specified.
...1	....	Third character arrangement table is specified as an address.
...0	....	Third character arrangement table is specified as a name or is not specified.
....	1...	Fourth character arrangement table is specified as an address.
....	0...	Fourth character arrangement table is specified as a name or is not specified.
....	.1..	FCB is specified as an address (3800 only).
....	.0..	FCB is specified as a name or is not specified.
....	..xx	Reserved.

18 Number of copies to be printed on this transmission.

19 Starting copy number.

20 Reserved.

22 Number of copies to be forms flashed on this transmission.

23 Table reference character for copy modification.

24 The last 4 bytes of a module name or a pointer to the copy modification control record.

28 The 4 character name of a forms overlay frame.

32 The last 4 bytes of a member name or a pointer to the first character arrangement table module.

36 The last 4 bytes of a member name or a pointer to the second Character arrangement table module.

40 The last 4 bytes of a member name or a pointer to the third character arrangement table module.

44 The last 4 bytes of a member name or a pointer to the fourth character arrangement table module.

SVC 82 (0A52) DASDR macro - is type 4, gets no lock, calls module IGC0008B. PLM is *OS/VS2 Utilities Logic*. APF protected. GTF data is:

R15 and R0 - No applicable data.
R1 Address of the parameter list.
VOLSER cccc volume serial number.
DA-ADDR hhhhhhhh hhhhhhhh field displayed depends on the options in effect for the SVC routine.

Option	Direct access address content.
Analyze or format	6-byte track address.
Post UCB	8-byte track address.
Address of alternate track CCHH	N/A.
Unlabeled volume	8-byte track address.
New volume	N/A.

PLIST parameter list up to 16 bytes long pointed to by R1. First 4 bytes include a flag byte defining the function to be performed, and a 3-byte UCB address. The fourth, eighth, and twelfth bytes, when present, will contain a flag indicating the last element (4-bytes) in the list. Last flag bit settings are:

Bytes
0 Function byte as follows:

8F	New volume.
1F	Address of the alternate track CCHH.
00	ANALYSE or FORMAT.
08	POST UCB.
88	Unlabeled volume.
F8	Delete the DEB analyze of label.

1-3 Address of the UCB.
The contents of the remaining bytes are determined by the function specified in byte 0.
(Function 8F)
4 80 Flag byte in the last element.
5-7 Address of DCB.
(Function 1F)
4 80 Flag byte in the last element.
5-7 Address of the alternate track CCHH.
(Function 00)
4-7 Address of the alternate track CCHH.
8 80 Flag byte in the last element.
9-11 Address of alternate track information.
(Function 08)
4-7 Address of the serial number.
8 80 Flag byte in the last element.
9-11 Address of the VTOC.
(Function 88)
4-7 Address of the serial number.
8-11 Address of the VTOC.
12 80 Flag byte in the last element.
13-15 Address of the DEB.
(Function F8)
4-7 Address of the serial number.
8-11 Address of the VTOC.
12 80 Flag byte in the last element.
13-15 Address of the DEB.

SVC 83 (0A53) SMFWTM macro - is type 3, gets no lock, calls module IEEMB830. PLM is *OS/VS2 System Logic Library*. APF protected. GTF data is:

R15 and R0 - No applicable data.
R1 The address of an SMF record that is to be written to an SMF data set.

SVC 84 (0A54) GRAPHICS macro - is type 1, gets LOCAL lock, calls module IGC084 (E.P. = IGC00084). PLM is *OS/VS2 Graphics Access Method Logic*. GTF data is:

R15 No applicable data.

R0 High-order two bytes have the buffer restart address stored in the UCB; two low-order bytes contain the address of the UCB.

R1 Zeros.

SVC 85 (0A55) DDRSWAP macro - is type 3, gets LOCAL lock, calls module IGC0008E. PLM is *OS/VS2 System Logic Library*. APF protected. GTF data is:

R15, R0 and R1 - No applicable data.

SVC 86 (0A56) ATLAS macro - is type 4, gets no lock, calls module IGC0008F. PLM is *OS/VS2 Utilities Logic*. APF protected. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list.
PLIST parameter list is 8 bytes long; format is:

Bytes			
0	Flag byte; bit settings are:		
	1...	....	User's channel program is not reexecutable
	.xxx	xxxx	Reserved.
1-3	Address of the IOB.		
4	Flag byte; bit settings are:		
	1...	....	IEHATLAS is the calling program.
	.1..	....	Partial count (CCHH only) has been passed by the calling program.
	..1.	....	Special write CCW is required for a track overflow record.
	...1	....	Special write CCW is not required.
	....	xxxx	Reserved.
5	Address of the 5-byte track address of the complete (CCHHR) or partial count (CCHH) field passed by the calling program. If entry is from the IEHATLAS program (byte 4, bit 0 in parameter list), this address points to the CCHH part of the count field.		
CCHHR cccc	Track address		

SVC 87 (0A57) DOM macro - is type 3, gets LOCAL and CMS locks, calls module IEAVXDOM (E.P. = IGC0005G). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Values are as follows:
- NEG List of ID numbers and REPLY=YES not specified
- 00 One ID number and REPLY=YES not specified
- 04 One ID number and REPLY=YES specified
- 0C List of ID numbers and REPLY=YES specified

R1 List of ID numbers or 24-bit right-adjusted ID number of the message to be deleted. Contents are determined by R0.
PLIST up to 40 bytes long. It is a series of 4-byte entries. Each entry is a message ID word; the last entry is identified by 1 in the first bit of the high-order byte.

SVC 88 (0A58) MOD88 macro - is type 3, gets no lock, calls module IGC0008H (E.P. = IGC00088). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.

R0 User defined job entry code.

R1 Address of the DCB associated with the current task when the SVC was issued.

DEB	hhhhhhhh	address of the DEB taken from the DCB pointed to by R1.
DSSTAT	hh	data set status flags field taken from the DEB.
DEVMOD	hh	device modifier field taken from the DEB.

SVC 89 (0A59) Reserved.

SVC 90 (0A5A) Reserved.

SVC 91 (0A5B) VOLSTAT macro - is type 3, gets no lock, calls module IGC0009A. PLM is *OS/VS2 SYS1.LOGREC Error Recording Logic*. GTF data is:

R15 No applicable information.
R0 If negative, contains the address of the UCB.
If positive, contains address the of the DCB.
R1 Contents:
The SVC was issued by CLOSE.
X'32' The SVC was issued by DDR.
X'33' The SVC was issued by EOD.
X'63' The SVC was issued by EOV.
Any other, the SVC was issued by UNALLOCATION.

SVC 92 (0A5C) TCBEXCP macro - is type 1, gets LOCAL lock, calls module IECVEXCP (E.P. = IGC092). PLM is *OS/VS2 I/O Supervisor Logic*. GTF data is:

R15 No applicable data.
R0 Address of the TCB to which the I/O operation is related.
R1 Address of the IOB associated with this request. (If the high order bit is zero, SVC 92 is functionally equivalent to SVC 0; if the high order bit is one, SVC 92 is functionally equivalent to SVC 114.)

DDNAME	cccccccc	Name of the associated DD statement.
DCB	xxxxxxxx	Address of the DCB associated with this I/O request.
DEB	xxxxxxxx	Address of the DEB associated with this I/O request.

SVC 93 (0A5D) TGET macro - is type 3, gets LOCAL and CMS locks, calls module IGC0009C. PLMs are *OS/VS2 TCAM Logic* and *OS/VS2 MVS VTIOC and TCAS Logic*. (See also following TPUT description.) GTF data is:

R15 No applicable data.
R0 Two high-order bytes are reserved. Two low-order bytes contain the buffer size in bytes.
R1 High-order byte is a flag byte; three low-order bytes contain the address of the buffer that is to receive data from the input line. Flag byte bit settings are:

Bits

1...	....	TGET specified.
0...	....	TPUT specified.
.1..	....	Reserved.
..1.	....	Reserved for TPUT.
...1	....	NOWAIT specified; control should be returned to the program that issued the TGET whether or not an input line is available from the terminal. If no input line is obtained, a return code of 4 will be found in register 15.
...0	....	WAIT specified; control will not be returned to the program that issued the TGET until an input line has been put into the program's buffer. If an input line is not available from the terminal, the issuing program will be put into a wait state until a line does become available and is placed in program's buffer.
....	1...	Reserved for TPUT.
....	.1..	Reserved for TPUT.
....	..10	Reserved for TPUT.
....	..01	ASIS specified; normal or minimal editing will be performed.
....	..00	EDIT specified; further editing will be performed in addition to the normal (ASIS) editing.

SVC 93 (0A5D) TPUT macro - is type 3, gets LOCAL and CMS locks, calls Module IGC0009C. PLM is *OS/VS2 TCAM Logic*. (See also preceding TGET description.) GTF data is:

R15 Pointer to the USERID if specified. (See flag byte, bit 1 in reg 1).

R0 Two high-order bytes contain the address space identifier (ASID) of the destination terminal.

Two low-order bytes contain the size of the input buffer in bytes.

R1 The high-order is a flag byte; low order bytes have the address of the buffer to hold line of output. Flag byte bit settings are:

Bits		
1...	....	TGET specified.
0...	....	TPUT specified.
.1..	....	USERID specified by register 15.
..1.	....	LOWP specified; the terminal will not receive any inter-terminal messages from non-supervisory routines if TSBITOFF is on, even if a key-zero task is sending messages. Can only be specified on TPUT with ASID or USERID.
..0.	....	HIGHP specified; the terminal will receive inter-terminal messages if TSBITOFF is on. Can only be specified with ASID or USERID.
...1	....	NOWAIT specified; control should be returned to program that issed TPUT, whether or not TIOC buffers are available for the output line. If buffers are not available, a return code of 4 will be placed in register 15
...0	....	WAIT specified; control will not be returned to the program that issued the TPUT until an output line has been placed in a TIOC output buffer. If no buffers are available, the issuing program will be put into a wait state until buffers do become available and the output line is placed in them.
....	1...	HOLD specified; the program that issued the TPUT cannot continue its processing until this output line has been either written to the terminal or deleted.
....	0...	NOHOLD specified; control should be returned to the program that issued the TPUT as soon as the output line has been placed on the output queue.
....	.1..	BREAKIN specified; output has precedence over input. If the user at the terminal is transmitting, he is interrupted, and this output line is sent. Any data that was received before the interruption is kept and displayed at the terminal following this output line.
....	.0..	NOBREAK specified; input has precedence over output. The output message will be placed on the output queue to be printed at some future time when the user is not entering a line.
....	..10	CONTROL specified; the line is composed of terminal control characters and will not print or move the carriage on the terminal.
....	..01	ASIS specified; normal minimal editing will be performed.
....	..00	EDIT specified; further editing will be performed in addition to the normal ASIS editing.
....	..11	FULSCR specified; no editing is performed.

SVC 94 (0A5E) STCC macro - is type 3, gets LOCAL and CMS locks, calls module IGC0009D. PLMs are *OS/VS2 TCAM Logic* and *OS/VS2 MVS VTIOC and TCAS Logic*. GTF data is:

SVC 94 is called by the following macro instructions; TCLEARQ, STBREAK, STCOM, STTIMEOU, STCC, STATTN, STAUTOLN, STSIZE, GTSIZE, STAUTOCP, STAUTSRM, RTAUTSRM, STCLEAR, STTRAN, STFSMODE, STLINENO, TCABEND and TSEND. GTF data is:

Entry from TCLEARQ:

R15 No applicable data.
R0 Contents:
Bytes
0 01 Entry code.
1-3 0 Reserved.
R1 Contents:
Bytes
0 80 INPUT specified.
00 OUTPUT specified.
1-3 0 Reserved.

Entry from STBREAK:

R15 No applicable data.
R0 Contents:
Bytes
0 04 Entry code.
1-3 0 Reserved.
R1 Contents:
Bytes
0 80 YES specified.
00 NO specified.
1-3 0 Reserved.

Entry from STCOM:

R15 No applicable data.
R0 Contents:
Bytes
0 05 Entry code.
1-3 0 Reserved.
R1 Contents:
Bytes
0 80 YES specified.
00 NO specified.
1-3 0 Reserved.

Entry from STTIMEOU:

R15 No applicable data.
R0 Contents:
Bytes
0 06 Entry code.
1-3 0 Reserved.
R1 Contents:
Bytes
0 80 YES specified.
0 NO specified.
1-3 0 Reserved.

Entry from STCC:

R15 No applicable data.
R0 Contents:
Bytes
0 07 Entry code.
1-3 0 Reserved.
R1 Contents:
Bytes
0 Flag byte; bit settings are:

Bits		
1...	....	First operand specified.
.1..	....	ATTN specified.
..1.	....	LD specified.
...1	....	CD specified.
0000	0000	No operands specified; retain the previously-used characters.

1 0 Reserved.

2 hh Hexadecimal representation of any EBCDIC character on the terminal keyboard except new line (NL) and carriage return (CR) control characters.

c Character representation of any EBCDIC character on the terminal keyboard.

hh Character - delete the control character; the hexadecimal representation of any EBCDIC character on the terminal keyboard except new line (NL) and carriage return (CR) characters.

c Character representation of any EBCDIC character on the terminal keyboard.

Entry from STATTN:

R15 No applicable data.

R0 Contents:

Bytes

0 8 Entry code.

1 00 Reserved.

2 hh Line byte; number of consecutive lines of output that can be directed to the terminal before the keyboard will unlock.

00 Output line counting is not used.

3 hh Tens byte; tens of seconds that can elapse before the keyboard will unlock.

00 Locked keyboard timing is not used.

R1 Contents:

Bytes

0 Flag byte

1... .... LINES specified.

.1.. .... TENS specified.

..1. .... Input address specified.

0000 0000 No operands specified; results in a NOP instruction.

1-3 Character string address; if zeros, no character string was specified.

Entry from STAUTOLN:

R15 No applicable data.

R0 Contents:

Bytes

0 09 Entry code.

1-3 Address of a fullword containing the number to be assigned to the first line of terminal input.

R1 Contents:

Bytes

0 00 Reserved.

1-3 Address of a fullword containing the increment value used in assigning line numbers.

Entry from STSIZE:

R15 No applicable data.

R0 Contents:

Bytes

0 0A Entry code.

1-2 Reserved should be zeros.

3 Lines byte; number of lines (depth) that can appear on the screen.

R1 Contents:

Bytes

0-2 Reserved; should be zeros.

3 Size byte; the logical line size (width), in characters, for the terminal.

Entry from GTSIZE, STAUTOCP, STAUTSRM, RTAUTSRM:

R15 No applicable data.

R0 Contents:

Bytes

0 Entry codes are:

OB GTSIZE.

OC STAUTOCP.

OD STAOUTPT.

OE RTAUTSRM.

1-3 000000 Reserved.

R1 No applicable data; should be zero.

Entry from STCLEAR:
R15 No applicable data.
R0 Contents:
Bytes
0 10 Entry code.
1-3 Reserved; should be zeros.
R1 Contents:
Bytes
0 Reserved; should be zeros.
1-3 Erasure character string address.
Entry from STTRAN
R15 No applicable data.
R0 Contents:
Bytes:
0 0F Entry code.
1 Flag byte
1... .... NOTRAN specified.
.1.. .... NOCHAR specified.
..1. .... TCHAR and SCHAR specified.
2 Terminal character to be translated in the system.
3 System character to be translated at the terminal.
R1 Address of the parameter list containing the address and the name of the user table.
Entry from STFSMODE:
R15 No applicable data.
R0 Contents:
Bytes
0 12 Entry code.
1-3 0 Reserved.
R1 Contents:
Bytes
0 80 ON specified.
00 OFF specified.
1-3 0 Reserved.
Entry from STLINENO:
R15 No applicable data.
R0 Contents:
Bytes:
0 13 Entry code.
1-3 0 Reserved.
R1 Contents:
Bytes
0 80 ON specified.
00 OFF specified.
1-2 0 Reserved.
3 Line number byte; the screen line number that specifies where the next non-full-screen message should appear.
Entry from TCABEND:
R15 No applicable data.
R0 Contents as follows;
Bytes:
0 00 Entry code.
1-3 0 Reserved.
R1 No applicable data.
Entry from TSEND:
R15 No applicable data.
R0 Contents as follows;
Bytes:
0 02 entry code.
1-3 0 Reserved.
R1 No applicable data.

SVC 95 (0A5F) SYSEVENT macro - is type 1, gets LOCAL, SRM, SALLOC, and DISP locks (dependent on SYSEVENT code in register 0), calls module IRARMINT (E.P. = IGC095). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 For some SYSEVENT codes, contains the return code value.
R0 Contents:
Bytes
0-1 Zeros, address space identifier (ASID), or not applicable.
2 Contents variable; see the SYSEVENT summary (immediately following the SVC summary).
3 SYSEVENT code; see the SYSEVENT summary.
R1 Contents variable; see the SYSEVENT summary.

SVC 96 (0A60) STAX macro - is type 3, gets LOCAL lock, calls module IEAVAX00 (E.P. = IGC0009F). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.
R1 Address of the parameter list.
PLIST 20 bytes long; format is:
Bytes
0-3 Address of the user program to get control at the time of the attention interruption.
4-5 Size of the input buffer (max 4095).
6-7 Size of the output buffer (max 4095).
8-11 Address of the output buffer.
12-15 Address of the input buffer.
16 STAX option flag byte; bit settings are:
Bits
1... .... Reserved.
.0.. .... REPLACE = YES.
.1.. .... REPLACE = NO.
..1. .... DEFER = YES.
...1 .... DEFER = NO.
.... 1111 Reserved.
17-19 Address of user parameters for the user program.

SVC 97 (0A61) IKJEGS9G macro - is type 3, gets LOCAL lock, calls module IGC0009G. PLM is *OS/VS2 TSO Command Processor Logic: Volume III (TEST)*. GTF data is:

Used by any module of the tested program; as a breakpoint handler, the TCBTCP bit is X'1' in the current TCB.
R15, R0, and R1 - No applicable data.
Used by any module of the TSO test command processor as a subroutine of the TSO TEST; the current TCBTCP bit is X'0' and registers contain:
R15 No applicable data.
R0 Contents:
Bytes
0 Entry codes are:
40 Set the TCBTCP bit to '1'.
20 Set the TCBTCP bit to '0'.
10 Alter the TCBTRN field.
08 Alter the second word of RBOPSW field.
04 Alter the specific register in SVC 97's SVRB register save area.
04 Alter all registers in SVC 97's SVRB register save area.
02 Alter the floating-point register in the TCB save area.
01 Set the RB wait count to 0.
1-3 Address of the target TCB, PRB, or IRB.

R1 Contents:

Entry Code	Bytes	
40	0-3	Not applicable.
20	0-3	Not applicable.
10	0	Not applicable.
	1-3	TCBTRN value.
08	0	Instruction length, completion code and program mask.
	1-3	Address of the value for the second word of the RBOPSW field.
04	0	Register number.
	1-3	Address of the new value.
04	0	X'FF'
	1-3	Address of the 64-byte value.
02	0	Floating-point register number.
	1-3	Address of the new value for the register.
01	0-3	Not applicable.

SVC 98 (0A62) PROTECT macro - is type 4, gets LOCAL and CMS locks, calls module IGC0009H. PLM is *OS/VS2 DADSM Logic*. GTF data is:

R15 and R0 - No applicable data.
R1 Address of the parameter list.
PLIST first 4-bytes of the parameter list; format is:

Byte		
0	Entry code.	
	01	ADD function.
	02	REPLACE function.
	03	DELETE function.
	04	LIST function.
	05	TTR function.
1-3	Depends on the function indicated in byte 0:	
	000000	Add function.
	000000	Replace function.
	000000	Delete function.
	hhhhhh	LIST function - address of an 80-byte buffer.

SVC 99 (0A63) DYNALLOC macro - is type 3, gets LOCAL and CMS locks, calls module IGC0009I. PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.
R1 Address of the parameter list.
PLIST 20 bytes long.
Refer to *OS/VS2 System Programming Library: Job Management* for details on the PLIST.

SVC 100 (0A64) IKJEFFIB macro - is type 3, gets LOCAL and CMS locks, calls module IKJEFF00 (E.P. = IGC00100). PLM is *OS/VS2 TSO Command Processor Logic: Volume IV*. GTF data is:

R15 No applicable data.
R1 Address of the parameter list.
R0 Number identifying the caller.
PLIST 32 bytes long: format is:

Bytes	
0-3	Address of the TMP parameter list.
4-7	Pointer to the parameter list extension for OPERATOR or PROFILE processors.
8-11	Error return code.
12-19	Failing macro name.
20-21	Caller's ID number.
22-23	Length of the user-defined extension.
24-31	Reserved.

SVC 100 is used by the SUBMIT, OUTPUT, OPERATOR, PROFILE and CANCEL/STATUS processors.

SVC 101 (0A65) QTIP macro - is type 1, gets LOCAL and CMS locks, calls load module IGC0010A. PLM is *OS/VS2 TCAM Logic*. GTF data is:

R15 Contents:

Bytes

0 Zero.

1-3 Depends on the entry code in R0:

Entry Code	R15 Contents (Bytes 1-3)
00	Not applicable.
01	Address of the two word parameter list: Word 1 Address of the USERID. Word 2 Address of the password.
03	Entry address of QTIP0030 within IEDAYAA.
04-0B	Not applicable.
0C	Zero means the queue flush is allowed.
0D	Not applicable.
0E	With save area address in R1, not applicable; without save area address in R1, entry address of QTIP0140 within IEDAYOO.
0F-10	Not applicable.
12-13	Entry address of IEDAYQT1.
15-16	Address of the TSB.
17	Address of the RMPL.
18	(Same as 11-13).
1B	Address of TIOCRPT.
1C	Entry address of QTIP02080 within IEDAYII.
1D	Address of the RMPL when called by IEDAY8.

R0 Contents:

Bytes

0-2 Zeros.

3 Entry codes used:

00	IEDAYAA used; SVC call given.
01	IEDAY88 used; SVC call given.
03	IEDAYAA used; internal branch entry taken.
04	IEDAYHH used; SVC call given.
05-09	IEDAYII used; SVC call given.
0A	IEDAYLL used; SVC call given.
0B-0D	IEDAYOO used; SVC call given.
0E	With save area address in R1, IEDAYOO used, SVC call given; without a save area address in R1, IEDAYOO used, internal branch entry taken.
0F-10	IEDAYOO used; SVC call given.
12-13	IEDAYGP used; branch entry taken.
15-16	IEDAYAA used; SVC call given.
17	IEDAY88 used; SVC call given.
18	IEDAYOO used; internal branch entry taken.
1B	IEDAY88 used; SVC call given.
1C	IEDAYII used; internal branch entry taken.
1D	IEDAYGP used; SVC call given by IEDAY8, internal branch entry taken from IGC0009C.

R1 Contents:

Bytes

0 Zero.

1-3 Zero or address of a 12 word parameter list which is to be restored upon exit from SVC 101.

SVC 101 is used only by TSO and the MCP, and is the interface between these functions for cross-address space communication and data movement.

SVC 102 (0A66) AQCTL macro - is type 3, gets LOCAL and CMS locks, calls module IEDQEB (E.P. = IGC0010B). PLM is *OS/VS2 TCAM Logic*. GTF data is:

R15 and R0 - No applicable data.
R1 Address of the parameter list.
PLIST the parameter list is either one, two or three full words, the last of which has the high order bit on (X'80') to indicate the end. Byte zero of the first word contains the function code.

FUNCTION	BYTE 0	BYTES 1-3
Opctl/network control	X'84'	Pointer to the APCIB.
Move in address space	X'08'	From pointer.
	X'00'	To pointer.
	X'80'	Pointer to the length.
Tpost to ready queue	X'0C'	Pointer to the RCB.
	X'00'	Pointer to the RCB.
	X'80'	Reserved.
	or	
	X'8C'	Pointer to the RCB.
Get/Read	X'90'	Pointer to the ECB.
Put/Write	X'94'	Pointer to the ECB.
Point	X'98'	Pointer to the ECB.
CKREQ	X'9C'	Pointer to the ECB.
Post ECB	X'20'	Pointer to the ECB.
	X'80'	Pointer to the ASID.
Qreset	X'A4'	Pointer to the ECB.

SVC 103 (0A67) XLATE macro - is type 3, gets LOCAL lock, calls module IGC0010C. PLM is *OS/VS2 OPEN/CLOSE/EOV Logic*. GTF data is:

R15 No applicable data.
R0 Length of the field to be translated.
R1 Contents:
Bytes
0 Action bytes; codes are:
00 Translate from EBCDIC to ASCII.
80 Translate from ASCII to EBCDIC.
1-3 Address of the field to be translated.

SVC 104 (0A68) TOPCTL macro - is type 4, gets no lock, calls module IGC0010D. PLM is *OS/VS2 TCAM Logic*. APF protected. GTF data is:

R15 No applicable data.
R0 Indicates the subroutine to be executed:
Bytes
0-2 0000 0001 IGC0010D entry point routine.
0000 0002 GTFIELDA decode routine.
0000 0003 STTNME operator command addressing routine.
0000 0004 IEDQCA02 scan routine.
R1 Address of the operator control work area.

SVC 105 (0A69) IMGLIB macro - is type 3, gets no lock, calls module IGC0010E. PLM is *OS/VS2 SAM Logic*. GTF data is:

R15 and R0 - No applicable data.
R1 Indicates the actions to be taken:
Bytes
0-3 0000 0000 Construct a DCB and DEB for SYS1.IMAGLIB
hhhh hhhh Delete the DCB at this address and also the DEB pointed to by this DCB.

SVC 106 (0A6A) Reserved.

SVC 107 (0A6B) MODESET macro - is type 6, gets no lock, calls module IEAVMODE (E.P. = IGC107). PLM is *OS/VS2 System Logic Library*. APF protected. GTF data is:

R15 and R0 - No applicable data.
R1 Parameter list:

Bytes		
0-2	Reserved (must be zero).	
3	Indicator bits:	
	0000	No action.
	0001	Invalid.
	0010	Place the TCB key in the RBOPSW field of the RB.
	0011	Set the RBOPSW key to zero.
	 0000	No action.
	 0100	Turn on the state bit in RBOPSW field of the RB (problem state).
	 1000	Invalid.
	 1100	Turn off the state bit in RBOPSW field of the RB (supervisor state).

SVC 108 (0A6C) Reserved.

SVC 109 (0A6D) ESR (type 4) SVC - is type 4.

Routes control to types 3 and 4 extended supervisor service routines based on the routing code in register 15.

Code	Macro	Description
00		Reserved.
01		Reserved.
02		Reserved.
03		Reserved.
04		Reserved.
05		Reserved.
06	VSAM MSS Support	SVC - calls module IGX00006. PLM is SY26-3825.
07	MFSTART(RMF)	Authorization required - calls module IGX00007.
08		Reserved.
09		Reserved.
0A		Reserved.
0B		Reserved.
0C		Reserved.
0D	MFSTART (MF/1)	Authorization required - calls module IGX00013. PLM is *OS/VS2 System Logic Library*.
0E	MFDATA (MF/1)	Authorization required - calls module IGX00014. PLM is *OS/VS2 System Logic Library*.
0F		Task termination - calls module IGX00015. PLM is *OS/VS2 System Logic Library*.
10		STAE exit - calls module IGX00016. PLM is *OS/VS2 System Logic Library*.
11		Reserved.
12		Reserved.
13		Reserved.
14		Reserved.
15		Reserved.
16	MFDATA(RMF)	Authorization required - calls module IGX00022.
17		Reserved.
18		Reserved.
19		Reserved.
1A		Reserved.
1B		Reserved.
1C		Reserved.
1E		Reserved.
1F		Reserved.
20		Reserved.
21		Reserved.
22		Internal data collection for RMF.

SVC 110 (0A6E) No macro - is type 4, gets no lock, calls module IEE00110. PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 Zeros indicate the first entry into the SVC routine.
Nonzeros indicate a request to FREE the CSCB pointed by the address in register 1.

R0 No applicable data.

R1 Address of the CSCB.

SVC 111 (0A6F) No macro - is type 2, gets LOCAL and CMS locks, calls module IGC111. PLM is *OS/VS2 JES2 Logic*. GTF data is:

R15 No applicable data.

R0 Contains the function indicator in the low-order byte; refer to the HASPSSSM (SVCHAM) for JES2 or IATDMEB for JES3 program listing for an interpretation.

R1 If positive, contains the address of the RPL. If negative (complemented), contains the address of the ACB.

SVC 112 (0A70) PGRLSE macro - is type 1, gets LOCAL and SALLOC locks, calls module IEAVPSI (E.P. = IGC112). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Starting address of the virtual area to be operated on.

R1 End address of that area plus 1.

SVC 113 (0A71) PGFIX/PGFREE/PGLOAD/PGOUT macro - is type 1, gets LOCAL and SALLOC locks, calls module IEAVPSI (E.P. = IGC113). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 If the high-order bit of register 1 is off, contains the second word of the virtual subarea list (VSL).

R0 If positive, contains the address of the ECB.

R1 If the high-order bit is on, contains the address of the VSL.
If high-order bit is off, contains the first word of the VSL; register 15 will contain the second word.

Virtual Subarea List

Byte 0 Flags:

Bit 0	(1... ....)	This bit indicates that bytes 1-3 are a chain pointer to the next VSL entry to be processed; bytes 4-7 are ignored, but the checking of this bit is subject to the setting of byte 4, bit 1. This feature allows several parameter lists to be chained as a single logical parameter list.
Bit 1	(.1.. ....)	PGFIX is to be performed; reserved, set by macro instruction.
Bit 2	(..1. ....)	PGFREE is to be performed; reserved, set by macro instruction.
Bit 3	(...1 ....)	PGLOAD is to be performed; reserved, set by macro instruction.
Bit 4	(.... 1...)	PGRLSE is to be performed; reserved, set by macro instruction.
Bit 5	(.... .1..)	Reserved.
Bit 6	(.... ..1.)	Long-term PGFIX is to be performed; reserved, set by macro instruction.
Bit 7	(.... ...1)	Reserved.

Start Address:

The virtual address of the origin of the virtual area to be processed.

Byte 4 Flags:

Bit 0	(1... ....)	This flag indicates the last entry of the list. It is set in the last doubleword entry in the list.
Bit 1	(.1.. ....)	When this flag is set, the entry in which it is set is ignored. This bit takes precedence over byte 0, bit 0.
Bit 2	(..1. ....)	Reserved.
Bit 3	(...1 ....)	This flag indicates that a return code of 4 was issued from a page service function other than PGRLSE.
Bit 4	(.... 1...)	Reserved.
Bit 5	(.... .1..)	PGOUT is to be performed; reserved, set by macro instruction.
Bit 6	(.... ..1.)	KEEPREAL option of PGOUT is to be performed; reserved, set by macro instruction.
Bit 7	(.... ...1)	Reserved.

End Address + 1:

The virtual address of the byte immediately following the end of the virtual area.

SVC 114 (0A72) EXCPVR macro - is type 1, gets LOCAL lock, calls module IECVEXCP (E.P. = IGC114). PLM is *OS/VS2 I/O Supervisor Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the IOB associated with this request.

DDNAME	cccccccc	Name of the associated DD statement.
DCB	xxxxxxxx	Address of the DCB associated with this I/O request.
DEB	xxxxxxxx	Address of the DEB associated with this I/O request.

SVC 115 (0A73) Reserved.

SVC 116 (0A74) ESR (type 1) SVC - is type 1, gets LOCAL lock.

Routing code in register 15 determines the type 1 SVC routine to be executed.

Code	Macro	Description
00	IECTRDTI	BTAM 3270 read initial UCB scan - calls module IECTSVC (E.P. = IECTRDIL). PLM is *OS/VS2 BTAM Logic*.
01	IECTATNR	BTAM 3270 attention reset - calls module IECTSVC (E.P. = IECTATRL). PLM is *OS/VS2 BTAM Logic*.
02	CHNGNTRY	BTAM 3270 CHNGNTRY skip - calls module IECTSVC (E.P. = IECTCHSL). PLM is *OS/VS2 BTAM Logic*.
03	IECTCHGA	BTAM 3270 CHNGNTRY activate - calls module IECTSVC (E.P. = IECTCHAL). PLM is *OS/VS2 BTAM Logic*.
04	RESETPL	BTAM 3270 read initial - calls module IECTSVC (E.P. = IECTRSTL). PLM is *OS/VS2 BTAM Logic*.
05		Reserved.
06		Reserved.
07		Reserved.
08	CALLDISP	Dispatcher call - calls module IEAVEEXP (E.P. = IEAVEXSV). PLM is *OS/VS2 System Logic Library*.
09		Reserved.
0A		Reserved.
0B		Reserved.
0C		Reserved.
0D		Reserved.
0E		Reserved.

SVC 117 (0A75) DEBCHK macro - is type 2, gets LOCAL lock, calls load module IFGDEBCK. PLM is *OS/VS2 OPEN/CLOSE/EOV Logic*. GTF data is:

R15 contains the value 2.

R0 Bits 0-7 Access Method Value

X'82'	VTAM
X'84'	TCAMAP
X'81'	SUBSYS
X'80'	ISAM
X'40'	BDAM
X'20'	SAM
X'20'	BPAM
X'10'	TAM
X'08'	GAM
X'04'	TCAM
X'02'	EXCP
X'01'	VSAM
X'00'	None

Bits 8-31 Type Function Code

0	Verify
1	Add
2	Delete
3	Purge

R1 Bits 0-7 X'00'

Bits 8-31 Address of the DCB if the type code is not PURGE.
Address of the DEB if the type code is PURGE.

SVC 118 (0A76) Reserved.

SVC 119 (0A77) TESTAUTH macro - is type 1, gets LOCAL lock, calls module IEAVTEST (E.P. = IGC119). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Applies only if flag bit 7 in register one is zero.
If positive, contains the authorization code.
If negative, does not contain the authorization code.

R1 Bytes have meaning as follows:

BYTE	MEANING	
0	Reserved - must be set to zero.	
1	Flag bits	
	1111	Reserved.
	.... 1...	RBLEVEL=2 (applies only to KEY and/or STATE).
	.... 0...	RBLEVEL=1 (applies only to KEY and/or STATE).
	.... .1..	STATE=YES.
	.... .0..	STATE=NO.
	.... ..1.	KEY=YES.
	.... ..0.	KEY=NO.
	1	FCTN=code (see register 1 byte 3).
	0	no FCTN specified.
2	Reserved - must be set to zero.	
3	FCTN code - applies only if flag bit 7 is '0'.	

SVC 120 (0A78) GETMAIN/FREEMAIN macro - is type 1, gets LOCAL lock, calls module IEAVGM00 (E.P. = IGC120). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 bytes as follows:
- 0 Zero.
- 1 Key of storage to be obtained/freed for subpools 229, 230, 231 or 241 for a branch entry only.
- 2 Subpool number of storage to be obtained/freed.
- 3 Option byte:
  - 00 Conditional GETMAIN.
  - 01 Conditional FREEMAIN.
  - 02 Unconditional GETMAIN.
  - 03 Unconditional FREEMAIN.
  - 04 Page boundary bit (may be OR'ed with either form of GETMAIN.)

R0 The number of bytes of storage to be obtained or freed (Zero for a subpool FREEMAIN).

R1 The address of the area to be freed (Zero for GETMAIN requests).

SVC 121 (0A79) VSAM macro - is type 1, gets LOCAL lock, calls module IGC121 PLM is *OS/VS2 VSAM Logic*. GTF data is:

R15 Contains the pointer to the buffer control block.

R0 Contains the pointer to the place holder entry, used for a record management request.

R1 Contains the pointer to the IOMB (VSAM I/O management control block) that replaces the IOB.

SVC 122 (0A7A) ESR(type2)SVC - is type 2, gets LOCAL lock.

Routes control to type 2 extended supervisor service routines based on a routing code in register 15.

Code	Macro	Description
00		Reserved.
01		Reserved.
02		Reserved.
03		Reserved.
04		Reserved.
05	EVENTS	Calls module IEAVEVT0 (E.P. = IEAVET1). PLM is *OS/VS2 System Logic Library*.

R0 Bytes have meaning as follows:

Bytes	Meaning
0	Flag bits
1...	ENTRIES = n (create request); delete is requested if FC=5.
.111 1111	Reserved.
1	Reserved.
2-3	Number of ENTRIES requested or zero.

R1 Address of the EVENT table if a delete is requested.

06	Reserved.
07	Reserved.
08	Reserved.
09	Reserved.

SVC 123 (0A7B) PURGEDQ macro - is type 2, gets LOCAL and DISP locks (for non-MVS/System Extensions), get only the DISP lock (for MVS/System Extensions), calls module IEAVEPD0 (E.P. = IGC123). APF protected. GTF data is:

R15 No applicable data.
R0 Parameter to be passed to the RMTR if the SRB is purged.
R1 Address of the parameter list.

SVC 124 (0A7C) TPIO macro - is type 1, gets LOCAL locks, calls module ISTAPC22. PLM is *OS/VS2 VTAM Logic*. GTF data is:

R15 No applicable data.
R0 Bytes have meaning as follows:

Byte	Meaning		
0	Flag bits		
	x...	....	Reserved.
	.1..	....	on LCPB indicates.

Bits 2-7

Code	Meaning
00	Specific request.
04	Any request.
08	Open.
0C	TPPOST.
0F	CLOSE ACB.
10	Session control request.

1-3 DEB address.
R1 Work element address.

SVC 125 (0A7D) EVENTS macro - is type 1, gets LOCAL lock, calls module IEAVEVT0 (E.P. = IGC125). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.
R0 Bytes have meaning as follows:

Bytes	Meaning		
0	Flag bits		
	1...	....	WAIT = YES.
	.1..	....	WAIT = NO.
	..1.	....	ECB = address.
	...1	1111	Reserved.
1-3	Address of LAST = ENTRY or, address of ECB if ECB = specified.		

R1 Address of the EVENT table.

SVC 126 (0A7E) MSS Interface - is type 3, gets local and CMS locks, branches to ICB2SIOF, ICBVPI00, ICBVQM00, and ICBVSC00. PLM is *OS/VS2 MSSC Logic*. APF protected. GTF data is:

R15 and R0 No Applicable data.
R1 Contains address of the request block.

SVC 127 (0A7F) Reserved.

SVC 128 (0A80) Reserved.

SVC 129 (0A81) Reserved.

SVC 130 (0A82) RACHECK macro - is type 2, gets no lock, calls module ICHRCK00. GTF data is:

R15 and R0 No applicable data.
R1 Address of the parameter list.

SVC 131 (0A83) RACINIT macro - is type 2, gets no lock, calls module ICHRIN00. APF protected. GTF data is:

R15 and R0 No applicable data.
R1 Address of the parameter list.

SVC 132 (0A84) RACLIST macro - is type 2, gets no lock, calls module ICHRSV00. APF protected. GTF data is:

R15 and R0 No Applicable data.
R1 Address of the parameter list.

SVC 133 (0A85) RACDEF macro - is type 2, gets no lock, calls module ICHRDF00. APF protected. GTF data is:

R15 and R0 No Applicable data.
R0 Address of the parameter list.

SVC 134 (0A86) Reserved.

SVC 135 (0A87) Reserved.

SVC 136 (0A88) Reserved.

SVC 137 (0A89) ESR macro (MVS/System Extensions only) - is type 6, gets no locks, calls module, IGC137. The routing code in register 15 identifies the type 6 SVC routine to be executed.

Code	Macro	Description
00	CALLDISP	Dispatcher call - calls module, IEAVEDS0.
01		Reserved.
02		Reserved.
03		Reserved.

SVC 138 (0A8A) Reserved.

SYSEVENT Summary

This summary describes system events that are indicated by entry to the system resources manager (SRM) via direct branch or SVC 95. For each system event, this summary provides the following information:

- The SYSEVENT code and mnemonic. For example: SYSEVENT 01 hex (TIMEREXP). The SYSEVENT code is taken from the low-order byte of general register 0, and is shown in hexadecimal format.
- The condition represented by the SYSEVENT code.
- Locks held on entry to SRM (acquired by the caller or by the SVC first level interruption handler). *Note:* Some callers will hold other locks, in addition to those shown.
- Register contents after entry to SRM. (This data is displayed in GTF SRM trace records, described in Section 3).
- Information returned by SRM (where applicable).

For additional SYSEVENT code information, refer to *OS/VS2 System Logic Library*.

SYSEVENT 00 HEX (PPMODE): TSEVENT 00, issued when a TSO command or a subcommand of EDIT and TEST is about to be executed. TSEVENT is a carryover from earlier TSO releases.

Locks held on entry to SRM: Local.
Register 0: Bytes 0-3 contain an ASID (or zero, indicating the current ASID).
Register 1: First 4 bytes of the command or subcommand name.
Register 15: Last 4 bytes of the command or subcommand name.

SYSEVENT 01 HEX (TIMEREXP): Issued by TOD clock initialization when the clocks have been initialized, and from TOD clock vary-CPU-online when a good TOD clock and clock comparator are once again available.

Locks held on entry to SRM: Local.
Register 0: Bytes 0-1 contain the ASID (or zero, indicating the current ASID). Byte 3 contains the SYSEVENT code.
Register 1: Byte 3 contains X'01' if the entry is from system TOD clock initialization.

SYSEVENT 02 HEX (TERMWAIT): Issued by TGET and TPUT when the user enters a terminal wait.

Locks held on entry to SRM: Local.
Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.
Register 1: Contains X'00' for an input terminal wait; X'80' for an output terminal wait.

SYSEVENT 03 HEX (NIOWAIT): Issued by WAIT macro processing when a task in an address space enters a long wait.

Locks held on entry to SRM: Local.
Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

SYSEVENT 04 HEX (USERRDY): Issued when an SRB has been scheduled for an address space in which QUIESCE is running, or for a swapped out address space.

Locks held on entry to SRM: Dispatcher.
Register 0: Bytes 0-1 contain an ASID; byte 3 contains the SYSEVENT code.

SYSEVENT 05 HEX (TIME) - applies to MVS/System Extensions: Issued when the SRM timer internal has expired.

Locks held on entry to SRM: Local.
Register 0: Byte 3 contains the SYSEVENT code.

SYSEVENT 06 HEX (MEMCREAT): Issued when an ASID has been associated with a new address space and space has been obtained for an ASCB and OUSB.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID; byte 3 contains the SYSEVENT code.

Register 1: Byte 3 contains: X'01' for a START; X'02' for a LOGON; X'03' for a MOUNT.

SRM returns the following information:

Register 1: Byte 0 contains: X'00' if the address space should be created; X'80' if the address space should not be created because of a resource shortage.

SYSEVENT 07 HEX (MEMDEL): Issued when the storage associated with an ASCB is to be freed and an ASID is to be disassociated from an address space.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID; byte 3 contains the SYSEVENT code.

SRM returns the following information:

Register 1: Byte 3 contains X'00': Memory - delete processing can proceed (applies to MVS/System Extensions).

SYSEVENT 08 HEX (JOBSELCT): Issued when an address space has begun using system services on behalf of a new job, a START or MOUNT command, or a TSO session.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 2 contains the performance group number, byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of an 8-byte area containing the jobname or user ID.

SYSEVENT 09 HEX (JOBTERM): Issued when an address space has completed using system resources on behalf of a job, a START or MOUNT command, or a TSO session.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of an 8-byte area containing the jobname or user ID.

SYSEVENT 0A HEX (INITATT): Issued whenever an initiator attaches a task.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Byte 1 bit 6 contains a '1' indicating that the job step is authorized to issue DONTSWAP/OKSWAP. Bit 6 contains a '0' indicates that the jobstep is not authorized. Bit 8 contains a '1' indicates that DPRTY is specified in the JCL. Bit 8 contains a '0' indicates that DPRTY is defaulted. This byte 1 description applies to MVS/System Extensions. Byte 2 contains the performance group number of the attached task, or 0; byte 3 contains the dispatching priority at which to set this address space.

SYSEVENT 0B HEX (INITDET): Issued whenever an initiator detaches a task.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: byte 3 contains the dispatching priority at which to set this address space.

SYSEVENT 0C HEX (QSCEST): Issued during quiesce processing when the status of all associated tasks has been determined.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Byte 0 contains: X'00' if the address space is not in the long wait; X'80' if all tasks in the address space are in the long wait.

SRM returns the following information:

Register 1: Byte 3 contains: X'00' if the RCT is to continue with quiesce; X'08' if the address space should be restored to its original status.

SYSEVENT 0D HEX (QSCECMP): Issued when the RCT has completed quiesce processing for an address space. For TSO, used to determine if an address space is logically or physically swapped.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Byte 0 contains: X'00' if the address space is not in the long wait; X'80' if the address space is in the long wait.

SRM returns the following information:

Register 1: Byte 0 contains X'00' if a USERRDY was just received, X'80' if no USERRDY was received since the QSCEST; byte 2 contains the SWAP reason code; byte 3 contains X'00' if the RCT is to schedule a swap-out, X'04' if the RCT is to wait on the ASCBQECB for a logically swapped address space, X'08' if the address space is to be restored.

SYSEVENT 0E HEX (TRANSWAP) - applies to MVS/System Extensions: Issued to force a swap out. After the subsequent swap-in, frames are allocated from preferred storage and the address space is marked 'nonswappable'. TRANSWAP prevents programs from allocating frames in reconfigurable storage. This SYSEVENT is also issued by BTAM OPEN and OLTEP.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of the ECB to be posted if a dependency exists on the transition (or zero if no dependency exists). An ECB can be specified only if the request is for the current address space.

SRM returns the following information:

Register 1: byte 3 contains: X'00' if the transition request was honored; X'04' if a transition was done previously.

Post codes:

X'00' - The transition is complete.

X'04' - The address space became nonswappable before it could be swapped out.

SYSEVENT 0F HEX (SWOUTCMP): Issued when all I/O required to swapout an address space has completed.

Locks held on entry to SRM: SALLOC.

Register 0: Bytes 0-1 contain an ASID; byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of a parameter list.

Parameter list: Bytes 0-1 contain the number of pages swapped out; bytes 2-3 contain the number of pages to be swapped in (working set size); bytes 4-5 contain the current number of pages freed at swap-out without I/O; byte 7 contains a flag indicating whether the address space is waiting for unfinished real storage manager service (wait indicated by bit 7 set to one).

SYSEVENT 10 HEX (SWPINST): Issued by RSM to notify of SWAP status.

Locks held on entry to SRM: SALLOC.

Register 0 Bytes 0-1 contain an ASID; byte 3 contains the SYSEVENT code.

Register 1: Byte contains swap-in status code: X'01' if swap-in is starting; X'02' if stage one swap-in has completed.

SYSEVENT 11 HEX (SWINFL): Issued when swap-in processing failed to obtain or initialize the LSQA storage for an address space.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID; byte 3 contains the SYSEVENT code.

SYSEVENT 12 HEX (QSCEFL): Issued when the RCT failed to complete quiesce processing because of an abnormal situation.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

SYSEVENT 13 HEX (RSTORCMP): Issued when the RCT has completed restore processing for an address space.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Byte 0 contains: X'00' if the address space is runnable; X'80' if the address space is in the long wait.

SYSEVENT 14 HEX (ENQHOLD): Issued when a user's execution is delayed because of a request for a resource being held by another user. It is also issued when an authorized program issues RESERVE for a DASD device which is SHARED or SHAREDUP and obtains control of the resource, even if no user's execution is delayed.

Locks held on entry to SRM: Local and global CMS.

Register 0: Bytes 0-1 contain an ASID of the address space holding the resource; byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of the minor QCB for the resource involved.

SYSEVENT 15 HEX (ENQRLSE): Issued when a contention situation or a potential contention situation has disappeared because of the release of a resource by a user for whom an ENQHOLD SYSEVENT had previously been received.

Locks held on entry to SRM: Local and global CMS.

Register 0: Bytes 0-1 contain an ASID of address space holding resource during the contention situation; byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of the minor QCB for the resource involved.

SYSEVENT 16 HEX (RSMCNSTS): Issued at system initialization and when real storage has been configured into or out of the system (because of a VARY storage command or a storage error).

Locks held on entry to SRM: SALLOC.

Register 0: Byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-1 contain the number of pages of functioning real storage; bytes 2-3 contain the number of pages that will be on the available page queue at the time the available page queue below limit SYSEVENT is issued.

SYSEVENT 17 HEX (AVQLOW): Issued when the number of available real storage page frames has fallen below the available page queue low limit.

Locks held on entry to SRM: SALLOC.

Register 0: Byte 3 contains the SYSEVENT code.

Register 1: Byte 3 contains X'01' if the number of real pages on the available page queue has dropped below the AVQ limit; X'02' if the number of real pages on the available page 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 exceeded the allowable value.

SYSEVENT 18 HEX (AVQOK): Issued when enough real storage pages have been freed to alleviate a shortage condition.

Locks held on entry to SRM: SALLOC.

Register 0: Byte 3 contains the SYSEVENT code.

SYSEVENT 19 HEX (SQALOW): Issued when a critical shortage of SQA pages exists.

Locks held on entry to SRM: SALLOC.

Register 0: Byte 3 contains the SYSEVENT code.

Register 1: Byte 3 contains X'01' if the first (most remaining SQA) threshold is passed; X'02' if the second threshold is passed.

SYSEVENT 1A HEX (SQAOK): Issued when an SQA page shortage has been relieved.

Locks held on entry to SRM: SALLOC.

Register 0: Byte 3 contains the SYSEVENT code.

Register 1: Byte 3 contains X'01' if the first (most remaining SQA) threshold is passed; X'02' if the second threshold is passed.

SYSEVENT 1C HEX (DEVALLOC): Issued when a device allocation choice must be made from two or more candidates.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of a list of three fullword addresses; the first pointing to a list of candidate UCB addresses; the second pointing to a list of addresses of UCBs already allocated to the requesting jobstep; the third pointing to a two-word return area.

SRM returns the following information:

Register 15: Byte 3 contains X'00' if successful; X'08' if unsuccessful.

Register 1: Contains the same address as input register 1.

Return area: Word 1 contains the address of the selected candidate list entry. (Note that register 1 points indirectly to this area and to the candidate list.)

SYSEVENT 1D HEX (CONFIGCH): Issued when a VARY command has been issued for a channel or a processor.

Locks held on entry to SRM: None.

Register 0: Byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the pointer to the SMF type 22 (VARY ONLINE or VARY OFFLINE) record which describes the configuration change.

SYSEVENT 1E HEX (VERIFYPG): Issued when an interpreter has received a performance group number that needs verification.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain the ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Byte 3 contains the performance group number.

SRM returns the following information:

Register 1: Byte 2 contains the following for a TSO user:

0 if the performance group number is valid.

1 if the performance group number is invalid.

Byte 2 contains the following for a non-TSO users:

0 if the performance group number is non-zero whether valid or invalid.

1 if the performance group number is zero.

SYSEVENT 1F HEX (RESETPG): Issued when the system operator has entered a RESET command for a particular address space.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain the ASID; byte 3 contains the SYSEVENT code.

Register 1: Byte 3 contains the new performance group number.

SRM returns the following information:

Register 1: Byte 2 contains: X'00' if the reset request was honored; X'04' if the new performance group number is invalid; X'08' if the ASID is not currently assigned.

SYSEVENT 20 HEX (NEWIPS): Issued when the system operator has entered a SET command with the IPS keyword.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain the ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of the WMST (workload manager specification table) describing the new IPS.

SRM returns the following information:

Register 1: Bytes 0-3 contain the address of the WMST to free.

SYSEVENT 21 HEX (ALTCPREC): Issued when some processor has had to be configured out of the system as a result of an error.

Locks held on entry to SRM: None.

Register 0: Bytes 0-1 contain the ASID (or zero, indicating current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the processor address of the failed processor.

SYSEVENT 22 HEX (TGETTPUT): Issued when a TGET or TPUT was satisfied.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain the ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Byte 0 is a flag byte; bit 0 contains 0 if TGET was satisfied, or 1 if TPUT was satisfied; bit 1 contains 0 if all the data in the TSO input message was transferred by the TGET, 1 if all the data in TSO input message was not transferred by the TGET.

SYSEVENT 23 HEX (SYQSCST): Issued when the system start/stop routine has been entered to stop the system.

Locks held on entry to SRM: None.

Register 0: Bytes 0-1 contain the ASID (or zero, indicating current ASID); byte 3 contains the SYSEVENT code.

Note: SYSEVENT 23 HEX is not traced by GTF.

SYSEVENT 24 HEX (SYQSCCMP): Issued when the system start/stop routine is about to restart the system.

Locks held on entry to SRM: None.

Register 0: Bytes 0-1 contain the ASID (or zero, indicating current ASID); byte 3 contains the SYSEVENT code.

Note: SYSEVENT 24 HEX is not traced by GTF.

SYSEVENT 25 HEX (SETDMN): Issued when the operator enters a SETDMN command to change constraint and/or MPL target control values for a domain.

Locks held on entry to SRM: Local

Register 0: Byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain a pointer to the SETD parameter list (mapped by the IRASETD macro).

SRM returns the following information:

Register 15: byte 3 contains: X'00' if successful; X'04' if the domain invalid; X'08' if the minimum exceeds the maximum; X'0C' if AOBJ or DOBJ is invalid; X'10' if the SETDMN input is incompatible with the current IPS.

SYSEVENT 26 HEX (REQSERVC): Issued by SMF during job or session termination, and by the TSO TIME command, to obtain user related-service data.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of a 3 word area where the service data is to be stored.

SRM returns the following information:

Register 15: Byte 3 contains X'04' if data was lost due to the accumulation control block error; otherwise it contains X'00'.

Register 1: Contains the same address as input register 1.

Return area for TSO User: Word 1 contains total service; Word 2 contains a total transaction active time for all transactions; Word 3 contains the last performance group number in bytes 0-1, and the total number of transactions in bytes 2-3.

Return area for non-TSO User: Word 1 contains total service; Word 2 contains the total transaction active time; word 3 contains the last performance group number in bytes 0-1.

SYSEVENT 27 HEX (REQPGDAT): Issued by SMF during step termination to obtain user paging data.

Note: This SYSEVENT is intended for use only by SMF because the related data fields in the OUSB and OUXB are reset to 0.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of an area where the paging data is to be stored.

SRM returns the following information:

Register 15: Byte 3 contains X'04' if data is lost due to the accumulation control block error; otherwise it contains X'00'.

Register 1: Contains the same address as input register 1.

Return area: Word 1 contains a count of non-VIO page-ins; word 2 contains a count of non-VIO page-outs; word 3 contains a count of non-VIO reclaims; word 4 contains a count of VIO page-ins; word 5 contains a count of VIO page-outs; word 6 contains a count of VIO reclaims; word 7 contains a count of pages swapped in; word 8 contains a count of pages swapped out; word 9 contains a count of swap-outs. Word 10 contains a count of common area page-ins; word 11 contains a count of common area reclaims; word 12 contains a count of pages stolen; word 13 contains a count of LPA page-ins; word 14 contains a count of page reclaims; words 15 and 16 contain a count of processor page-seconds.

SYSEVENT 28 HEX (COPYDMDT): Issued to notify SRM that a DISPLAY command with the keyword DMN has been entered.

Locks held on entry to SRM: Local.

Register 0: Byte 3 contains the SYSEVENT code.

Register 1: Contains a pointer to a 2584 byte data area.

SRM returns the following information: Register 1: contains the same address as input register 1.

Return Area: word 1 contains a count of the domains in bytes 0-1; the remainder of the area contains a copy of the domain descriptor table.

SYSEVENT 29 HEX (DONTSWAP): Issued to notify SRM that the issuing address space must not be swapped out until an OKSWAP or an INITDET or an INITATT SYSEVENT has occurred.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID of the issuing address space or zero; byte 3 contains the SYSEVENT code.

SRM returns the following information:

Register 1: Byte 3 contains: X'08' if the request is not authorized or if the outstanding count of DONTSWAP requests has reached its maximum value; X'04' if the request is not for the current address space; X'00' if the request to mark the address space as "not swappable" was honored.

SYSEVENT 2A HEX (OKSWAP): Issued to notify SRM that the issuing address space may again be considered for swapping.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID of the issuing address space or zero; byte 3 contains the SYSEVENT code.

SRM returns the following information:

Register 1: Byte 3 contains: X'08' if the request is not authorized; X'04' if the request is not for the current address space; X'00' if the request to mark the address space as swappable was honored.

SYSEVENT 2B HEX (REQSWAP): Causes an address space to be swapped out. Issued, for example, when VARY storage wants to swap out the address space that occupies the storage to be taken offline.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of the ECB to be posted if a dependency exists on the requested swap (or zero if no dependency exists on the swap). An ECB can be specified only if the request is for the current address space.

SRM returns the following information:

Register 1: Byte 3 contains X'00' if the swap-out request was honored; X'0C' if the request was ignored because the address space was in the process of being swapped.

Post Codes:

X'00' - Requested swap-out is complete

X'04' - Address space became non-swappable before it could be swapped out.

SYSEVENT 2C HEX (BRINGIN): Issued when a CANCEL command has been issued for a job. BRINGIN ensures that if a job has been swapped-out because of a resource shortage, the cancellation can take effect without waiting until the shortage is relieved.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

SRM returns the following information:

Register 1: Byte 3 contains: X'00' if the swap-in request was honored; X'08' if the request was ignored because the address space was in the process of being swapped.

SYSEVENT 2D HEX (WKLDINIT): Issued by MF/1 to request that SRM begin collecting workload activity data.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of a global, fixed data collection buffer.

SRM returns the following information:

Register 1: Bytes 0-3 contain X'00' if the data collection buffer was accepted.

Register 15: Byte 3 contains: 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.

SYSEVENT 2E HEX (WKLDCOLL): Issued by measurement facility routines at the end of a reporting interval to collect workload activity.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of a fixed buffer, into which the collected workload activity measurements are to be copied.

SRM returns the following information:

Register 1: Bytes 0-3 are unchanged.

Register 15: Byte 3 contains: 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.

SYSEVENT 2F HEX (WKLDTERM): Issued by measurement facility routines to terminate workload activity data recording, at the termination of measurement or when an IPS change has occurred.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain 0.

SRM returns the following information:

Register 1: Bytes 0-3 contain the address of the global, fixed workload activity data collection buffer no longer being used by SRM.

Register 15: Byte 3 contains: X'00' if the request was honored and no exception conditions were found; X'40' if the data collection buffer has not been established.

SYSEVENT 30 HEX (unnamed): Issued by the SRM when the control function must be invoked immediately (that is, without waiting for the next SYSEVENT to be issued by another component).

Locks held on entry to SRM: None.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of the issuing SRB.

SYSEVENT 31 HEX (REQSVDAT): Issued by SMF during job or session termination to obtain user related-service data.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of an area where the service data is to be stored.

SRM returns the following information:

Register 15: Byte 3 contains X'40' if data was lost due to an accumulation control block error; otherwise it contains X'00'.

Register 1: Contains the same address as input register 1.

Return area for TSO User: Word 1 contains the total service; word 2 contains the total transaction active time for all transactions; word 3 contains the last performance group number in bytes 0-1, and the total number of transactions in bytes 2-3; word 4 contains the session residency time; word 5 contains the session processor service; word 6 contains the I/O service; word 7 contains the storage service; word 8 contains the SRB service.

Return area for non-TSO User: Word 1 contains the total service; word 2 contains the total transaction active time; word 3 contains the last performance group number in bytes 0-1; word 4 contains the session residency time.

SYSEVENT 32 HEX (HOLD) applies to MVS/System Extensions: Issued to notify SRM that the issuing address space must not be swapped out until a NOHOLD (SYSEVENT 33 HEX).

Locks held on entry: No lock required or prohibited.

Register 0: Bytes 0-1 contains the ASID (or zero indicating current ASID); byte 3 contains the SYSEVENT code.

SYSEVENT 33 HEX (NOHOLD) applies to MVS/System Extensions: Issued to notify SRM that the issuing address space, which has previously issued a HOLD SYSEVENT, may be again considered for swapping.

Locks held on entry: No lock required or prohibited.

Register 0: Bytes 0-1 contains the ASID (or zero indicating current ASID); byte 3 contains the SYSEVENT code.

Module Summary

For each module name prefix, (first three characters in the name of an object module), this summary identifies the corresponding component and program logic manual (PLM). Where the module name prefix is also a system message number prefix, the appropriate message manual is also identified.

- Components are identified by component number (5752-SC1xx for VS2 SCP components, 5744-Axx for emulators). Refer to the component summary (immediately following the module summary) for component names, component microfiche order numbers, and primary PLM order numbers.
- Publications are identified by order number.

AHL Components: 5752-SC1xx, xx = 11. PLMs: SY28-0643. Message manual: GC38-1002.

AKJ Components: 5752-SC1xx, xx = T5. PLMs: SY28-0652.

AMA Components: 5752-SC1xx, xx = 12,16. PLMs: SY28-0643. Message manual: GC38-1002.

AMD Components: 5752-SC1xx, xx = DE,D0,11,12,15,18. PLMs: SY26-3832, SY28-0643. Message manual: GC38-1002.

HEW Components: 5752-SC1xx, xx = 04,05. PLMs: SY26-3814, SY26-3815. Message manual: GC38-1007.

HMA Components: 5752-SC1xx, xx = 30. PLMs: SY28-0685. Message manual: GC28-0673.

HMB Components: 5752-SC1xx, xx = 14. PLMs: SY28-0643. Message manual: GC38-1002.

IAS Components: 5752-SC1xx, xx = BH,B2. PLMs: SY24-6000.

IAT Components: 5752-SC1xx, xx = BA. PLMs: SY28-0612.

IBC Components: 5752-SC1xx, xx = I0,I1. PLMs: SY35-0005. Message manual: GC38-1005.

ICA Components: 5752-SC1xx, xx = I2. PLMs: SY35-0005.

ICB Components: 5752-SC1xx, xx = DR,DP. PLMs: SY35-0013. Message manual: GC38-1000.

ICG Components: 5752-SC1xx, xx = DQ,DT,CC. PLMs: SY35-0014, -0016. Message manual: GC38-1000.

IDA Components: 5752-SC1xx, xx = DA,DE,D6. PLMs: SY26-3825 SY26-3826 SY26-3834. Message manual: GC38-1000.

IDC Components: 5752-SC1xx, xx = DU, DK, SY35-0010. PLMs: SY35-0013. Message manual: GC38-1000.

IDD Components: 5752-SC1xx, xx = DG. PLMs: SY26-3834.

IDF Components: 5752-SC1xx, xx = 22. PLMs: GY30-2000.

IEA Components: 5752-SC1xx, xx = B4, CH, CJ, CK, CL, CM, CP, CR, CU, CV, C3, C4, C5, C8, C9, D1, 02, 10. PLMs: SBOF-8210, SY26-3827, SY26-3834, SY28-0623, SY28-0678. Message manual: GC38-1002.

IEB Components: 5752-SC1xx, xx = SC1DN,UA,UG,UH,UJ,UK,U6,U7,U8,U9. PLMs: SY35-0005. Message manual: GC38-1005.

IEC Components: 5752-SC1xx, xx = CA,CC,C3,C6,D0,20. PLMs: SY26-3823, SY26-3832, SY26-3834, SY27-7246. Message manual: GC38-1002.

IED Components: 5752-SC1xx, xx = T3,T8,21. PLMs: SY28-0651, SY30-2040. Message manual: GC38-1002.

IEE Components: 5752-SC1xx, xx = B4,B5,B8,B9,CK,CM,CV,CX,CZ,C4,C5,T4,00,20. PLMs: SBOF-8210, SY28-0623, SY28-0651. Message manual: GC38-1002.

IEF Components: 5752-SC1xx, xx = B2,B3,B4,B5,B6,B7,B8,B9,DE,T4,00,01. PLMs: SBOF-8210, SY26-3825, SY28-0623. Message manual: GC38-1002.

IEH Components: 5752-SC1xx, xx = UC,UD,UF,UY,U0,U2,U3. PLMs: SY35-0005. Message manual: GC38-1005.

IEI (Message number prefix). Message manual: GC38-1002.

IEW Components: 5752-SC1xx, xx = C2,C7. PLMs: SBOF-8210. Message manual: GC38-1007.

IEZ Components: 5752-SC1xx, xx = B9. PLMs: SBOF-8210.

IFA Components: 5752-SC1xx, xx = 02. PLMs: SBOF-8210. Message manual: GC38-1002.

IFB Components: 5752-SC1xx, xx = CD. PLMs: SY28-0678. Message manual: GC38-1045.

IFC Components: 5752-SC1xx, xx = CD. PLMs: SY28-0678. Message manual: GC38-1045.

IFD Components: 5752-SC1xx, xx = 06. PLMs: SY28-0676. Message manual: GC38-1006.

IFF Components: 5752-SC1xx, xx = G0,07. PLMs: SY27-7241, SY27-7242, SY27-7260. Message manual: GC27-6974.

IFG Components: 5752-SC1xx, xx=DE,D0,D1. PLMs: SY26-3825, SY26-3826, SY26-3827, SY26-3832, SY26-3834.

IFH Components: 5752-SC1xx, xx=UE. PLMs: SY35-0005.

IFN Components: 5752-SC1xx, xx=03. PLMs: SY33-8041.

IFO Components: 5752-SC1xx, xx=03. PLMs: SY33-8041. Message manual: GC33-4021.

IGA Components: 5752-SC1xx, xx=CY. PLMs: None.

IGC Components: 5752-SC1xx, xx=BH, B8, CC, CE, C3, DA, DC, DE, D0, D1, D3, D4, D7, D8, G0, T1, T3, T4, UD, UF, U0, 06, 09, 11, 20, 21; 5744-AG1. PLMs: SBOF-8210, SY26-3820, SY26-3825, SY26-3827, SY26-3828, SY26-3831, SY26-3832, SY28-0623, SY28-0643, SY28-0676, SY28-0678, SY30-2040, SY35-0004, SY35-0005.

IGE Components: 5752-SC1xx, xx=CC. PLMs: SY26-3823. Message Manual: GC38-1002.

IGF Components: 5752-SC1xx, xx=CE. PLMs: SBOF-8210, SY28-0623, SY28-0678. Message manual: GC38-1002.

IGG Components: 5752-SC1xx, xx=C3, DB, DD, DE, DF, DL, D0, D1, D3, D4, D5, D6, D7, D8, G0, T3, UF, U0, U6, 09, 20, 21; 5744-ASI. PLMs: GY21-0013, SY21-0012, SY24-5162, SY24-5163, SY26-3826, SY26-3827, SY26-3828, SY26-3831, SY26-3832, SY26-3834, SY27-7246, SY30-2040, SY35-0005, SY35-0011.

IGX Components: 5752-SC1xx, xx=CG,CQ,C3. PLMs: SY26-3823.

IHB (Message number prefix.) Message manual: GC38-1002.

IHC Components: 5752-SC1xx, xx=07. PLMs: SY27-7242.

IHD Components: 5752-SC1xx, xx=07. PLMs: SY27-7242.

IHE Components: 5752-SC1xx, xx=07. PLMs: SY27-7242.

IHJ Components: 5752-SC1xx, xx=09. PLMs: SY26-3820. Message manual: GC38-1002.

IIN Components: 5744-AM1 PLMs: GY27-7187. Message manual: GC27-6951.

IIO Components: 5744-AK1 PLMs: GY27-7228. Message manual: GC27-6948.

IIP Components: 5744-AL1 PLMs: GY27-7229. Message manual: GC27-6952.

IIQ Components: 5744-AH1 PLMs: GY33-7011. Message manual: GC33-2008.

IIR Components: 5744-AG1 PLMs: GY33-7012. Message manual: GC33-2009.

IIT Components: 5744-AM1 PLMs: GY27-7187. Message manual: GC27-6951.

IIU Components: 5744-AJ1 PLMs: GY27-7238. Message manual: GC27-6948.

IIV Components: 5744-AS1 PLMs: GY26-3741. Message manual: GC26-3777.

IKJ Components: 5752-SC1xx, xx=T0, T1, T2, T3, T4, T7, T8, T9. PLMs: SY27-7269, SBOF-8210, SY28-0623, SY28-0650, SY28-0651, SY28-0652, SY28-0654, SY30-2040, SY33-8548, SY35-0004. Message manual: GC38-1002, GC38-1046.

IKT Components: 5752-SC1T9. PLMs: SY27-7269, SB0F-8210. Message manuals: GC38-1002, GC38-1046.

ILR Components: 5752-SC1xx, xx=CW. PLMs: SBOF-8210 Message manual: GC38-1002.

IMC Components: 5752-SC1xx, xx=B5. PLMs: SBOF-8210. Message manual: GC38-1002.

IMD Components: 5752-SC1xx, xx=D1. PLMs: SY26-3827.

IOD Components: 5752-SC1xx, xx=DM.. PLMs: SY24-5167.

IOE Components: 5752-SC1xx, xx=DM.. PLMs: SY24-5167.

IPD Message manual: GC38-1002.

IRA Components: 5752-SC1xx, xx=CX. PLMs: SBOF-8210. Message manual: GC38-1002.

IRB Components: 5752-SC1xx, xx=CQ. PLMs: SBOF-8210. Message manual: GC38-1004.

IST Components: 5752-SC1xx, xx=23. PLMs: SY27-7256, SY28-0621. Message Manual: GC28-1002.

The following module summary information is included in the system microfiche:

Module Index

Provides the following information for each module:

- Distribution library (DLIB)
- Component
- Change status (new, old, update)
- Alias
- Object module release status information (OBJRSI)

Module Directory

Identifies the modules contained in each distribution library.

Module Status

Provides the following information for each distribution library:

- Module names and aliases
- Module sizes
- Identification of new modules, and size changes for old modules
- Module SSI (system status index), old and new
- Summary data (total number of modules, alias; total number of modules added, changed, deleted)

Component Summary

For each OS/VS2 component, this summary identifies the component microfiche, the object modules that belong to the component, and the program logic manual (PLM) that provides primary documentation.

- Object modules that belong to the component are identified by module name prefix. A single prefix may apply to modules of more than one component, and to more than one PLM. To identify all components or all PLMs for a particular module name prefix, refer to the module summary earlier in this section.
- PLMs are identified by order number.

5752-SC1BH — JES2. Microfiche: SJD2-4230. Module name prefixes: IAS,IGC,HASP. Primary PLM: SY24-6000.

5752-SC1BZ — MSS Recovery Management. Module name prefixes. IDC Primary PLM. SBOF-8210.

5752-SC1B2 — External Writer. Microfiche: SJD2-4240. Module name prefixes: IAS,IEF. Primary PLM: SBOF-8210.

5752-SC1B3 — Scheduler Restart. Microfiche: SJD2-4250. Module name prefixes: IEF. Primary PLM: SBOF-8210.

5752-SC1B4 — Allocate/Unallocate. Microfiche: SJD2-4260. Module name prefixes: IEA,IEE,IEF. Primary PLM: SBOF-8210.

5752-SC1B5 — SWA Manager. Microfiche: SJD2-4270. Module name prefixes: IEE,IEF. Primary PLM: SBOF-8210.

5752-SC1B6 — Initiator. Microfiche: SJD2-4280. Module name prefixes: IEF. Primary PLM: SBOF-8210.

5752-SC1B8 — Master Scheduler Commands. Microfiche: SJD2-4300. Module name prefixes: IEE,IEF,IGC. Primary PLM: SBOF-8210.

5752-SC1B9 — Converter/Interpreter. Microfiche: SJD2-4310. Module name prefixes: IEE,IEF,IEZ. Primary PLM: SBOF-8210.

5752-SC1CA — DASD ERP. Microfiche: SJD2-4320. Module name prefixes: IEC,IGE. Primary PLM: SY26-3823.

5752-SC1CB — Unit Record ERP. Microfiche: SJD2-4330. Module name prefixes: IGE. Primary PLM: SY26-3823.

5752-SC1CC — Tape ERP/VES. Microfiche: SJD2-4340. Module name prefixes: IEC,IGC,IGE. Primary PLM: SY26-3823.

5752-SC1CD — OBR/EREP/RDE. Microfiche: SJD2-4350. Module name pretixes: IFB,IFC,IGE. Primary PLM: SY28-0678. (OBR1RDE), SY28-0773 (EREP).

5752-SC1CE — RMS(Recovery Management Support). Microfiche: SJD2-4360. Module name prefixes: IGC,IGE,IGF. Primary PLM: SBOF-8210.

5752-SC1CF — Extended SVC Router. Microfiche: SJD2-4370. Module name prefixes: ICG. Primary PLM: SBOF-8210.

5752-SC1CG — SVC109. Microfiche: SJD2-4380. Module name prefixes: IGX. Primary PLM: SBOF-8210.

5752-SC1CH — Virtual Storage Management. Microfiche: SJD2-4390. Module name prefixes: IEA. Primary PLM: SBOF-8210..

5752-SC1CJ — Contents Supervisor. Microfiche: SJD2-4400. Module name prefixes: IEA. Primary PLM: SBOF-8210.

5752-SC1CK — Communications Task. Microfiche: SJD2-4410. Module name prefixes: IEA,IEE. Primary PLM: SBOF-8210.

5752-SC1CL — Task Management. Microfiche: SJD2-4420. Module name prefixes: IEA. Primary PLM: SBOF-8210.

5752-SC1CM — Recovery/Termination. Microfiche: SJD2-4430. Module name prefixes: IEA,IEE. Primary PLM: SBOF-8210.

5752-SC1CP — Extended Precision Floating Point. Microfiche: SJD2-4440. Module name prefixes: IEA. PLM: SBOF-8210.

5752-SC1CQ — MF/1. Microfiche: SJD2-4450. Module name prefixes: IGX,IRB. Primary PLM: SBOF-8210.

5752-SC1CR — Real Storage Management. Microfiche: SJD2-4460. Module name prefixes: IEA. Primary PLM: SBOF-8210.

5752-SCICS — Cond Assembly Switch. Microfiche: none.

5752-SC1CT — BLDLIST Microfiche: None.

5752-SC1CU — Region Control Task. Microfiche: SJD2-4470. Module name prefixes: IEA. Primary PLM: SBOF-8210.

5752-SC1CV — Timer Supervision. Microfiche: SJD2-4480. Module name prefixes: IEA,IEE. Primary PLM: SBOF-8210.

5752-SC1CW — Auxiliary Storage Management. Microfiche: SJD2-4490. Module name prefixes: ILR. Primary PLM: SBOF-8210

5752-SC1CX — System Resources Manager. Microfiche: SJD2-4500. Module name prefixes: IEE,IRA. Primary PLM: SBOF-8210.

5752-SC1CY — Radix Partition Tree Service. Microfiche: SJD2-4510. Module name prefixes: IGA. Primary PLM: None.

5752-SC1CZ — MP Reconfiguration. Microfiche: SJD2-4520. Module name prefixes: IEE. Primary PLM: SBOF-8210.

5752-SC1C2 — Overlay Supervisor. Microfiche: SJD2-4540. Module name prefixes: IEW. Primary PLM: SBOF-8210.

5752-SC1C3 — IOS. Microfiche: SJD2-4550. Module name prefixes:

IEA,IEC,IGC,IGE,IGG,IGX. Primary PLM: SY26-3823.

5752-SC1C4 — DIDOCS. Microfiche: SJD2-4560. Module name prefixes: IEA,IEE. Primary PLM: SBOF-8210.

5752-SC1C5 — Supervisor Control. Microfiche: SJD2-4570. Module name prefixes: IEA,IEE. Primary PLM: SBOF-8210.

5752-SC1C6 — EXCP. Microfiche: SJD2-4580. Module name prefixes: IEC. Primary PLM: SY26-3823.

5752-SC1C7 — FETCH. Microfiche: SJD2-4590. Module name prefixes: IEW. Primary PLM: SBOF-8210.

5752-SC1C8 — NIP. Microfiche: SJD2-4600. Module name prefixes: IEA. Primary PLM: SY28-0623.

5752-SC1C9 — IPL Microfiche: SJD2-4610 Module name prefixes: IEA. Primary PLM: SY28-0623.

5752-SC1DA — Block Processor. Microfiche: SJD2-4620. Module name prefixes: IDA,IGC. Primary PLM: SY26-3825.

5752-SC1DB — SAM Subsystem Interface. Microfiche: SJD2-4630. Module name prefixes: IGE,IGG. Primary PLM: SY26-3832.

5752-SC1DC — Password Protect. Microfiche: SJD2-4640. Module name prefixes: IGC. Primary PLM: SY26-3827.

5752-SC1DD — 3505/3525 Reader/Punch. Microfiche: SJD2-4650. Module name prefixes: IGG. Primary PLM: SY26-3832.

5752-SC1DE — VSAM/VSAM Catalog. Microfiche: SJD2-4660. Module name prefixes: AMD,IDA,IEF,IFG,IGC,IGG. Primary PLM: SY26-3825, SY26-3826.

5752-SC1DF — IBM 3890 Document Processor. Microfiche: SJD2-4670. Module name prefixes: IGE,IGG. Primary PLM: SY24-5163.

5752-SC1DG — VBP. Microfiche: SJD2-4680. Module name prefixes: IDA,IDD. Primary PLM: SY26-3834.

5752-SC1DH — Catalog Controller 3 CVOL Processor. Microfiche: SJD2-4690. Module name prefixes: IGC,IGG. Primary PLM: SY36-3860.

5752-SC1DJ — Window Intercept. Microfiche: SJD2-4700. Module name prefixes: None (macros only). Primary PLM: SY26-3834.

5752-SC1DK — Access Method Services. Microfiche: SJD2-4710. Module name prefixes: IDC. Primary PLM: SY35-0010.

5752-SC1DL — IBM 3886 OCR. Microfiche: SJD2-4720. Module name prefixes: IGE,IGG. Primary PLM: SY24-5162.

5752-SC1DM — IBM 3895 Document Reader/Inscriber. Microfiche: SJD2-xxxx. Module name prefix IOD. Primary PLM: None.

5752-SC1DN — IBM OS/VS2 3540 Logic. Module name prefixes: IOE, IEB, IGG, IGE. Program PLM: SY24-5167.

5752-SC1DP — MSSC. Microfiche: SJD2-5370. Module name prefixes: ICB, IEA. Primary PLM: SY35-0013.

5752-SC1DQ — MSCTC. Microfiche: SJD2-5470. Module name prefixes: ICG. Primary PLM: SY35-0016.

5752-SC1DR — MSRC. Microfiche: SJD2-5380. Module name prefixes: ICB. Primary PLM: SY35-0013.

5752-SC1DT — MSS TRACE. Microfiche: SJD2-5400. Module name prefixes: ICG. Primary PLM: SY35-0014.

5752-SC1DU — MSS Utilities. Microfiche: SJD2-5410. Module name prefixes: IDC. Primary PLM: SY35-0013.

5752-SC1D0 — SAM. Microfiche: SJD2-4730. Module name prefixes: AMD,IEC,IFG,IGC,IGE,IGG. Primary PLM: SY26-3832.

5752-SC1D1 — Open/Close/EOV. Microfiche: SJD2-4740. Module name prefixes: IEA,IFG,IGC,IGG,IMD. Primary PLM: SY26-3827.

5752-SC1D2 — PAM. Microfiche: SJD2-4750. Module name prefixes: IGC,IGG. Primary PLM: SY26-3832.

5752-SC1D4 — DADSM. Microfiche: SJD2-4770. Module name prefixes: IGC,IGG. Primary PLM: SY26-3828.

5752-SC1D5 — OCR. Microfiche: SJD2-4780. Module name prefixes: IGG. Primary PLM: GY21-0013.

5752-SC1D6 — MICR. Microfiche: SJD2-4790. Module name prefixes: IGG. Primary PLM: GY21-0012.

5752-SC1D7 — DAM. Microfiche: SJD2-4800. Module name prefixes: IGC,IGG. Primary PLM: SY26-3831.

5752-SC1D8 — ISAM. Microfiche: SJD2-4810. Module name prefixes: IGC,IGG. Primary PLM: SY26-3833.

5752-SC1G0 — GAM. Microfiche: SJD2-4820. Module name prefixes: IFF,IGC,IGE,IGG. Primary PLM: SY27-7260.

5752-SC1I0 — IBCDMPRS. Microfiche: SJD2-4830. Module name prefixes: IBC. Primary PLM: SY35-0005.

5752-SC1I1 — IBCDASDI. Microfiche: SJD2-4840. Module name prefixes: IBC. Primary PLM: SY35-0005.

5752-SC1I2 — ICAPRTBL. Microfiche: SJD2-4850. Module name prefixes: ICA. Primary PLM: SY35-0005.

5752-SC1PV — Private Macros. Microfiche: none.

5752-SC1T0 — TSO EDIT. Microfiche: SJD2-4860. Module name prefixes: IKJ. Primary PLM: SY33-8548.

5752-SC1T1 — TSO TEST. Microfiche: SJD2-4870. Module name prefixes: IGC,IKJ. Primary PLM: SY35-0004.

5752-SC1T2 — TSO Utilitites. Microfiche: SJD2-4880. Module name prefixes: IKJ. Primary PLM: SY28-0652.

5752-SC1T3 — TSO TIOC. Microfiche: SJD2-4890. Module name prefixes: IED,IGC,IGG,IKJ. Primary PLM: SY30-2040.

5752-SC1T4 — TSO Scheduler. Microfiche: SJD2-4900. Module name prefixes: IEE,IEF,IGC,IKJ. Primary PLM: SBOF-8210, SY28-0651.

5752-SC1T5 — Link Loadgo Prompter. Microfiche: SJD2-4910. Module name prefixes: AKJ. Primary PLM: SY28-0652.

5752-SC1T8 — TSO TCAM Subroutines. Microfiche: SJD2-4920. Module name prefixes: IED,IKJ. Primary PLM: SY30-2040.

5752-SC1T9 — VTIOC TCAS Microfiche: SJD2-6196 Module name prefixes IKT, described in SY27-7269, SBOF-8210 and SYB8-0606.

5752-SC1UA — IEBPTPCH. Microfiche: SJD2-4930. Module name prefixes: IEB. Primary PLM: SY35-0005.

5752-SC1UC — IEHMOVE. Microfiche: SJD2-4940. Module name prefixes: IEH. Primary PLM: SY35-0005.

5752-SC1UD — IEHINITT. Microfiche: SJD2-4950. Module name prefixes: IEH,IGC. Primary PLM: SY35-0005.

5752-SC1UE — IEHSTATR. Microfiche: SJD2-4960. Module name prefixes: IFH. Primary PLM: SY35-0005.

5752-SC1UF — IEHATLAS. Microfiche: SJD2-4970. Module name prefixes: IEH,IGC,IGG. Primary PLM: SY35-0005.

5752-SC1UG — IEBTCRIN. Microfiche: SJD2-4980. Module name prefixes: IEB,IGE. Primary PLM: SY35-0005.

5752-SC1UH — IEBISAM. Microfiche: SJD2-4990. Module name prefixes: IEB. Primary PLM: SY35-0005.

5752-SC1UJ — IEBDG. Microfiche: SJD2-5000. Module name prefixes: IEB. Primary PLM: SY35-0005.

5752-SC1UK — IEBCOMPR. Microfiche: SJD2-5010. Module name prefixes: IEB. Primary PLM: SY35-0005.

5752-SC1UN MVS Device Support Facilities. Microfiche: SJD2-6180. Module name prefixes: ICK. Primary PLM: SY35-0030.

5752-SC1UR Stand Alone Device Support Facilities. Microfiche: SJD2-6180. Module name prefixes: ICL. Primary PLM: SY35-0030.

5752-SC1UX — SGIEH402. Microfiche: None.

5752-SC1UY — IEHUCAT. Microfiche: SJD2-5020. Module name prefixes: IEH. Primary PLM: SY35-0005.

5752-SC1U0 — IEHDASDR. Microfiche: SJD2-5030. Module name prefixes: IEH,IGC,IGG. Primary PLM: SY35-0005.

5752-SC1U2 — IEHLIST. Microfiche: SJD2-5040. Module name prefixes: IEH. Primary PLM: SY35-0005.

5752-SC1U3 — IEHPROGM. Microfiche: SJD2-5050. Module name prefixes: IEH. Primary PLM: SY35-0005.

5752-SC1U6 — IEBCOPY. Microfiche: SJD2-5060. Module name prefixes: IEB, IGG. Primary PLM: SY35-0005.

5752-SC1U7 — IEBGENER. Microfiche: SJD2-5070. Module name prefixes: IEB. Primary PLM: SY35-0005.

5752-SC1U8 — IEBUPDTE. Microfiche: SJD2-5080. Module name prefixes: IEB. Primary PLM: SY35-0005.

5752-SC1U9 — IEBEDIT. Microfiche: SJD2-5090. Module name prefixes: IEB. Primary PLM: SY35-0005.

5752-SC100 — SMF Scheduler. Microfiche: SJD2-5120. Module name prefixes: IEE,IEF. Primary PLM: SBOF-8210.

5752-SC101 — Mapping/Common Supervisor Macros. Microfiche: SJD2-5130. Module name prefixes: IEF. Primary PLM: SYB8-0606.

5752-SC102 — SMF. Microfiche: SJD2-5140. Module name prefixes: IEA,IFA. Primary PLM: SBOF-8210.

5752-SC103 — Assembler XF. Microfiche: SJD2-5150. Module name prefixes: IFN,IFO. Primary PLM: SY33-8041.

5752-SC104 — Linkage Editor. Microfiche: SJD2-5160. Module name prefixes: HEW. Primary PLM: SY26-3815.

5752-SC105 — Loader. Microfiche: SJD2-5170. Module name prefixes: HEW. Primary PLM: SY26-3814.

5752-SC106 — OLTEP. Microfiche: SJD2-5180. Module name prefixes: IFD,IGC,IGE. Primary PLM: SY28-0676.

5752-SC107 — GSP. Microfiche: SJD2-5190. Module name prefixes: IFF,IHC,IHD,IHE. Primary PLM: SY27-7242.

5752-SC109 — Checkpoint/Restart. Microfiche: SJD2-5200. Module name prefixes: IGC,IGG,IHJ. Primary PLM: SY26-3820.

5752-SC10E — PWF Support (Power Warning Feature Support). Microfiche: SJD2-5110. Module name prefixes: ICF. Primary PLM: SBOF-8210.

5752-SC111 — GTF. Microfiche: SJD2-5220. Module name prefixes:

AHL,AMD,IGC. Primary PLM: SY28-0643.

5752-SC112 — AMASPZAP. Microfiche: SJD2-5230. Module name prefixes: AMA. Primary PLM: SY28-0643.

5752-SC113 — AMDPRDMP. Microfiche: SJD2-5240. Module name prefixes: AMD. Primary PLM: SY28-0643.

5752-SC114 — AMBLIST. Microfiche: SJD2-5250. Module name prefixes: HMB. Primary PLM: SY28-0643.

5752-SC115 — AMDSADMP. Microfiche: SJD2-5260. Module name prefixes: AMD. Primary PLM: SY28-0643.

5752-SC116 — AMAPTFLE. Microfiche: SJD2-5270. Module name prefixes: AMA. Primary PLM: SY28-0643.

5752-SC118 — AMDPRDMP/EDIT. Microfiche: SJD2-5280. Module name prefixes: AMD. Primary PLM: SY28-0643.

5752-SC120 — BTAM. Microfiche: SJD2-5290. Module name prefixes: IEC,IEE,IGC,IGE,IGG. Primary PLM: SY27-7246.

5752-SC121 — TCAM. Microfiche: SJD2-5300. Module name prefixes: IED,IGC,IGE,IGG. Primary PLM: SY30-2059.

5752-SC123 — VTAM. Microfiche: SJD2-5320. Module name prefixes: IST. Primary PLM: SY27-7256, SY28-0621.

5752-SC130 — SMP. Microfiche: SJD2-5330. Module name prefixes: HMASMP. Primary PLM: SY28-0685.

The microfiche index, included in the component microfiche, provides the following additional information for each component:

- Module names
- Module change status (new, old, updated) and release number
- Module microfiche level and microfiche count for updated modules
- Module linkage editor characteristics (REUS, RENT, REFR)

Storage Summary

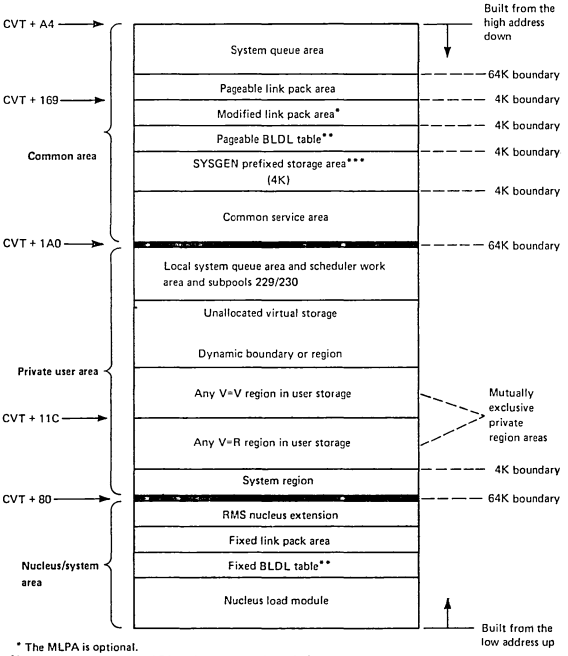
This summary briefly describes the use of storage in OS/VS2. Topics are:

- Storage Maps
- Storage Protection
- Storage Subpools

For more information on storage usage, refer to the publication *OS/VS2 System Programming Library: Initialization and Tuning Guide*.

Storage Maps

The following figures describe the layout of real and virtual storage. For a description of the prefixed storage area, refer to "PSA" in the "Data Area Descriptions" portion of Volume 3.



* The MLPA is optional.
** The pageable and fixed BLDL tables are mutually exclusive.
*** The SYSGEN PSA is only for MP systems.

Figure 30. Virtual Storag. Layout for Single Address Space

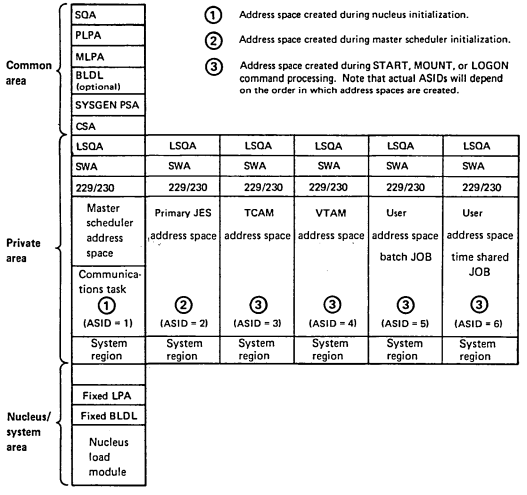


Figure 31. Virtual Storage Layout for Multiple Address Spaces

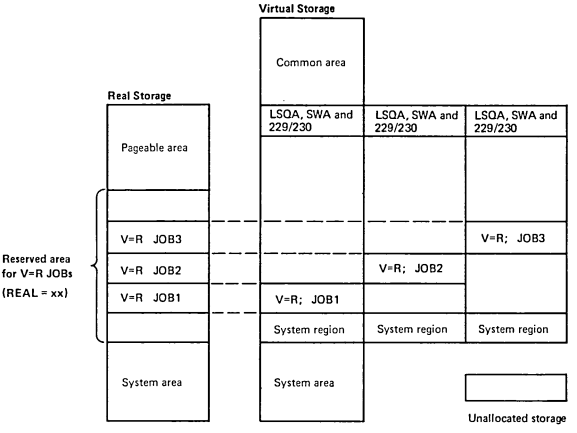


Figure 32. Mapping of V=R Regions Into Real Storage

Storage Protection

For each 2K block of real storage, there is a seven-bit control field called a "key in storage." This key is used as follows:

Access Control Bits: Bits 0-3 are matched against the four-bit protection key in the PSW whenever information is stored, or whenever information is fetched from a location that is protected against fetching.

In OS/VS2, the 16 protection keys provided by the PSW (and matched against the access control bits) are assigned as follows:

Key	Assigned to:
0	Supervisor and other system functions that require access to the private area of storage.
1	Job scheduler and job entry subsystem.
2	VSPC. 3,4 (Reserved).
5	Data management, including IOS, ASM, block processor, and OPEN/CLOSE/EOV.
6	TCAM and VTAM.
7	IMS.
8	All V=V problem programs.
9-15	V=R problem programs (each protected by a unique protection key).

Fetch Protection Bit: Bit 4 indicates whether protection applies to fetch-type references. A zero indicates that only store-type references are monitored, and that fetching with any protection key is permitted; a one indicates that protection applies to both fetching and storing. No distinction is made between the fetching of instructions and the fetching of operands.

Reference Bit: Bit 5 is associated with dynamic address translation. It is normally set to one each time that a location in the related 2K storage block is referred to either for storing or for fetching of information.

Change Bit: Bit 6 is also associated with dynamic address translation. It is set to one each time that information is stored into the corresponding 2K block of storage.

Storage Subpools

A subpool is a group of logically related storage blocks identified by a subpool number. In a request for virtual storage, a subpool number indicates the type of storage that is requested. The meaning of each subpool number is defined in the following table.

Subpool Number	Indicates Request for	Attributes of Subpool	Notes
0 - 127	Space within a region	Job-oriented Pageable Job step's protection key* Fetch-protected	These are the only valid subpool numbers for problem programs. A request for a higher number will cause the problem program to be abnormally terminated. When subpool 0 is requested by programs in supervisor state and key 0, subpool 252 is assigned.
128			Reserved for compatibility with VS1. Treated as an error.
129 - 226			Undefined.
227	Fixed, global space (explicitly assigned and freed.)	Requester protection key Fixed System-oriented Explicitly assigned and freed Fetch-protected	Multiple-key system queue area space is assigned from the common service area.
228	Fixed, global space (explicitly assigned and freed.)	Requester protection key Fixed System-oriented Explicitly assigned and freed Not fetch-protected	Multiple-key system queue area space is obtained from the common service area.
229	Private area storage	Requester protection key Pageable Fetch-protected	Freed automatically at task termination. Assigned from top of private area.
230	Private area storage	Requester protection key Pageable Not fetch-protected	Freed automatically at task termination. Assigned from top of private area.
231	Space within CSA (explicitly assigned and freed)	Requester protection key Pageable Fetch-protected System-oriented Explicitly assigned and freed	Assigned in common service area.
232			Reserved. Treated as an error. Used in OS/VS2 Release 1 for TSO external page storage.

*A job step protection key is supplied from the TCB; a user protection key supplied from the PSW or via the GETMAIN macro.

Subpool Number	Indicates Request for	Attributes of Subpool	Notes
233	Space within LSQA (task-related)	Job-oriented Fixed Protection key = 0 Task-related Swappable Not fetch-protected	Allows a task running in key 0 to acquire accountable, fixed, protected storage that is job-oriented and freed at end of task. Space is assigned from subpool 253.
234	Space within LSQA (job-step-related)	Job-oriented Fixed Protection key = 0 Job-step-related Swappable Not fetch-protected	Allows a task running in key 0 to acquire accountable, fixed, protected storage that is job-oriented and freed at end of job step. Space is assigned from subpool 254.
235	Space within LSQA (explicitly assigned and freed)	Job-oriented Fixed Protection key = 0 Explicitly assigned and freed Not fetch-protected Swappable	Allows a task running in key 0 to acquire non-accountable, fixed, protected storage that is job-oriented. Space is assigned from subpool 255.
236	Space within SWA	For system use only Protection key = 1 Not fetch-protected	To assign or free pageable virtual storage for the scheduler work area.
237	Space within SWA	For system use only Protection key = 1 Not fetch-protected	To assign or free pageable virtual storage for the scheduler work area.
238			Reserved for compatibility with OS/VS1. Treated as an error.
239	Fixed, Global Space (explicitly assigned and freed)	Fetch-protected Protection key = 0 Explicitly assigned and freed	System queue area space obtained from the common service area (CSA). Treated as subpool 227 key 0 space.
240	Space within a region (job-step-related)	Job-oriented Pageable Job step's protection key Fetch-protected Job-step-related	Treated as subpool 250 to maintain compatibility with MFT and OS/VS1. Automatically freed at end of step.
241	Space within CSA	System-oriented Pageable User protection key Explicitly assigned and freed Not fetch-protected	Assigned in the common service area.
242	Nonpageable V = R region	For scheduler use only	A new nonpageable (V = R) region is assigned or an existing nonpageable region is freed.
243			Reserved. Treated as an error. Used in OS/VS2 Release 1 for SQA space.
244			Reserved. Treated as an error. Used in OS/VS2 Release 1 for SQA space.
245	Space within SQA (explicitly assigned and freed)	System-oriented Fixed Protection key = 0 Explicitly assigned and freed Not fetch-protected	Allows a task running in key 0 to acquire non-accountable, fixed, protected storage that is system-oriented.
246			Reserved. Treated as an error. Used in MVT to exchange regions.
247	Pageable (V = V) region	For scheduler use only	A new pageable (V = V) region is assigned or an existing pageable region is freed. External page storage allocation is assumed when using this subpool.
248			Reserved. Treated as an error. Used in MVT for rollout/rollin.
249			Reserved. Treated as an error. Used in OS/VS2 Release 1 for LSQA segments.
250	Space within a region	Job-oriented Pageable Job step's protection key Job-step-related Fetch-protected	Allows a task running in supervisor state and key 0 state to acquire unprotected storage in the user's region. All subpool 250 requests are assigned subpool 0 of the associated task.

Subpool Number	Indicates Request for	Attributes of Subpool	Notes
251	Space within a region	Job-oriented Job-step's protection key Job-step-related Fetch-protected	Allows an authorized task to acquire accountable, unprotected, pageable storage in the user's partition. Space is job-oriented and automatically freed at the termination of the job step. Used for modules not loaded into Subpool 252 from the low end of storage.
252	Space within a region	Job-oriented Protection key = 0 Job-step-related Not fetch-protected	Allows a task running in key 0 to acquire accountable, pageable, protected storage in the user's region that is job-oriented and automatically freed at the termination of the job-step task. Used for reenterable modules from authorized libraries.
253	Space within LSQA (task-related)	Job-oriented Fixed Protection key = 0 Task-related Not fetch-protected Swappable	Allows a task running in key 0 to acquire fixed, accountable, protected storage in the LSQA for the user's region that is job-oriented and freed when the task terminates.
254	Space within LSQA (job-step-related)	Job-oriented Fixed Protection key = 0 Job-step-related Swappable Not fetch-protected	Allows a task running in key 0 to acquire fixed, accountable, protected storage in the LSQA for the user's region that is job-oriented and freed when the job step terminates.
255	Space within LSQA (explicitly assigned and freed)	Job-oriented Fixed Protection key = 0 Explicitly assigned and freed Swappable Not fetch-protected	Allows a task running in key 0 to acquire fixed, non-accountable, protected storage in the LSQA that is job-oriented and must be explicitly freed.

Serialization Summary

This summary describes the use of locks and system ENQ/DEQ names.

Lock Summary

In OS/VS2, the set of locks is categorized by hierarchy (from highest to lowest) as follows:

DISP	Dispatcher - serializes certain global functions and storage in addition to users of the global intersect (applies to MVS/System Extensions).
ASM	Auxiliary storage management - serializes auxiliary storage resources.
SALLOC	Space allocation - serializes the resources of real storage management (RSM) and virtual storage management (VSM), and some auxiliary storage management resources.
IOSYNCH	I/O supervisor synchronization - serializes the IOS purge function and other IOS resources.
IOSCAT	IOS channel availability table lock - serializes IOS processor related save area.
IOSUCB	IOS unit control block lock - serializes access and updates to the unit control blocks. There is one lock per UCB.
IOSLCH	IOS logical channel queue lock - serializes access and updates to the IOS logical channel queues. There is one lock per channel queue.
SRM	System resources manager lock - serializes the use of SRM control blocks and associated data.
CMSEQDQ	ENQ/DEQ Cross Memory Services - Serializes ENQ/DEQ functions and the use of the ENQ/DEQ control blocks. (As applicable to MVS/System Extensions.)
CMS	General Cross Memory Services - this lock is used by all other global functions. (As applicable to MVS/System Extensions.)
LOCAL	Local Memory - serializes the memory related resources and users of the local intersect. (As applicable to MVS/System Extensions.)

The use of locks is based on the following considerations:

- At any one time, a processor can hold only one lock per class (as listed above). The cross memory services category is an exception. A processor can hold both cross memory services locks but only if they are unconditionally requested at the same time. For each processor, the location, (PSA+2FA) indicates the locks currently held.**
- Page faults are permitted for programs that own the LOCAL and/or CMS locks, but not for programs that own locks higher in the hierarchy.
- When requesting the CMS lock, a program must already own the LOCAL lock. When requesting any other lock, a program need not own locks that are lower in the hierarchy.
- Locks can be requested conditionally or unconditionally. However, only locks higher than those currently held by the processor can be requested unconditionally.

The following table lists the class, type, location and PSACLHS bit of each lock.

Lock Name	Category	Type	Location	PSACHLS bit
DISP	Global	Spin	IEAVESLA+0	1000
ASM	Global	Spin	(*)	0800
SALLOC	Global	Spin	IEAVESLA+4	0400
IOSYNCH	Global	Spin	(*)	0200
IOSCAT	Global	Spin	(*)	0100
IOSUCB	Global	Spin	(*)	0080
IOSLCH	Global	Spin	(*)	0040
SRM	Global	Spin	IEAVESLA+8	0004
CMSEQDQ	Glogal	Suspend	IEAVSLA+18 HEX	0002
CMS	Global	Suspend	IEAVESLA+20 HEX	0002
LOCAL	Local	Suspend	ASCB+80 HEX	0001

IEAVESLA is pointed to by the lock interface table. The lock interface table is pointed to by PSA location, PSA+2FC hex.

* For each processor, the PSACLHT field (PSA + 280 hex) contains the addresses of the ASM and IOS locks currently held.

Each lock is a fullword of storage. If the lock value is zero, the lock is available (that is, not owned). For owned locks, lock contents and descriptions are listed below.

Category/Type	Contents	Description
Global-Spin	0000 00xx	ID of the owning processor, as found in the PSACPULA field (PSA+206).
Global-Suspend	00xx xxxx	ASCB address.**
Local-Suspend	0000 00xx	ID of the owning processor.
	FFFF FFFF	Lock is held, but the holder is currently inactive due to an interruption while the lock was held.
	7FFF FFFF	Lock is held, but the holder is currently nondispatchable due to a page fault or a pending unconditional request for the CMS lock. When the page fault is resolved or the CMS lock is released, the local lock contents will be changed to FFFF FFFF to allow the address space to be redispatched.

** If an address space owned the CMS lock and/or the CMSEQDQ lock and was interrupted or suspended, the ASCBCMSH flag in the ASCBFLG1 field of the ASCB is turned on and the cross memory services lock-held indicator in the PSAHLHI field of the PSA is turned off until the address space is redispatched. The ASCB address remains in the CMS lock and/or the CMSEQDQ lock until they are released.

For more complete information, refer to *OS/VS2 System Logic Library* For information about the use of locks by SVC routines, refer to the SVC summary earlier in this section.

Note: IEAVESLA can be located as follows: the first word of the lock interface table (PSA+2FC) contain the address of the dispatcher lock which is at IEAVESLA +0.

Lock Reference Table (new for MVS/System Extensions)

The table below begins at offset 0. Each entry is 3 words in length and contains the lock address, mask, and entry point for each lock and lock request type. To find an individual lock's address, mask, or entry point, use the following:

Location	Length	Description
Offset+0	4	Lock Address
Offset+4	4	Mask for specified lock
Offset+8	4	Entry point into lock manager

For each type of lock the 3 word entry described above is duplicated. The appropriate offsets are listed below.

Lock/Request	Conditional Obtain	Unconditional Obtain	Release	Release Disabled
Dispatcher	0	C	18	24
IOSCAT	30	3C	48	54
IOSUCB	60	6C	78	84
IOSLCH	90	9C	A8	B4
IOSYNCH	C0	CC	D8	E4
TPNCB	F0	FC	108	114
TPDNCB	120	12C	138	144
TPACBDEB	150	15C	168	174
ASM	180	18C	198	1A4
SALLOC	1B0	1BC	1C8	1D4
SRM	1E0	1EC	1F8	204
General CMS	210	21C	228	-
Local Lock	234	240	24C	-
Multiple Spin	-	-	258	-
ENQ/DEQ CMS	264	270	27C	-
All CMS	-	288	294	-
All CMS	-	-	2A0	-

ENQ/DEQ Summary

Following is a list of major and minor ENQ/DEQ names, the resources that they represent, and the modules that use them.

Major	Minor	Resource - Using Modules
SYSDSN	dsname	System data sets - IEEVSND2, IEFAB4DC, IEFSD102, IEFSD161, IGC0002B, IGC0002I, IGC00030, IGG020P1, IHJACP00 (Dequeue only - IEFAB4A6, IEFSD162, IFQ0RR0E, IGG020P3, IGG0290D, IGG03001)
SYSIAT	CKPT	JES3 CHKPNT data set - IATINTK, IATINGL, IATINJB. *Note 1:* This is a device RESERVE rather than an ENQ macro. *Note 2:* Major/minor names are defined in macro IATYITK, which is assembled in-line in module IATINTK. Modules IATINGL and IATINJB reference the in-line expansion.
SYSIEA01	IEA	Dump data set - IEAVTABD
SYSIEFSD	CHNGDEVS	UCB - IEEMB813
	DDRTPUR	Swap unit record tape device or - IGFDU0, IGFDT0, ALLOCATION
	DDRDA	Swap DASD device - IGFDD0, ALLOCATION
	Q4	UCB - IEEVCPU, IEEVPTH, IEE3603D, IEFAB421 (Dequeue only - IEE3103D, IEE3303D, IEE4203D, IEE4403D, IEE4803D, IEE4903D, IEE7303D)
	Q6	Protect key resource - IEFSD161, IEFSD166 (Dequeue only - IEFIB620)
	Q10	CSCB - IEECB800, IEECB866, IEEMB810, IEEVMNT1, IEEVSND6, IEEVSTAR, IEEVWAIT, IEE0303D, IEE0703D, IEE0803D, IEE3703D, IEE5103D, IEFIRECM, IEFJRECM (Dequeue only - IEESB665, IEFISEXR)
	RPLL	Job journal data set - IEFXB500
	STCQUE	Started task control - IEFJSWT, IEEVWAIT (Dequeue only - IEESB670)
	TSOQUE	TSO data sets - IEFJSWT, IEEVWAIT (Dequeue only - IEESB670)
SYSIGGV1	MCATOPEN	Master catalog - IGG0CLAC (Dequeue only - IGG0CLAD)
SYSIGGV2	Catalog name catalog - IGG0CLA3	
SYSIKJBC	RBA	TSO broadcast data set (rba = relative block address) - IEEVSND2, IEEVSND3, IEEVSND8, IKJEES10, IKJEES40, IKJEES75, IKJRBBCR
SYSIKJUA	OPENUADS	User attribute data set - IKJEFA10, IKJEFA20, IKJEFA30, IKJEFA40, IKJEFLE, IKJEFLL, IKJRBBCR
	userid	TSO users - IKJEFA12, IKJEFA20, IKJEFA30, IKJEFLB, IKJRBBCR, IKJRBBU0 (Dequeue only - IKJEFLS)
SYSIEWLP	dsname for SYSLMOD SYSLMOD data set - HEWLFINT (Dequeue only -HEWLFFNL)	
SYSSJWTP	JSCBWTP	Job step messages - IEEJB840
	RPL + ASID	Message data set - (rpl = request parameter list pointer, asid = address space identifier) - IGC0203E
SYSSMF01	data set	SYS1.MAN Data Set - IEEMB829, IFASMFDP
SYSZUSRL	ucbaddr	User label tracks - IFG0202C, IFG0554L
SYSVSAM	ccccnnnx	VSAM data sets (cccc = ACB address, nnn = control interval number, x = status) - IDA0200T, IDA0231T, IDA0557A, IGG0CLBG
SYSVTOC	VOLSER	VTOC - IGC0007H, IGG0CLBU, IGG020P1, IGG0290E, IGG03001, IGG0321J, IGG0321K, IGG0321L, IGG0321R, IGG0325A, IGG0325E, IGG0553A, IGG0860A (Dequeue only - IFG0RR0E, IGC0107H, IGG020P3, IGG0321I7, IGG0325H, IGG086AE)

Major	Minor	Resource - Using Modules
SYSZRBMF	ACTIVE	Indicates that MF/1 is already active - IRBMFMFC
SYSZCAXW	CAXW	CAXWA (catalog auxiliary work area) - IDACAT11, IDACAT12, IGG0CLBG
SYSZCSD	CSDCPUJS	CSD field - IEEVCPU, IEFICPUA (Dequeue only - IEFIB620)
SYSZEC16	PURGE	Purge data set - IGC0001F
SYSZIGGI	ASID	TSB - IGC0009C, IGG09302
SYSZIST0C	Configuration	RDT Segment/Checkpoint
	Restart Data Set ddname	Data Set - ISTINCR4
SYSZOPEN	dsname	System Data Sets - IGC0002B
SYSZPCCB	PCCB	PCCB (private catalog control block) - IEFAB4F4, IEFAB4F5, IEFAB4FE, IEFAB469, IGG0CLA3
SYSZPGAD	PAGEADD	Serializes the PAGEADD command.
SYSZPSWD	dsname	Password data set - IFG0195U, IFG0195V (Dequeue only - IFG0RR0E)
SYSZSIPS	IRARMSET	SYSEVENT - IEEMB812
SYSZSMF1	BUF	SMF buffer - IEEMB825, IEEMB830
SYSZTIOT	ASID+DSAB	QDB addr TIOT (asid = address space identifier, DSAB QDB addr = address of the DSAB QDB) - IDACAT11, IDACAT12, IEFDB400, IFG0TC0A, IFG019RA, IGC0002G, IGC0002I, IGC00030, IGG020R1 (Dequeue only - IFG0RR0E, IGG0290D, IGG03001)
SYSZVARY	CPU	Reconfiguration commands - IEEMPDM, IEEMPS03, IEEVCPU
	VALIDATE	Storage validation process - IEEMPVST
SYSZVMV	ucbaddr	Virtual volumes - IEFAB49B
SYSZVOLS	volserno	System volumes - IEFAB4F0, IEFAB4F1, IEFAB421, IFG0194C, IFG0194F, IFG01960, IFG0552N, IFG0554L, IGC0002B, IGC0008B (Dequeue only - IEFAB4A8, IEFAB477, IFG0194A, IGG0290D, IFG0194J, IGC0K05B)

The following table refines the ENQ/DEQ summary for the job scheduler modules of MVS.

	Enqueue		
Resource	**Share**	**Exclusive**	**Dequeue Only**
SYSDSN data set name	IEFAB4DC	IEFAB4DC	IEFAB4A6 IEFAB4DD IEFBB410
SYSIEFSD CHNGDEVS	IEFAB421 IEFAB471 IEFAB487 IEFAB488 IEFAB4EC IEFAB491	IEFAB421	IEFAB4E7 IEFAB4E8
SYSIEFSD Q4	IEFAB421 IEFAB4EC	IEFAB421	IEFAB4DD IEFAB4E7 IEFAB4E8
SYSIEFSD DDRTPUR	IEFAB421 IEFAB4EC IEFAB471 IEFAB491 IEFAB488 IEFAB487		IEFAB4E7 IEFAB4E8
SYSIEFSD DDRDA	IEFAB421 IEFAB4EC IEFAB471 IEFAB491 IEFAB438 IEFAB487		IEFAB4E7 IEFAB4E8
SYSIEFSD ALLOCTP		IEFAB425	IEFAB4E8
SYSZPCCB PCCB	IEFAB4F4 IEFAB4F5 IEFAB469	IEFAB4FE	IEFAB4EA
SYSZTIOT address space identifier and DSAB		IEFDB400	IEFDB402
SYSZVMV UCB address		IEFAB49B	IEFABB410 IEFAB4E8
SYSZVOLS volume serial number	IEFAB4F0 IEFAB4F1 IEFAB421	IEFAB4F0 IEFAB421	IEFAB4DD IEFAB4A8 IEFAB477
SYSZOPEN data set name	IEFDB4A1		IEFBB410

General System Flow

The following diagrams illustrate the general flow of control within OS/VS2.

- Nucleus Initialization
- Master Scheduler Initialization
- Start Primary JES
- Start Initiator
- Job Entry
- Job Execution

Note: Module names that appear in these figures are the names of load modules.

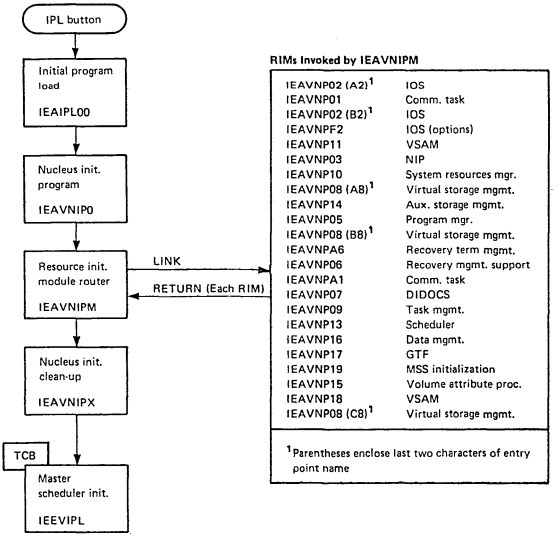
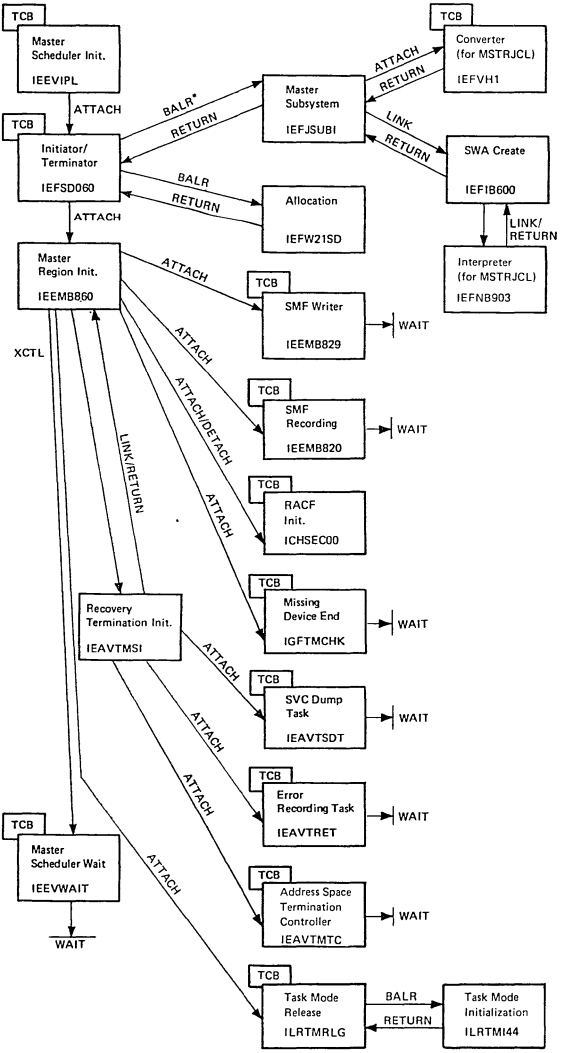
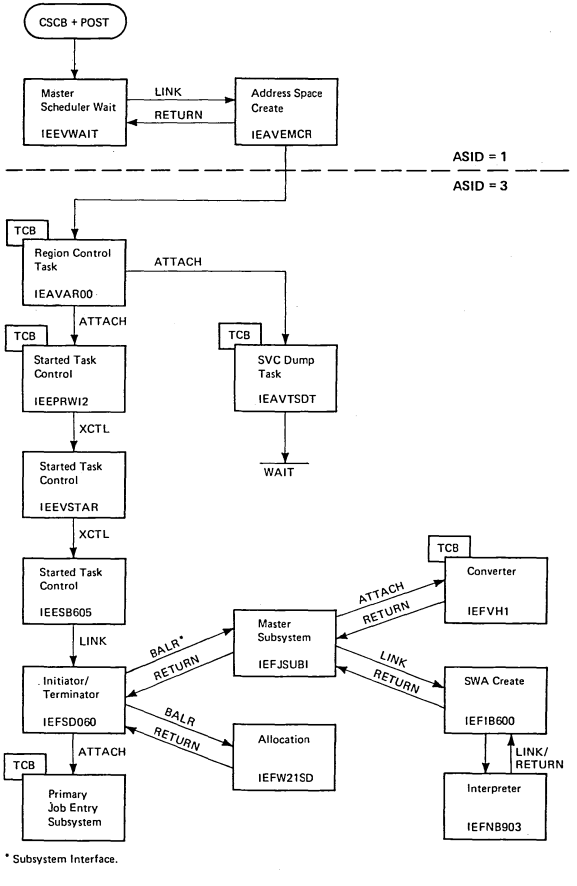


Figure 33. Nucleus Initialization



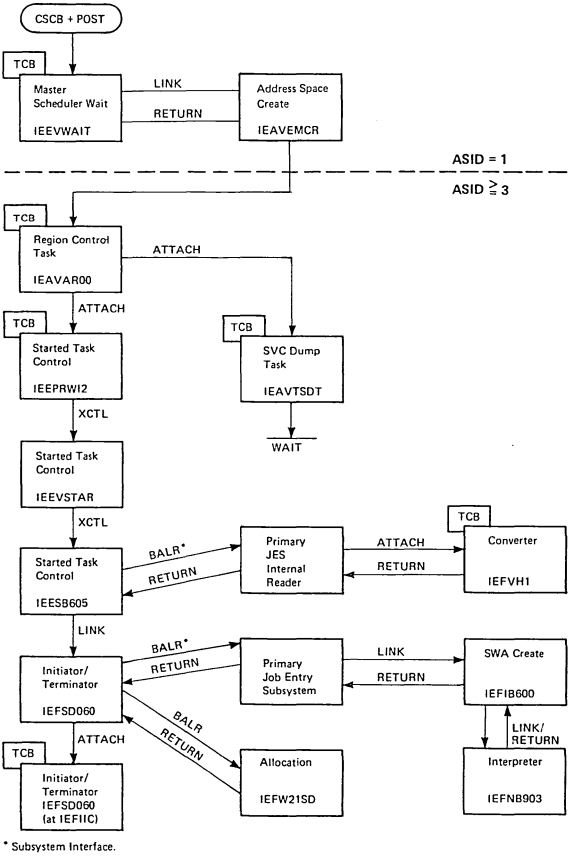
* Subsystem Interface.

Figure 34. Master Scheduler Initialization



* Subsystem Interface.

Figure 35. Start Primary JES



* Subsystem Interface.

Figure 36. Start Initiator

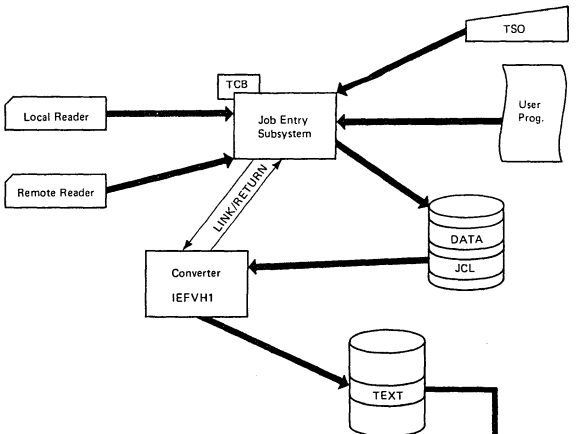


Figure 37. Job Entry

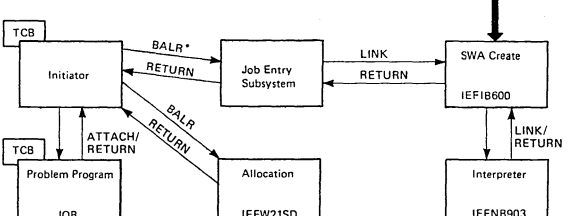


Figure 38. Job Execution

Acronyms

Following is a list of acronyms and their meanings. This list includes a subset of acronyms for all the data areas described in Volume 2 and Volume 3 of this publication, and in the *OS/VS2 Data Areas*, SYB8-0606 (microfiche).

For each data area acronym, the explanation identifies the data area mapping macro and any additional publications (other than SYB8-0606) that apply. For each mapping macro, a listing of the macro definition is available in the system microfiche.

ABDA	Abdump work area. Macro is IHAABDA.
ABDPL	Snap parameter list. Macro is IHAABDPL.
ABEND	Abnormal end of task.
ABP	Actual block processor. Macro is IEZABP.
ACA	ASM control area. Macro is ILRACA.
ACB	Access method control block. Macro is IFGACB.
ACDEB	VTAM data extent block. Macro is ISTACDEB.
ACE	ASM control element. Macro is ILRACE.
ACR	Alternate CPU recovery.
ACT	Account tables.
AFC	Available frame count.
AFM	Allocation function map.
AFMP	Allocation function map.
AFQ	Available frame queue.
AIA	ASM I/O request area. Macro is ILRAIA.
AIB	Application interface block. Macro is TAIBD.
AIT	Algorithm interface tables. Macro is IEFZB426.
ALCA	Allocation communication area. Macro is IEFZB432.
ALCC	Allocate catalog controls. Macro is IEFZB442.
ALCWA	Allocation work area. Macro is IEFZB425.
ALLOCDEF	Device allocation defaults. Macro is IEFZB445.
ALTIOSTB	TIOT status B overlay. Macro is IEFZB449.
AMB	Access method block. Macro is IDAAMB.
AMBL	Access method block list. Macro is IDAAMBL.
AMCBS	Access method control block structure, described in OS/VS2 Catalog Management Logic, SY26-3826. Macro is AMCBS.
AMDDATAX	Mapping of PRDMP input records. Macro is AMDDATA.
AMDSB	Access method data statistics block. Macro is IDAAMDSB.
AMWA	Access method work area.
APF	Authorized program facility.
APG	Automatic priority group.
AQE	Allocate queue element. Macro is IHAAQE.
AQMRB	Allocation queue manager parameter/communication area. Macro is IEFZB427.
ARDB	Address range definition block. Macro is IDAARDB.
ASCB	Address space control block. Macro is IHAASCB.
ASID	Address space identifier.
ASM	Auxiliary storage management.
ASMHD	ASM header. Macro is ILRASMHD.
ASMVT	ASM vector table. Macro is ILRASMVT.
ASPCT	Auxiliary storage page correspondence table. Macro is ILRASPCT.
ASVT	Address space vector table. Macro is IHAASVT.
ASXB	Address space extension block. Macro is IHAASXB.
ATB	Attention table entries.
ATCOM	Allocation/termination communications area.
ATTCH	ATTACH parameter list DSECT. Macro is IEZATTCH.
AVT	TCAM address vector block. Macro is TAVTD.
AWA	Interpreter work area. Macro is IEFVMAWA.
BASEA	Master scheduler resident data area. Macro is IEEBASEA.
BBMB	Background parameter block. Macro is IKJZBBMB.
BCBIR	Broadcast notices directory record. Macro is IKJZT302.
BCMSG	Broadcast notice message record. Macro is IKJZT303.
BCMSG	SYS1.BROADCAST data set.
BEB	Beginning-end block. Macro is IECDBEB.
BECA	EDIT communications area. Macro is IKJEBECA.
BIB	Base information block, described in SY26-3825. Macro is IDABIB.
BRKELEM	Break element. Macro is BRKELEM.
BTU	Basic transmission unit. Macro is TBTUD.

BUFC	Buffer control block. Macro is IDABUFC.
CAESTPA	Common allocation ESTAE exit parameter area. Macro is IEFZB447.
CAFM	Common allocation function map. Macro is IEFZB428.
CAT	Channel availability table. Macro is IECDCAT.
CCA	Catalog Communications area, described in OS/VS2 Catalog Management Logic, SY26-3826. Macro is IGGCCA.
CAXWA	Catalog auxiliary work area, described in OS/VS2 Catalog Management Logic, SY26-3826. Macro is IGGCAXWA.
CC	Condition code.
CCH	Channel check handler.
CCT	SRM CPU management control table. Macro is IRACCT.
CCW	Channel command word for TCAM. Macro is TCCWD.
CDE	Contents directory entry. Macro is IHACDE.
CHKWA	Checkpoint work area. Macro is IEEVCHWA.
CIB	Command input buffer. Macro is IEZCIB.
CICB	JES compatibility interface control block. Macro is IFGJCICB.
CIWA	Common internal work area. Macro is IEACIWA.
CIX	CI SVC exit list. Macro is IHACIX.
CMB	Cluster management block, described in SY26-3825. Macro is IDACMB.
CMS	Cross memory services.
COM	Communications parameter list. Macro is IEZCOM.
COMWA	Converter/interpreter common work area. Macro is IEFCOMWA.
CONTAB	Internal control table for TSO submit command. Macro is IKJEFFCT.
CPA	Channel program area. Macro is IDACPA.
CPA WSHD	Channel program area, working storage header, described in SY26-3825. Macro is IDAWSHD.
CPAB	Cell pool anchor block. Macro is IHACPAB.
CPID	Cell pool identifier.
CPPL	Command processor parameter list. Macro is IKJCPPL.
CPU	Central processing unit.
CQE	Console queue element. Macro is IHACTM.
CRT	Cathode ray tube.
CSA	Common service area.
CSCB	Command scheduling control block. Macro is IEECHAIN.
CSECT	Control section.
CSD	Common system data area. Macro is IHACSD.
CSOA	Command scan output area. Macro is IKJCSOA.
CSPL	Command scan parameter list. Macro is IKJCSPL.
CSW	Channel status word.
CTGOV	VSAM catalog control volume. Macro is IEZCTGCV.
CTGFL	VSAM catalog control field. Macro is IEZCTGFL.
CTGPL	VSAM catalog parameter list. Macro is IEZCTGPL.
CTGVL	VSAM catalog volume list. Macro is IEZCTGVL.
CTGWA	VSAM catalog scheduler work area. Macro is IEZCTGWA.
CUNESTPA	Common unallocation ESTAE exit parameter area. Macro is IEFZB441.
CUNI	Common unallocation interface. Macro is IEFZB439.
CVMAP	MP and K command parameter list. Macro is IEECVMAP.
CVRWA	Converter work area. Macro is IEFCVRWA.
CVT	Communications vector table. Macro is CVT.
CXSA	SVC 72 extended save area. Macro is IHACTM.
DACB	DAIR attribute control block. Macro is IKJDACB.
DAFM	Dynamic allocation function map. Macro is IEFZB407.
DAKEYDIC	Dynamic allocation key dictionary. Macro is IEFZB4D4.
DAKEYTAB	Dynamic allocation key table. Macro is IEFZB4D3.
DAPB00	DAIR entry code 00 parameter list. Macro is IKJDAP00.
DAPB04	DAIR entry code 04 parameter list. Macro is IKJDAP04.
DAPB08	DAIR entry code 08 parameter list. Macro is IKJDAP08.
DAPB0C	DAIR entry code 0C parameter list. Macro is IKJDAP0C.
DAPB10	DAIR entry code 10 parameter list. Macro is IKJDAP10.
DAPB14	DAIR entry code 14 parameter list. Macro is IKJDAP14.
DAPB18	DAIR entry code 18 parameter list. Macro is IKJDAP18.
DAPB1C	DAIR entry code 1C parameter list. Macro is IKJDAP1C.
DAPB24	DAIR entry code 24 parameter list. Macro is IKJDAP24.
DAPB28	DAIR entry code 28 parameter list. Macro is IKJDAP28.
DAPB2C	DAIR entry code 2C parameter list. Macro is IKJDAP2C.
DAPB30	DAIR entry code 30 parameter list. Macro is IKJDAP30.
DAPB34	DAIR entry code 34 parameter list. Macro is IKJDAP34.

DAPL	DAIR parameter list. Macro is IKJDAPL.
DASD	Direct access storage device.
DAT	Dynamic address translation.
DCB	Data control block. Macro is DCBD.
DCM	Display control module.
DD	Data definition.
DDNAME	Data definition name.
DDR	Dynamic device reconfiguration.
DDRCOM	Dynamic device reconfiguration communication table. Macro is IHADDR.
DEB	Data extent block. Macro is IEZDEB.
DEBAP	TCAM application program DEB. Macro is TDEBAPD.
DECB	Data event control block. Macro is IHADECB.
DEVTAB	Device table.
DEVNAMT	Device name table.
DFPB	Default parameter block. Macro is IKJPFDB.
DFPL	Default parameter list. Macro is IKJDFPL.
DIDOCS	Device independent display operator console support.
DIWA	Data insert work area. Macro is IDADIWA.
DMDT	Domain descriptor table. Macro is IRADMDT.
DOM	Delete operator message.
DOMC	Delete operator message control block. Macro is IHADOMC.
DOMPL	Parameter list for DOM - SVC 87. Macro is IHACTM.
DQE	Descriptor queue element. Macro is IHADQE.
DRQ	Data ready queue. Macro is TDRQD.
DSAB	Data set association block. Macro is IHADSAB.
DSABMASK	Data set association block mask. Macro is IEFZB4D8.
DSABQDB	DSAB queue descriptor block. Macro is IEFZB4D5.
DSCB	Data set control block. Macro is IECDSL1.
DSDR	Data set descriptor block.
DSENQT	Data set enq table. Macro is IEFZB902.
DSL	Data set label. No macro.
DSNT	Data set name table. Macro is IEFDSNT.
DSP	Device support processor.
DSPCTMAP	Data set page correspondence table. Macro is IDAVBPM.
DSRFM	Data set reservation/release routine function map. Macro is IEFZB4D6.
DUIDL	User data list. Macro is IKJEFUDL.
DVA	DEVTYP output. Macro is IHADVA.
DVCIDT	Device ID table. Macro is TDVCIDTD.
DVCT	Device characteristics table. Macro is IHADVCT.
DYNESTPA	Dynamic allocation ESTAE exit parameter area. Macro is IEFZB409.
DYNTCFRR	Dynamic allocation TCTIOT FRR parameter area. Macro is IEFZB451.
EBCDIC	Extended binary coded decimal interchange code.
EC	Extended control.
ECB	Event control block. Macro is IHAECB.
ECT	Environment control table. Macro is IKJECT.
EDB	Extent definition block. Macro is IDAEDB.
EDL	Eligible device list. Macro is IEFZB422.
EDT	Eligible devices table. Macro is IEFZB421.
EED	RTM extended error descriptor.
EIL	Communications task event indication list.
EMM	STC STAE exit parameter list. Macro is IEFZB820.
EP	Entry point.
EPA	External parameter area.
EPAL	External parameter area, SWA manager locate mode. Macro is IEFZB505.
EPAM	External parameter area, SWA manager, move mode. Macro is IEF2B506.
EPATH	ASM recovery audit trail area. Macro is ILREPATH.
EPFP	Extended precision floating point.
EREPL	Converter/interpreter ESTAE exit parameter list. Macro is IEFZB9RD.
ESA	Extended save area. Macro is RTMZESA.
ESD	External symbol dictionary.
ESDID	External symbol dictionary ID.
ESTA	SVC 13 use of the SVRB extended STAE parameter list. Macro is IHAESTA.
ETIORB	DSAB/TIOT entry build routine request block. Macro is IEFZB430.
EVNT	EVENT table. Macro is IHAEVNT.

EWA	Common ERP work area. Macro is EWAMAP.
EWD	DASD ERP work area. Macro is EWDMAP.
EXITL	Installation exits for TSO FIB commands. Macro is IKJEFFIE.
EXLSTA	User ACB exit list. Macro is IFGEXLST.
EXLSTD	EXITLIST. Macro is IHAEXLST.
EXTWA	Extend work area. Macro is IECEXTWA.
FBQE	Free block queue element. Macro is IHAFBQE.
FCAUD	Audit trail, open/close executors. Macro is IHAFCAUD.
FCBIM	FCBIMAGE in EXITLIST. Macro is IHAFCBIM.
FCT	Function control table - JES3.
FETWK	FETCH work area. Macro is IHAFETWK.
FFB2	Mapping macro of SVC 100 attach interface. Macro is IKJEFFB2.
FFIB	Mapping macro of SVC 100 interface. Macro is IKJEFFIB.
FLIH	First level interruption handler.
FMCB	VTAM function management control block. Macro is ISTFMCB.
FOE	Fixed ownership element. Macro is IHAFOE.
FPWA	Parse work area. Macro is IKJEFPWA.
FQE	Free queue element. Macro is IHAFQE.
FRR	Functional recovery routine.
FRRS	FRR stack. Macro is IHAFRRS.
FTPT	Communications task's parameter list for it's FRR or ESTAE. Macro is IHACTM.
GDA	Global data area. Macro is IHAGDA.
GDGNT	Generation data group name table IEFZB429.
GFA	General frame allocation.
GFPARMS	Parameter list to TSO general failure service routine. Macro is IKJEFFGF.
GMT	Greenwich mean time.
GSDA	Global system duplex area.
GSMQ	Global service management queue.
GSPL	Global service priority list.
GSR	Global shared resources, a VSAM data set processing option, described in SY26-3825.
GTF	Generalized trace facility.
GTFBCB	GTF buffer control block. Macro is GTFBCB.
GTFBLOK	GTF block. Macro is GTFBLOK.
GTFPCT	GTF primary control block. Macro is GTFPCT.
GTPB	GETLINE. Macro is IKJGTPB.
HC	Hardcopy.
HDR	RTM mapping of the AR-149 header. Macro is IHAHDR.
HISTORY	Internal history table for TSO SUBMIT command. Macro is IKJEFFHT.
HSKESTPA	JFCB housekeeping estae exit parameter area. Macro is IEFZB444.
HSKPWA	JFCB housekeeping work area. Macro is IEFZB437.
ICB	QSAM interruption control block. Macro is IHAICB.
ICQE	Interruption control queue element, described in SY26-3855. Macro is IGGICQE.
ICR	Independent component release.
ICT	SRM I/O management control table. Macro is IRAICT.
ICWA	Index create work area. Macro is IDAICWA.
ID	Identifier.
IDR	Identification record used in applying ICRs and PTFs.
IECALLWA	DADSM allocate work area. Macro is IECALLWA.
IECPRLWA	DADSM partial release work area. Macro is IECPRLWA.
IECSCRWA	DADSM scratch work area. Macro is IECSCRWA.
IEDQTCX	TCAM CVT extension.
IEFPARM	Initiator parameter list.
IEL	Initiator entrance, options exit list. Macro is IEZIEL.
IFDCOM	OLTEP common area. Macro is IEFCOM.
IHSA	Interruption handler save area. Macro is IHAIHSA.
IKJEFLWC	Parameter list for IKJEFLGM. Macro is IKJEFLWC.
IKJEFUAD	User attribute data set. Macro is IKJEFUAD.
IMCB	I/O management control block. Macro is IRAIMCB.
IMWA	Index modification work area. Macro is IDAIMWA.
INITDATA	GTF initialization data area. Macro is INITDATA.
INVT	Initial NIP vector table.
I/O	Input/output.
IOB	Input/output block. Macro is IEZIOB.

IOBEX	Input/output block extension, described in SY26-3855. Macro is IGGIOBEX.
IOBLOCKS	TCAM I/O control blocks. Macro is IOBLOCKS.
IOCX	Input/output supervisor communication extension table.
IOE	PART I/O request element. Macro is ILRIOE.
IOMB	I/O management block. Macro is IDAIOMB.
IOPL	Input/output parameter list. Macro is IKJIOPL.
IOQ	I/O queue element. Macro is IECBIOQ.
IORB	I/O request block. Macro is ILRIORB.
IOS	Input/output supervisor.
IOSB	I/O supervisor block. Macro is IECDIOSB.
IPIB	I/O supervisor purge interface block. Macro is IECDIPIB.
IPL	Initial program loader.
IPLDATA	IPLDATA. Macro is IEAPXNIP.
IPS	Installation performance specification.
IQE	Interruption queue element. Macro is IHAIQE.
IRB	Interruption request block.
IRT	I/O supervisor recovery table. Macro is IECDIRT.
IWA	Interpreter work area. Macro is IEFVMIWA.
IXSPL	Index search parameter list. Macro is IDAIXSPL.
JACT	Job account table.
JCL	Job control language.
JCLS	Job control language string.
JCT	Job control table. Macro is IEFAJCTB.
JES	Job entry subsystem.
JES2	Job entry subsystem 2.
JES3	Job entry subsystem 3.
JESCT	JES control table. Macro is IEFJESCT.
JFCB	Job file control block. Macro is IEFJFCBN.
JFCBE	Job file control block extension for the 3800. Macro is IEFJFCBE.
JFCBX	Job file control block extension. Macro is IEFJFCBX.
JMR	Job management record. Macro is IEFJMR.
JNLPARM	Journal write parameter list. Macro is IEFZB507.
JSCB	Job step control block. Macro is IEZJSCB.
JSEL	Job scheduling entrance list. Macro is IEFVJSEL.
JSOL	Job scheduling options list. Macro is IEFVJSOL.
JSR	Journal service routine.
JSTCB	Job step TCB.
JSUESTPA	Job/step unallocation ESTAE exit parameter area. Macro is IEFZB440.
JSWA	Job scheduling work area.
JSXL	Job scheduling exit list. Macro is IEFVJSXL.
K	1024 bytes.
LCA	Log control area. Macro is IEELCA.
LCB	Teleprocessing line control block. Macro is TLCBD.
LCCA	Logical configuration communication area. Macro is IHALCCA.
LCCAVT	Logical configuration communication area vector table. Macro is IHALCCAT.
LCH	Logical channel queue table. Macro is IECDLCH.
LCT	Linkage control table. Macro is IEFALLCT.
LDA	VSM local data area. Macro is IHALDA.
LGB	Line group block. Macro is TLGBD.
LGE	ASM logic group element. Macro is ILRLGE.
LGN	ASM logical group number.
LGVT	Logical group vector table. Macro is ILRLGVT.
LLE	Load list element. Macro is IHALLE.
LPA	Link pack area.
LPDE	Link pack directory entry. Macro is IHALPDE.
LRB	Logrec buffer. Macro is IHALRB.
LSCT	Logical swap control table.
LSD	List source descriptor. Macro is IKJLSD.
LSMQ	Local service management queue.
LSPL	Local service priority list.
LSQA	Local system queue area.
LWA	Logon work area. Macro is IKJEFLWA.
MB	Type 1 message buffer DSECT. Macro is IHAMB.
MC	Monitor call.
MCA	IEBCOP4 communications area. Macro is IEBMCA.
MCAWSA	Monitor call application work save area. Macro is MCAWSA.

MCCD	Monitor call class directory. Macro is MCCD.
MCCE	Monitor call control element. Macro is MCCE.
MCCLE	Monitor call class element. Macro is MCCLE.
MCEE	Monitor call event element. Macro is MCEE.
MCH	Machine check handler.
MCHEAD	Monitor call routing tables head. Macro is MCHEAD.
MCQE	Monitor call queue element. Macro is MCQE.
MCS	Multiple console support.
MCT	SRM storage management control table. Macro is IRAMCT.
MCRWSA	Monitor call router save area. Macro is MCRWSA.
MEL	Merge entrance list.
MF/1	System activity measurement facility.
MFCOA	MF/1 common options area. Macro is IRBMFCOA.
MFA	Malfunction alert.
MFMVT	MF/1 measurement vector table. Macro is IRBMFMVT.
MFCPT	MF/1 program control table. Macro is IRBMFPCT.
MFPMA	MF/1 program measurement area. Macro is IRBMFPMA.
MIH	Missing interruption handler.
MLWTO	Multiple line write-to-operator.
MMB	Monitor message block. Macro is IEAMMB.
MP	Multiprocessing.
MPL	Monitor parameter list. Macro is IEZMPL.
MQE	Monitor queue element. Macro is IEAMQE.
MSG	Message content block. Macro is IGFMSG.
MSGTABLE	TSO message table parmlist. Macro is IKJEFFMT.
MSRDA	Master scheduler resident data area.
MSS	Mass storage system.
MUG	Multi-unit generic.
MVCA	Mount and verify communication area. Macro is IEFZB433.
MVS	Multiple virtual storage.
MVV	IEHMOVE communications area. Macro is IEHMVV.
MWA	O/C/EOV main work area. Macro is IECDSECT.
NCB	VTAM node control block. Macro is ISTNCB.
NEL	Interpreter entrance list. Macro is IEFNEL.
NIP	Nucleus initialization program.
NIPMNTPL	NIP mount parameter list. Macro is IEAPMNIP.
NIPOPNPL	NIP open parameter list. Macro is IEAPMNIP.
NIPPAHDR	NIP parameter area header. Macro is IEAPPNIP.
NIPPAREA	NIP parameter area. Macro is IEAPPNIP.
NIPPTE	NIP parameter address table entry. Macro is IEAPPNIP.
NIPSCHDL	NIP schedule parameter list. Macro is IEAPMNIP.
NIPSPE	NIP system parameter queue entry. Macro is IEAPMNIP.
NVT	NIP vector table. Macro is IHANVT.
NWTOHDR	NIP write-to-operator message header. Macro is IEAPMNIP.
NWTORLST	NIP WTOR parameter list. Macro is IEAPMNIP.
OLTCB	Online test control block. Macro is OLTCB.
OLTEP	Online test executive program.
OPSVT	System resources manager performance specification vector.
ORE	Operator reply element. Macro is IHAORE.
OS/VS2	Operating System/Virtual Storage 2.
OUCB	SRM user control block. Macro is IRAOUCB.
OUSB	System resources manager user swappable block. Macro is IHAOUSB.
OUXB	System resources manager user extension block. Macro is IHAOUXB.
PAPL	Parse parameter list. Macro is IKJPPL.
PARAM	Initiator parameter list. Macro is IEFZB622.
PARMA	Parse descriptor element. Macro is IKJPARMA.
PARML	Parameter lists to installation exits. Macro is IKJEFFIE.
PARMLIST	Internal parameter list for the TSO CANCEL and STATUS commands. Macro is IKJEFFPT.
PARMTAB	NIP parameter address table. Macro is IEAPPNIP.
PART	Paging activity reference table. Macro is ILRPART.
PAT	Page allocation table. Macro is ILRPAT.
PCB	Page control block. Macro is IHAPCB.
PCBR	Page control block root. Macro is IHAPCBR.
PCCA	Physical configuration communication area. Macro is IHAPCCA.
PCCAVT	Physical configuration communication area vector table. Macro is IHAPCCAT.
PCCB	Private catalog control block. Macro is IEFPCCB.

PCCNTRLS	Private catalog control block routine controls. Macro is IEFZB450.
PCCW	Paging channel command work area. Macro is ILRPCCW.
PCI	Program controlled interruption.
PCT	Performance characteristics table. Macro is ILRPCT.
PDI	Passed data set information. Macro is IEFZB435.
PDS	Partitioned data set directory entry. Macro IHAPDS is for a PDS entry.
PEB	Process element block. Macro is TPEBD.
PECB	Process element control block. Macro is TPECBD.
PEWA	TCAM process entry work area. Macro is TPEWAD.
PFK	Program function key.
PFT	Page frame table.
PFTE	Page frame table entry. Macro is IHAPFTE.
PGPB	PVTGET parameter block. Macro is IKJPGPB.
PGT	Page table.
PGTE	Page table entry. Macro is IHAPGTE.
PICA	SPIE program interruption control area. Macro is IHAPICA.
PIE	SPIE program interruption element. Macro is IHAPIE.
PIRL	Purged I/O restore list. Macro is IECDPIRL.
PLH	VSAM place holder header and place holder entry. Macro is IDAPLH.
PLPA	Pageable link pack area.
PPL	Purge parameter list. Macro is IECBPPL.
PPT	Program properties table.
PQE	Partition queue element. Macro is IHAPQE.
PRB	Program request block.
PRLIST	Permanently resident reserved list.
PSA	Prefixed storage area. Macro is IHAPSA.
PSAB	Protected sphere AMBL block, described in SY26-3825.
PSB	Protected sphere block, described in SY26-3825.
PSCB	Protected step control block. Macro is IKJPSCB.
PSW	Program status word.
PSLIST	Public/storage list.
PTF	Program trouble fix.
PTPB	Putline parameter block. Macro is IKJPTPB.
PVT	Paging vector table. Macro is IHAPVT.
PWA	Processor work area. Macro is IGFPWA.
QCB	Queue control block. Macro is IHAQCB.
QDB	Queue descriptor block. Macro is IHAQDB.
QEL	Queue element. Macro is IHAQEL.
QIO	Queue manager secondary I/O parameter list. Macro is IHAQIO.
QMIOP	Queue manager input/output parameter list. Macro is IEFQMIOP.
QMPA	Queue manager parameter area. Macro is IEFQMNGR.
QSR	Quick start record.
QVOD	Queue verifier output data. Macro is IHAQVOD.
QVPL	Queue verifier parameter list. Macro is IHAQVPL.
R1BC	Broadcast data set record 1. Macro is IKJZT301.
RB	Request block. Macro is IHARB.
RBA	Relative byte address.
RBN	Real block number.
RCA	RSM recovery control area. Macro is IHARCA.
RCB	RTM recording control buffer. Macro is RTMRCB.
RCBSRB	Recording task SRB.
RCT	Region control task, routing control task.
RCTD	Region control task data area. Macro is IEARCTD.
RESPL	Resident module parameter list. Macro is RESPL.
RIM	Resource initialization module.
RLCT	SRM logical channel table. Macro is IRARLCT.
RLD	Relocation dictionary.
RLGB	Relogon buffer. Macro is IKJRLGB.
RMCA	SRM control area. Macro is IRARMCA.
RMCT	SRM control table. Macro is IRARMCT.
RMEP	SRM entry point descriptor. Macro is IRARMEP.
RME	SRM external entry point descriptor table. Macro is IRARMEX.
RMPL	ABEND or RTM resource manager parameter list. Macro is IHARMPL.
RMPT	SRM parameter table. Macro is IRARMPT.
RMS	Recovery management services initialization parameter list. Macro is IGFRMS. Also refers to recovery management services.
RMSB	SRM subroutine vector table. Macro is IRARMSB.

RMTR	System resources manager termination routine.
RPL	VSAM request parameter list. Macro is IFGRPL.
RQE	EXCP request queue element. Macro is IECDRQE.
RRPA	SRM recovery parameter area. Macro is IRARRPA.
RRPL	Recovery OPEN/CLOSE/EOV/DADSM parameter list. Macro is IECDSECS.
RRT	Resource resolution table. Macro is TRRTD.
RSM	Real storage management.
RSMHD	Real storage management header. Macro is IHARSMHD.
RSTWA	Restart work area. Macro is IEEVRSWA.
RSVT	Recovery stack vector table.
RTM	Recovery termination management.
RTCA	Recovery termination communication area. Macro is IHASDWA.
RTCT	Recovery termination control table. Macro is IHARTCT.
RTMCB	Recovery termination management control block.
RTM2WA	RTM2 work area. Macro is IHARTM2A.
RTTE	Rotate table entry.
RTVT	Rotate vector table.
RT1W	RTM1 work area. Macro is IHART1W.
RVT	Recovery management vector table. Macro is IHARVT.
RWA	Recovery work area. Macro is IGFRWA.
S/370	System/370.
SALLOC	Storage allocation.
SAMB	Sequential access method block, described in SY26-3855. Macro is IGGSAMB.
SART	ASM swap activity reference table. Macro is ILRSART.
SAST	Subsystem allocation sequence table. Macro is IEFJSAST.
SAT	ASM swap allocation table. Macro is ILRSAT.
SCA	SPIE control area. Macro is IHASCA.
SCB	STAE control block. Macro is IHASCB.
SCCW	ASM swap channel command work area. Macro is ILRSCCW.
SCRA	Supervisor control recovery area. Macro is IHASCRA.
SCT	Step control table. Macro is IEFASCTB.
SCVT	Secondary communication vector table. Macro is IHASCVT.
SDCT	ASM swap device characteristics table. Macro is ILRSDCT.
SDT	Start descriptor table.
SDUMP	SVC dump parameter list. Macro is IHASDUMP. Also refers to the SVC dump routine.
SDWA	System diagnostic work area. Macro is IHASDWA.
SECCORE	Security work area. Macro is IECDSECS.
SGT	Segment table.
SGTE	Segment table entry. Macro is IHASGTE.
SIC	System-initiated cancel.
SIOT	Step input/output table. Macro is IEFASIOT.
SIRB	System interruption request block.
SLIH	Second level interruption handler.
SLOT	Scheduler look-up table.
SMCA	System management facilities control area. Macro is IEESMCA.
SMF	System management facilities.
SMP	System modification program.
SNAP	Snap parameter list. Macro is IHASNAP. Also refers to the SNAP routine.
SPCT	RSM swap communication table. Macro is IHASPCT.
SPE	System parameter element.
SPIE	Specify program interruption element.
SPL	Service priority list. Macro is IHASPL.
SPP	SETPRT parameter list. Macro is IHASPP.
SPQE	Subpool queue element. Macro is IHASPQE.
SQA	System queue area.
SRB	Service request block. Macro is IHASRB.
SQS	System queue space.
SRM	System resources manager
SSARB	Subsystem allocation request block. Macro is IEFSSARB.
SSCR	Subsystem checkpoint record. Macro is IHJSSCR.
SSCVT	Subsystem communications vector table. Macro is IEFJSCVT.
SSERRBLK	Subsystem error block. Macro is IEFZB454.
SSIB	Subsystem identification block. Macro is IEFJSSIB.
SSMSGBLK	Subsystem message block. Macro is IEFZB455.
SSOB	Subsystem options block. Macro is IEFJSSOB.

SSRB	Suspended SRB save area. Macro is IHASSRB.
SSVT	Subsystem vector table. Macro is IEFJSSVT.
SSWA	Subsystem scheduler work area. Macro is IEFJSSWA.
STAE	Set task asynchronous exit.
STAES	STAE parameter table. Macro is STAES.
STAX	Set terminal attention exit.
STC	Started task control.
STCB	Subtask control block. Macro is TSTCBD.
STCINRDR	Started task control internal reader.
STEPL	Initiator STAE exit parameter list. Macro is IEFZB622.
STGST	MF/1 global storage table. Macro is IRBSTGST.
STMMV	MFROUTER measurement vector table. Macro is IRBSTMMV.
STOWPARM	STOW parameter list. Macro is IHASTOW.
STPB	Stack parameter block. Macro is IKJSTPB.
STPL	Stack parameter list. Macro is IKJSTPL.
STPRT	MF/1 program resource table. Macro is IRBSTPRT.
STRVT	MF/1 resource vector table. Macro is IRBSTRVT.
STSCT	MF/1 supervisor state control table. Macro is IRBSTSCT.
STSGT	MF/1 storage resource table. Macro is IRBSTSGT.
STSMA	MF/1 supervisor state measurements area. Macro is IRBSTSMA.
SVA	SWA virtual address.
SVC	SVC table entry. Macro is IHASVC. Refers to the supervisor call instruction.
SVRB	Supervisor request block.
SVT	Supervisor vector table.
SWA	Scheduler work area, subpools 236, 237.
SWAE	STC parameter list. Macro is IEEZB801.
SWAIC	SWA manager interface controls. Macro is IEFZB436.
SYSGEN	System generation.
S99PARMS	Dynamic allocation (SVC 99) parameter list. Macro is IEFZB4D0.
TABL	Data set table entry. Macro is IHJDSTAB.
TAIE	Terminal attention interruption element. Macro is IKJTAIE.
TAXE	Terminal attention exit element. Macro is IKJTAXE.
TCAM	Telecommunications access method.
TCAS	Terminal control address space.
TCAST	TCAS table. Macro is IKTTCAST.
TCB	Task control block. Macro is IKJTCB.
TCCW	Translation control block. Macro is IECDTCCW.
TCKPD	Checkpoint work area. Macro is TCKPD.
TCOMTAB	Test communication table. Macro is TCOMTAB.
TCPBD	TCAM channel program block. Macro is TCPBD.
TCPBD	TCAM channel program block for the 3330. Macro is TCPBD.
TCT	SMF timing control table. Macro is IEFTCT.
TCTIOT	Timing control table I/O table.
TCW	TOD clock work area.
TDATAD	TCAM disk data record area. Macro is TDATAD.
TDCM	Pageable DCM's. Macro is IEETDCM.
TDEB	Teleprocessing data extent block. Macro is TDEBD.
TDISPD	TCAM dispatcher DSECT. Macro is TDISPD.
TECB	TCAM test event control block. Macro is TTECBD.
TEXTUNIT	Dynamic allocation test unit. Macro is IEFZB4D1.
TFRRPARM	Timer functional recovery routine parameter list.
THB	MCH threshold block. Macro is IGFTHB.
TIOB	Teleprocessing I/O block. Macro is TIOBD.
TIOC	Terminal input-output coordinator.
TIOCBUF	TIOC buffer prefix. Macro is IKJTIOCB.
TIOCRPT	TIOC reference pointer table. Macro is IKJTIOCP.
TIOT	Task input/output table. Macro is IEFTIOT1.
TIM	Terminal input manager.
TLB	Translation lookaside buffer.
TMPWA	Terminal monitor program work area. Macro is IKJTMPWA.
TMRB	TIOT manager request block. Macro is IEFZB424.
TNT	TCAM terminal name table. Macro is TTNTD.
TOD	Time-of-day.
TOM	Terminal output manager.
TOPCAVTD	Operator control address vector table. Macro is TOPCAVTD.
TOPCED	Operator control element. Macro is TOPCED.
TP	Teleprocessing.

TPC	Timer supervision work area. Macro is IEAVVTPC.
TPCB	TCAM process control block. Macro is TPCBD.
TPL	TEST parameter list. Macro is IKJTPL.
TPRFD	TCAM buffer prefix. Macro is TPRFD.
TQCB	TCAM queue control block. Macro is TQCBD.
TQE	Timer queue element. Macro is IHATQE.
TRECB	TCAM resource element control block. Macro is TRECBD.
TRHDR	System trace header. Macro is IEAPXNIP.
TRM	TCAM terminal table entry. Macro is TTRMD.
TSB	Terminal status block. Macro is IKJTSB.
TSBX	TSB extension. Macro is IKJTSB.
TSCB	TCAM station control block. Macro is TSCBD.
TSENQ	TSO ENQ work area. Macro is IECDSECS.
TSGP	Time slice group table entry.
TSO	Time sharing option.
TSOINRDR	Time sharing option internal reader.
TSPT	Time slice pattern table.
TSTCWORK	TEST work area. Macro is TSTCWORK.
TTCB	TCAM task control block. Macro is TTCBD.
TTCXD	TCAM CVT extension. Macro is TTCXD.
TVCS	TSO/VTAM CSA area. Macro is IKTTVCS.
TVT	
TVWA	TSO/VTAM work area. Macro is IKTTVWA.
TWAR	TCAS work area. Macro is IKTCASWA.
TXTFT	Internal text format. Macro is IEFTXTFT.
UADS	TSO user attribute data set.
UCB	Unit control block. Macro is IEFUCBOB.
UCD	Data management and I/O supervisor control blocks. Macro is IEEUCDX.
UCM	Communications task unit control module. Macro is IEECUCM.
UCME	Unit control module entry.
UCMI	Unit control module identifier.
UNALCC	Unallocate catalog controls. Macro is IEFZB443.
UPCON	IEBUPDTE communications area. Macro is IEBUPCON.
UPT	TSO user profile table. Macro is IKJEUPT.
USDIR	TSO broadcast mail directory record. Macro is IKJZT304.
USERLAB	User label work area. Macro is IECDSECS.
USERTOT	User totaling facility save and work area. Macro is IECDSECS.
USMSG	Broadcast mail message record. Macro is IKJZT305.
UTILWORK	EDIT access method work area. Macro is IKJEBEUW.
V=R	Virtual equals real.
V=V	Virtual equals virtual.
VAT	Virtual address table.
VBN	Virtual block number.
VBP	Virtual block processor.
VBPH	Data set page correspondence table header. Macro is IDAVBPH.
VBPL	Logical group chain block. Macro is IDAVBPL.
VBPPL	VBP parameter list. Macro is IDDVBPPL.
VCB	Virtual I/O control block. Macro is IHAVCB.
VDSCB	Virtual data set control block. Macro is IDDVDSCB.
VGTT	VSAM global termination table, described in SY26-3825. Macro is IDAVGTT.
VIO	Virtual input/output.
VM&V	Volume mount and verify.
VMT	Volume mount table, described in SY26-3825. Macro is IDAVMT.
VMVESTPA	Volume mount and verify ESTAE exit and FRR parameter area. Macro is IEFZB452.
VMVRB	Volume mount and verify request block. Macro is IEFZB431.
VOLLABEL	Volume labels.
VOLSERNO	Volume serial number.
VRWPQEL	V=R wait post queue element. Macro is IHAWQPEL.
VSAM	Virtual storage access method.
VSL	RSM virtual sub-area list. Macro is IHAVSL.
VSM	Virtual storage management.
VSRT	VSAM shared resource table, described in SY26-3825. Macro is IDAVSRT.
VS/2	See OS/VS2.
VTAM	Virtual telecommunications access method.
VTIOC	VTAM terminal I/O coordinator.

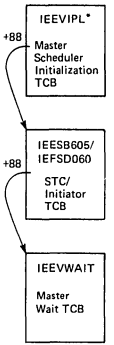
VTOC	Volume table of contents.
VTRACK	Virtual track buffer (also known as VIO buffer and window). Macro is IDDTRACK.
VUNT	Volunit table entry. Macro is IEFZB423.
VUT	Volume unload table. Macro is IEFZB438.
WAMT	Work load activity measurement table. Macro is IRAWAMT.
WICB	VIO control block. Macro is IDDWICB.
WMPGV	SRM performance group vector table.
WMST	SRM workload manager specifications table. Macro is IRAWMST.
WORKAREA	OS/VS catalog management workarea. Macro is WORKAREA.
WPL	WTO/WTOR/MLWTO/WTP parameter list definition. Macro is IEZWPL.
WPRB	Wait post request block. Macro is IHAWPRB.
WQE	Write-to-operator queue element. Macro is IHAWQE.
WSAG	Global work/save vector table.
WSAVT	Work/save area vector tables, described in Volume 2 and SY88-0606. Macro is IHAWSAVT.
WSCT	SRM workload manager control table.
WSHD	Working storage header, described in SY26-3825. Macro is IDAWSHD.
WTG	Where-to-go table. Macro is IECDSECS.
WTO	Write-to-operator.
WTOR	Write-to-operator with reply.
WTP	Write-to-programmer.
WWB	Write-to-operator wait block. Macro is IHACTM.
XPT	External page table.
XPTE	External page table entry. Macro is IHAXPTE.
XSA	SVC 34 extended save area. Macro is IEEXSA.
XTLST	Extent list. Macro is IHAXTLST.
XV	SVC 35 extended save area. Macro is IHACTM.
YSTAK	FRR stack attributes. Macro is IHAYSTAK.
ZB502	SWA block prefix. Macro is IEFZB502.
ZB831	Parameter list for IKJCE831. Macro is IKJZB831.

Section 6: Control Block Chains

This section illustrates the logical relationships of major system data areas (control blocks).

Control Block Chains

The following diagrams illustrate the logical relationships of major system control blocks.



*Contained in SYSGEN load module IEAVBK

Subtasks of Master Wait Task

IEEMB829 – SMF Writer

IEEMB820 – SMF Init.[1]

IGFTMCHK – MIH

RTM Tasks:

IEAVRTRET – Error Recording

IEAVTSDT – SVC Dump

IEAVTMTC – Address Space Termination

IEEMB803 – System Log Init.

IEEVEMCR – START JES[1]

IEExxxxx – Other Command Processor[1]

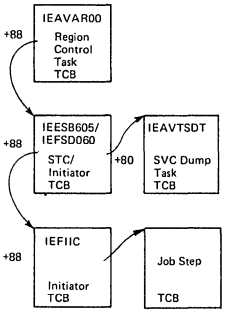
[1]Temporary Task

Notes:

TCB +80 hex (TCBNTC) points to the TCB for the task that was previously attached by this task's originating task.

TCB +88 hex (TCBLTC) points to the TCB for this task's last-created subtask.

Figure 39. TCB Structures – Master Scheduler

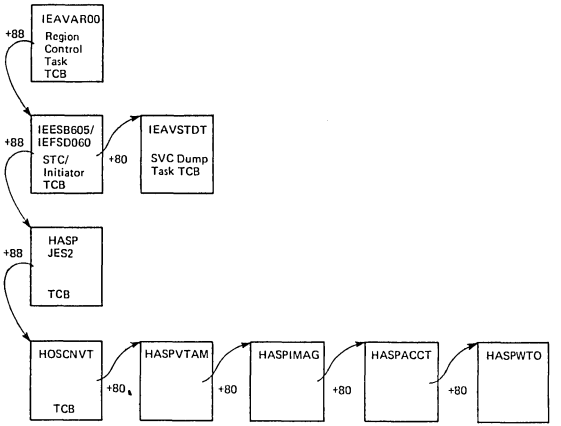


Notes:

TCB +80 hex (TCBNTC) points to the TCB for the task that was previously attached by this task's originating task.

TCB +88 hex (TCBLTC) points to the TCB for this task's last-created subtask.

Figure 40. TCB Structures – Initiator



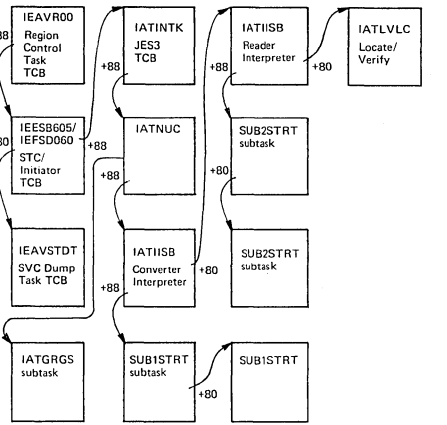
Notes:

TCB +80 hex (TCBNTC) points to the TCB for the task that was previously attached by this task's originating task.

TCB +88 hex (TCBLTC) points to the TCB for this task's last-created subtask.

The HASPVTAM subtask is present if at least one logical line has been defined.

Figure 41. TCB Structures – JES2

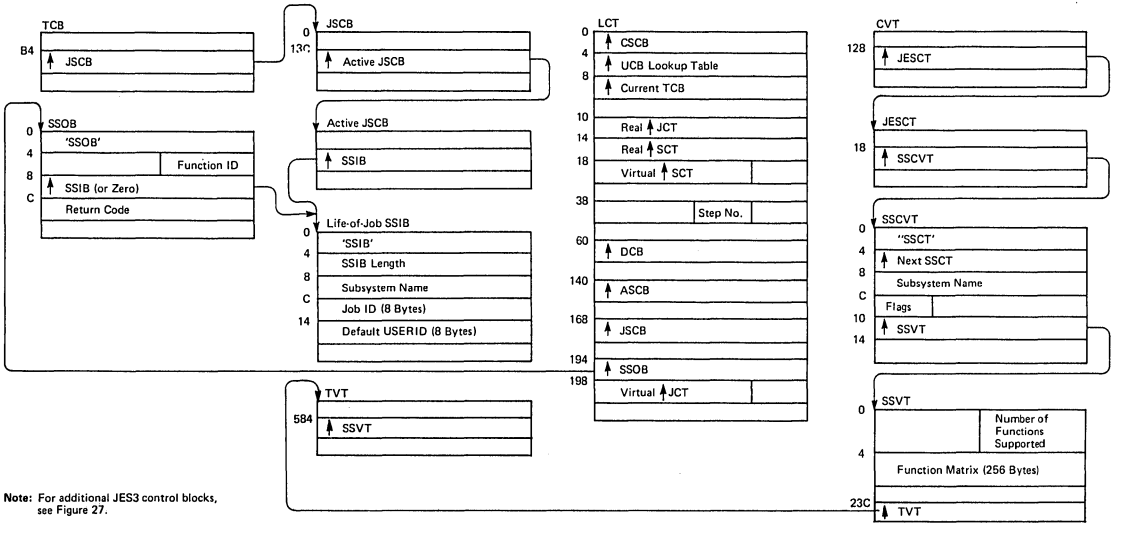


Notes:

TCB +80 hex (TCBNTC) points to the TCB for the task that was previously attached by this task's originating task.

TCB +88 hex (TCBLTC) points to the TCB for this task's last-created subtask.

Figure 42. TCB Structures – JES3



Note: For additional JES3 control blocks, see Figure 27.

Figure 43. Subsystem Interface Control Blbcks

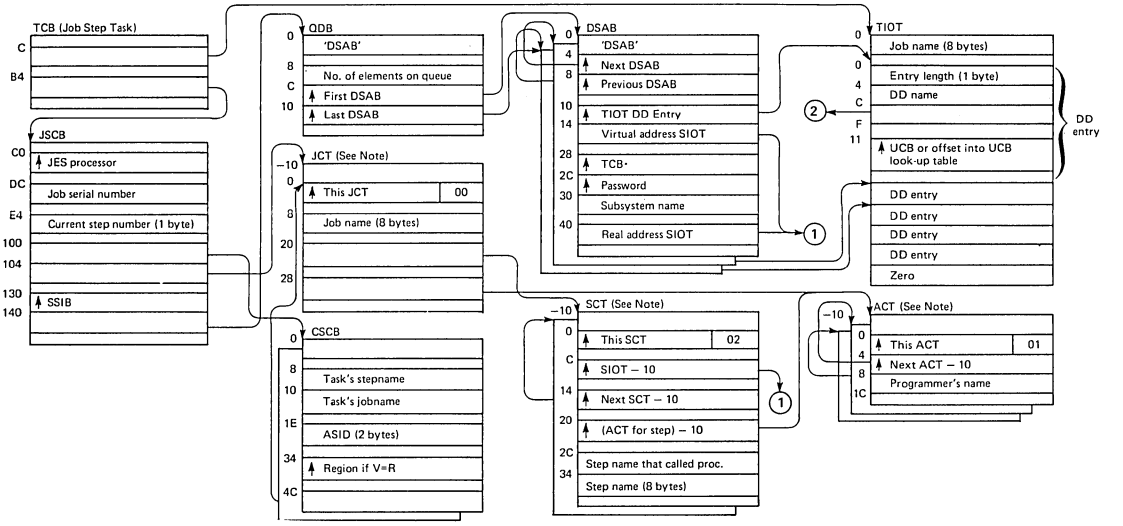


Figure 44. Job Scheduling Control Blocks (Part 1 of 2)

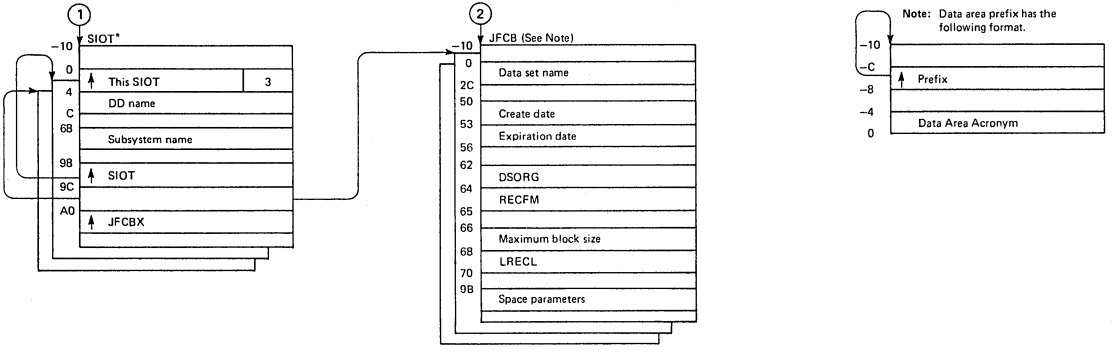


Figure 44. Job Scheduling Control Blocks (Part 2 of 2)

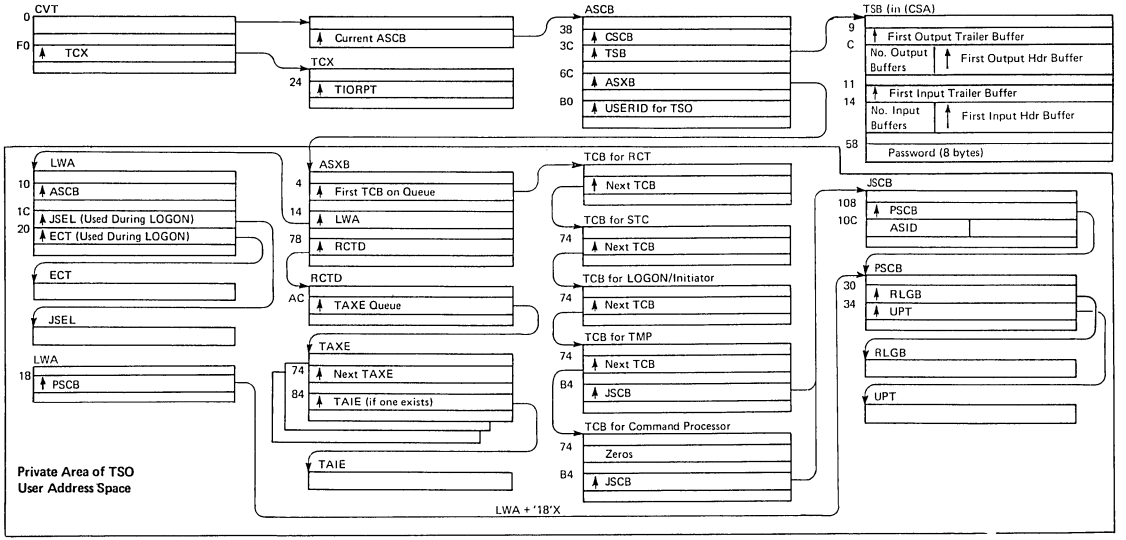


Figure 45. TSO/TCAM Control Blocks

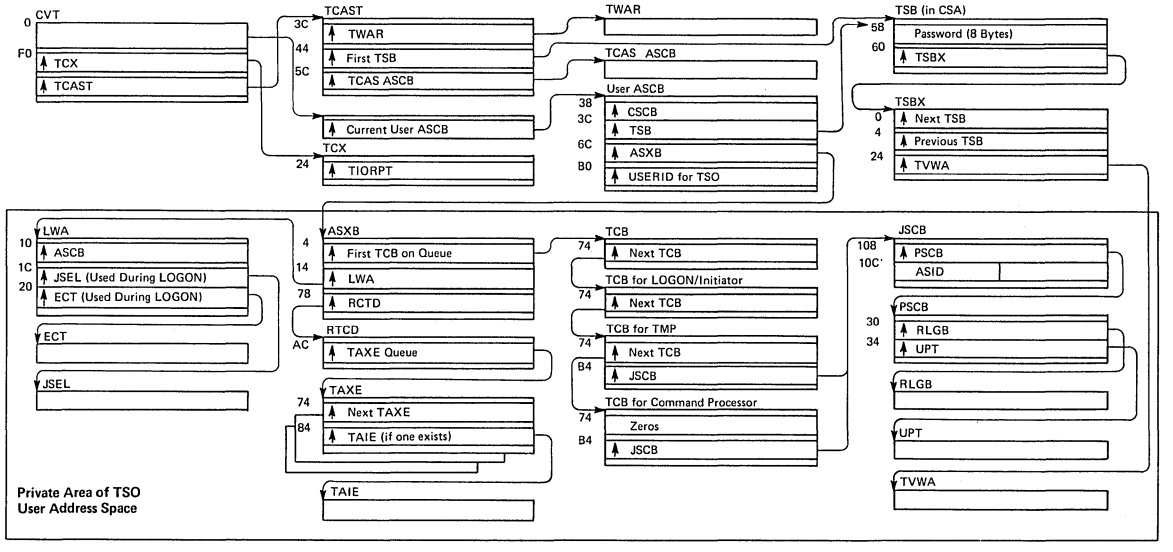


Figure 46. TSO/VTAM Control Blocks

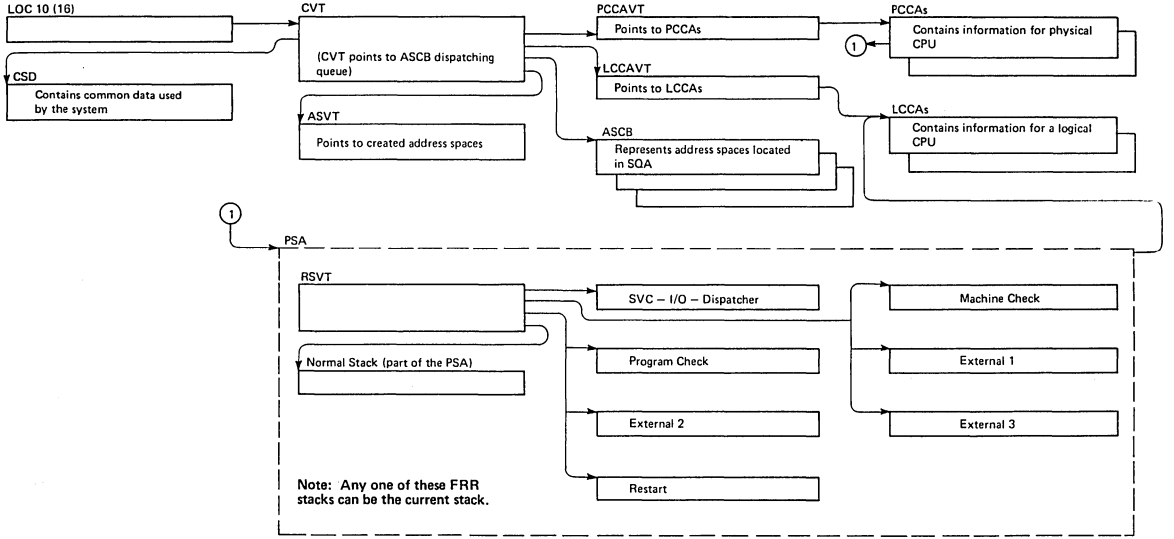


Figure 47. Supervisor-Control Control Blocks – Overview (Part 1 of 2)

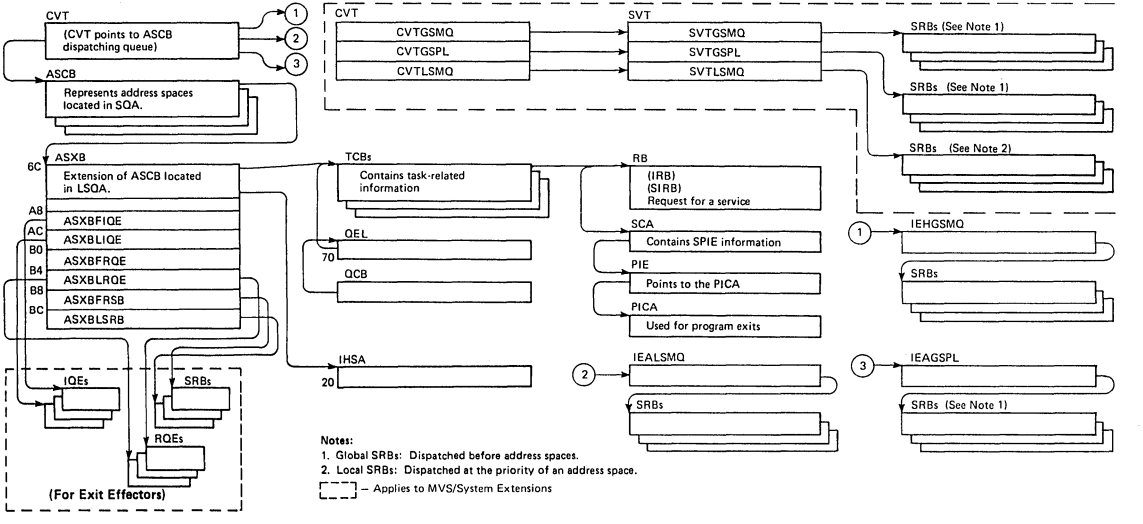


Figure 47. Supervisor-Control Control Blocks – Overview (Part 2 of 2)

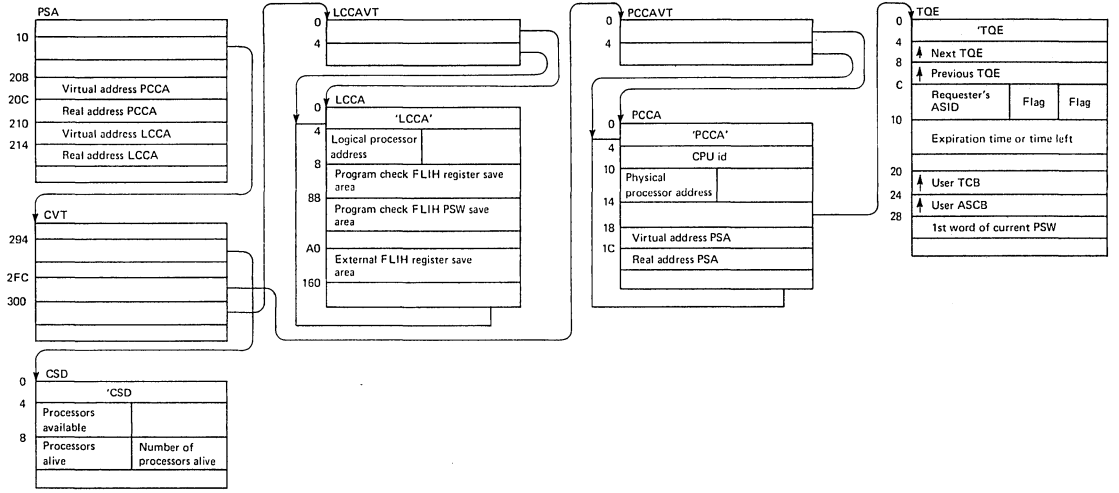
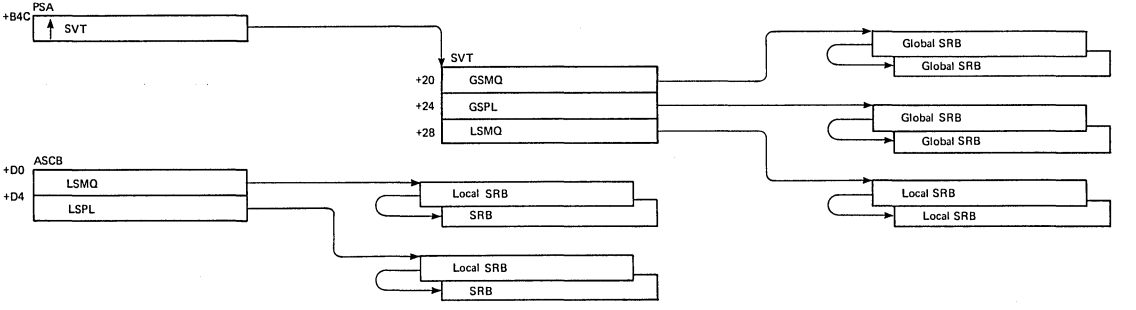


Figure 48. Supervisor-Control Control Blocks – Processor Control Blocks



Note:
The LSMQ in the SVT is maintained for compatability. The MVS System Extensions SRB schedule function places local SRBs on the address space LSMQ. All queues are single headed and each contains system and non-quiescable level SRBs.

Figure 49. Supervisor-Control Control Blocks – Dispatching Queues: GSMQ, LSMQ, GSPL (Applies to MVS/System Extensions)

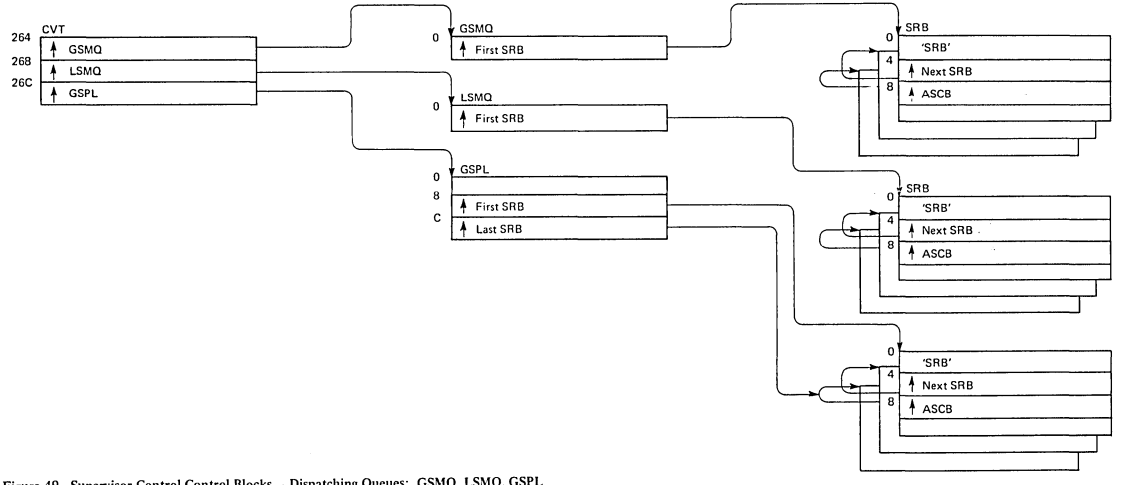


Figure 49. Supervisor-Control Control Blocks – Dispatching Queues: GSMQ, LSMQ, GSPL

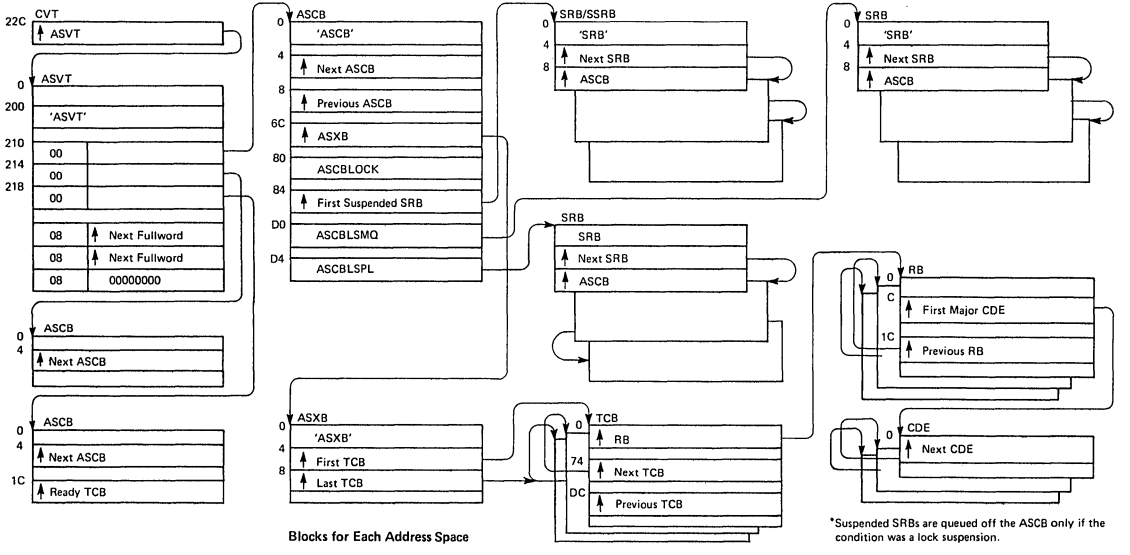


Figure 50. Supervisor-Control Control Blocks – Dispatching Queues: ASCB, TCB, LSPL (Applies to MVS/System Extensions)

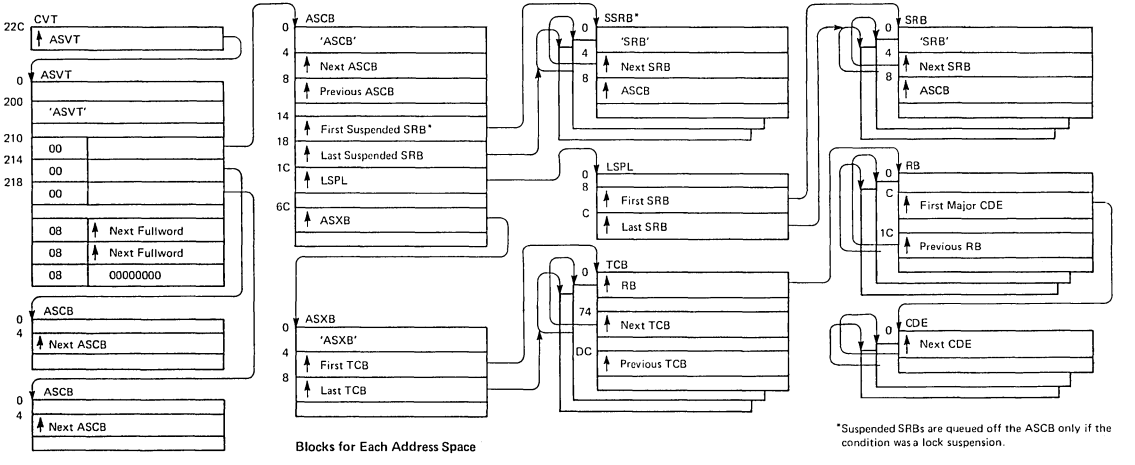
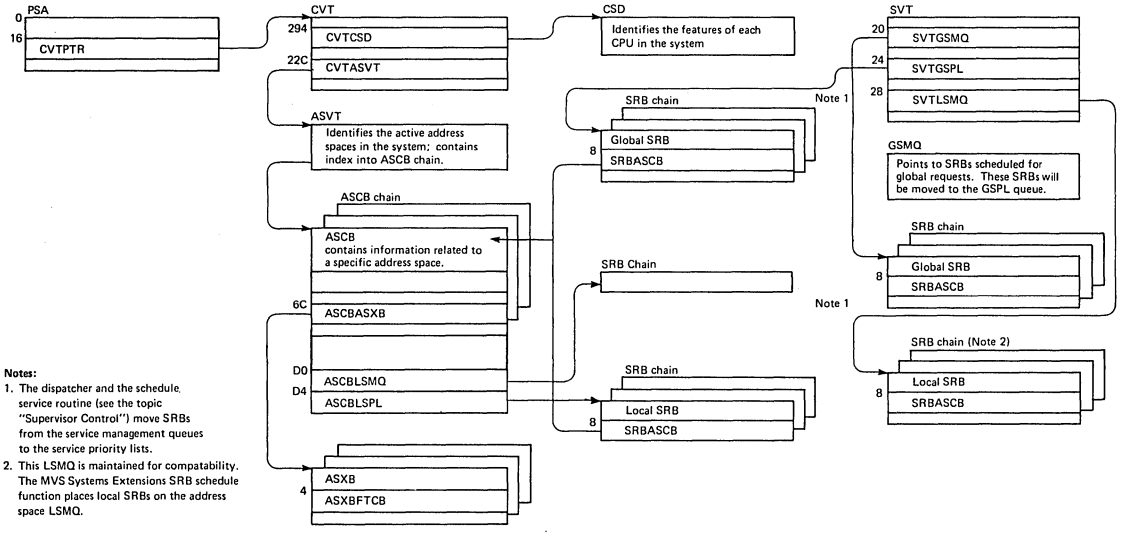


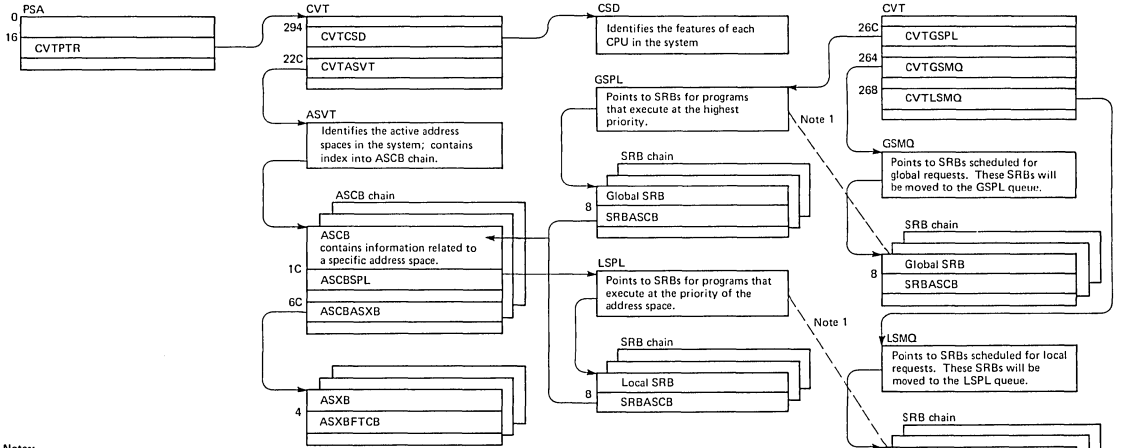
Figure 50. Supervisor-Control Control Blocks – Dispatching Queues: ASCB, TCB, LSPL



Notes:

1. The dispatcher and the schedule service routine (see the topic "Supervisor Control") move SRBs from the service management queues to the service priority lists.
2. This LSMQ is maintained for compatability. The MVS Systems Extensions SRB schedule function places local SRBs on the address space LSMQ.

Figure 51. Task Management Control Blocks – Overview (Part 1 of 2) (Applies to MVS/System Extensions)



Notes:

1. The dispatcher and the schedule service routine (see the topic "Supervisor Control") move SRBs from the service management queues to the service priority lists.

Figure 51. Task Management Control Blocks -- Overview (Part 1 of 2)

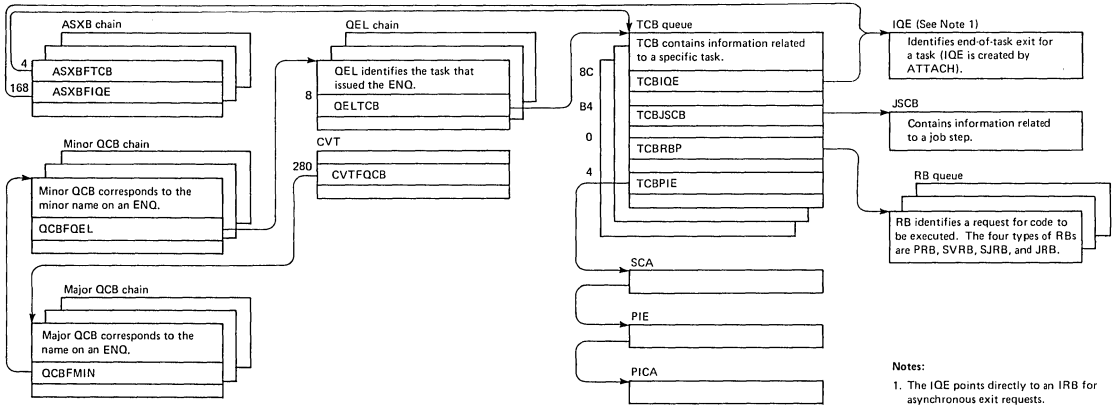


Figure 51. Task Management Control Blocks – Overview (Part 2 of 2)

ENQ Control Blocks:

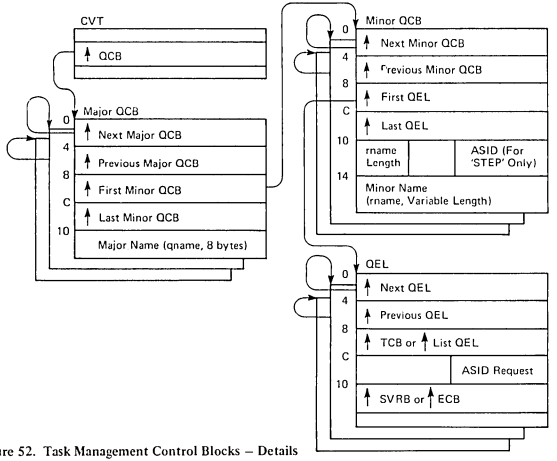
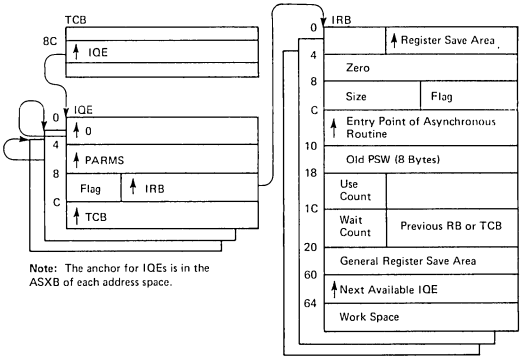


Figure 52. Task Management Control Blocks – Details

Asynchronous Exit Control Blocks:



Note: The anchor for IQEs is in the ASXB of each address space.

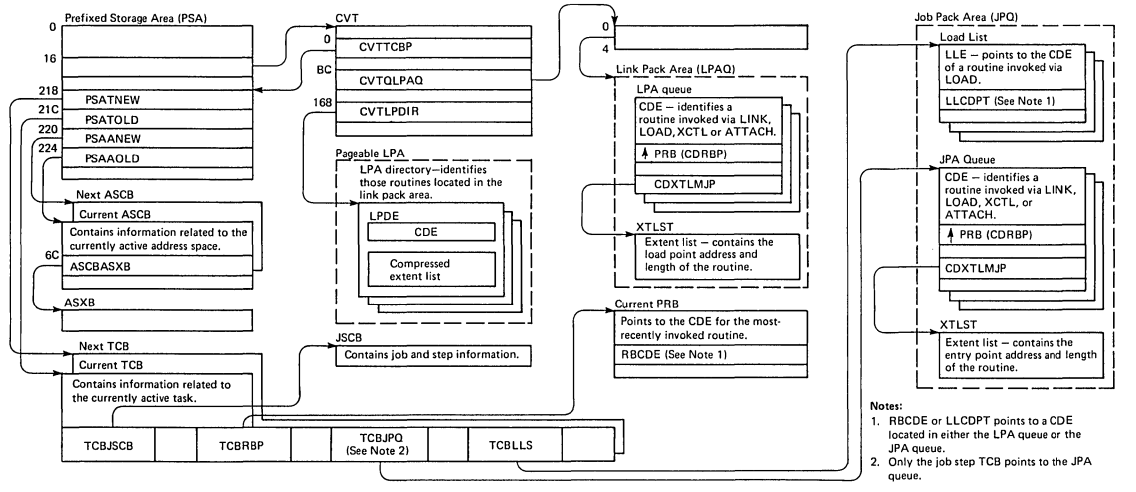


Figure 53. Program Management Control Blocks – Overview

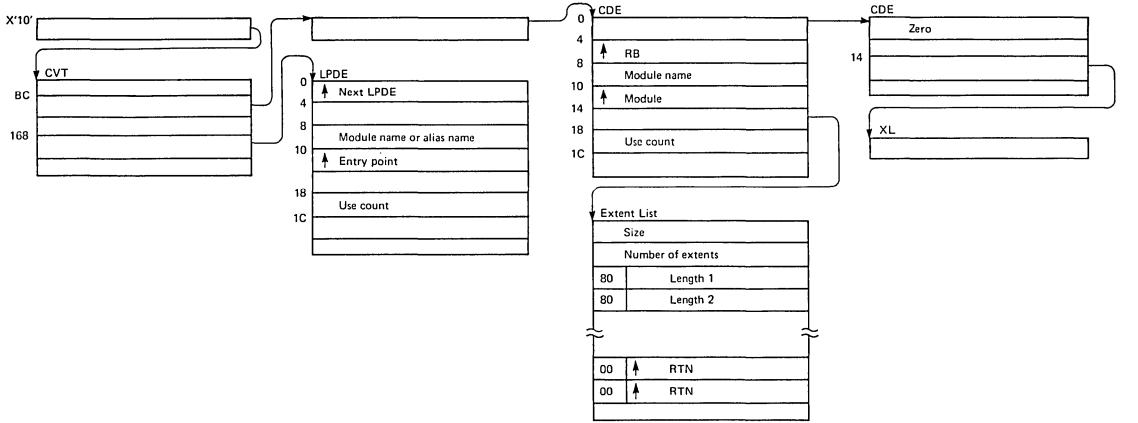


Figure 54. Program Management Control Blocks – Link Pack Area Queue

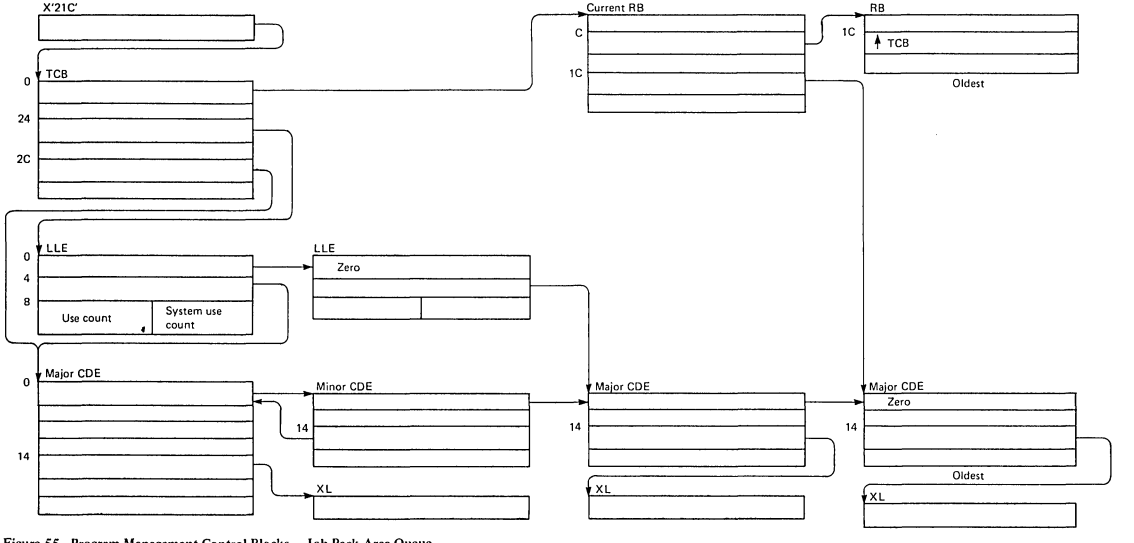


Figure 55. Program Management Control Blocks – Job Pack Area Queue

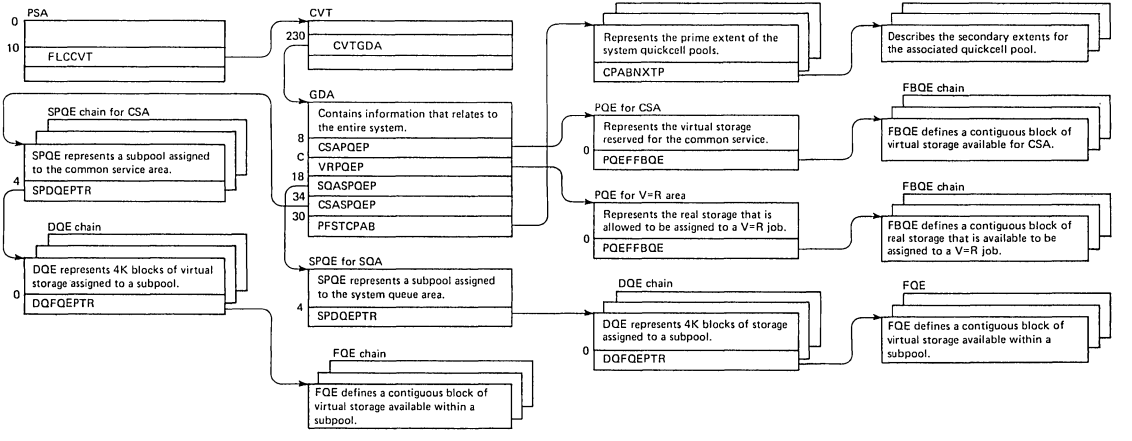


Figure 56. Virtual Storage Management Control Blocks — Overview (Part 1 of 2)

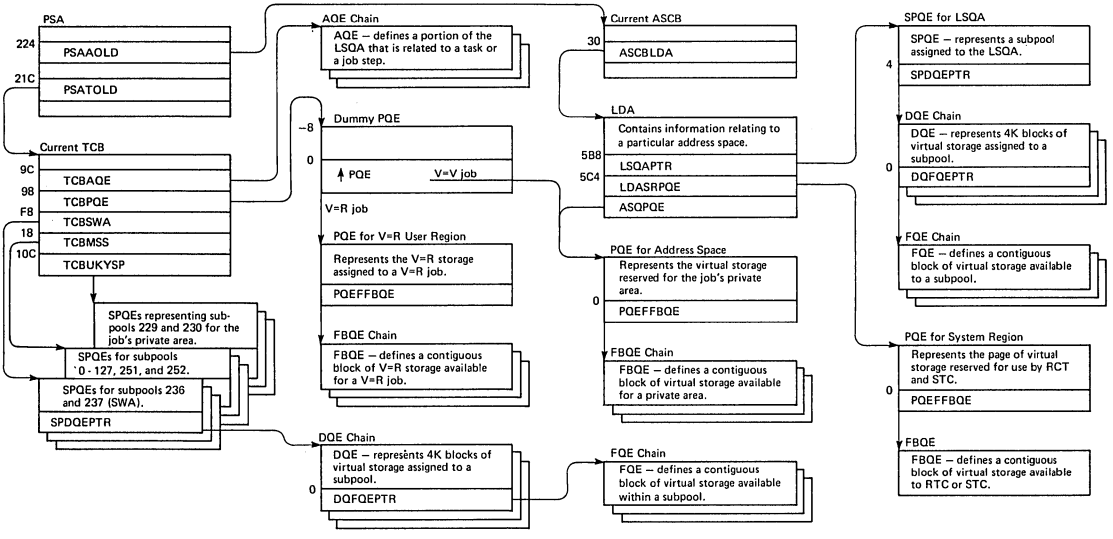


Figure 56. Virtual Storage Management Control Blocks – Overview (Part 2 of 2)

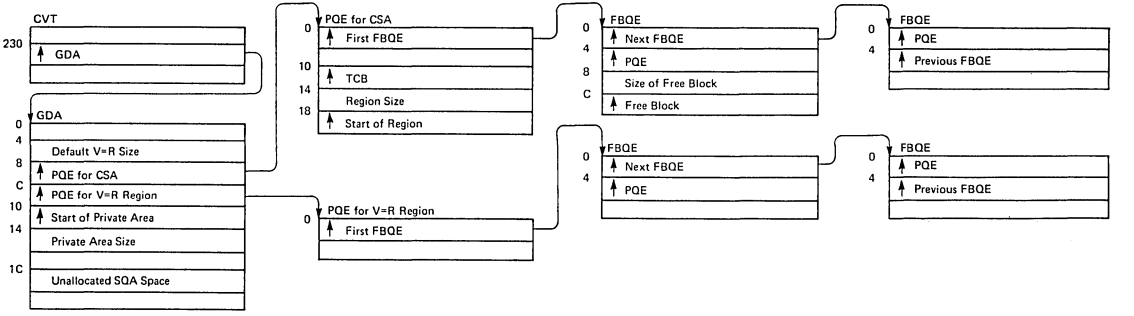


Figure 57. Virtual Storage Management Control Blocks – Global PQE Chain (V=V)

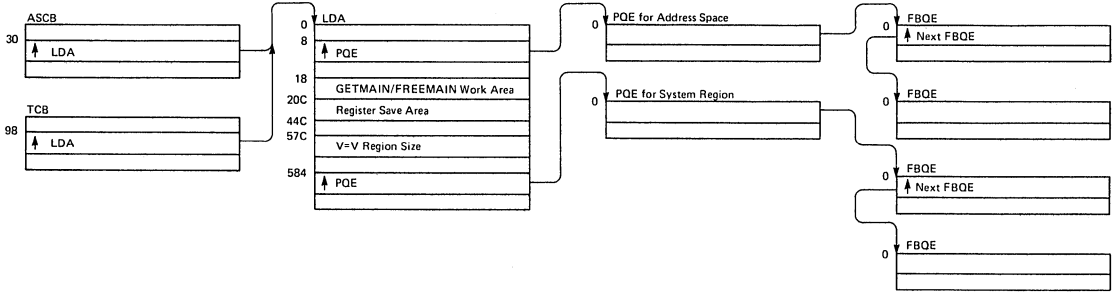


Figure 58. Virtual Storage Management Control Blocks – Local PQE Chain (V=V)

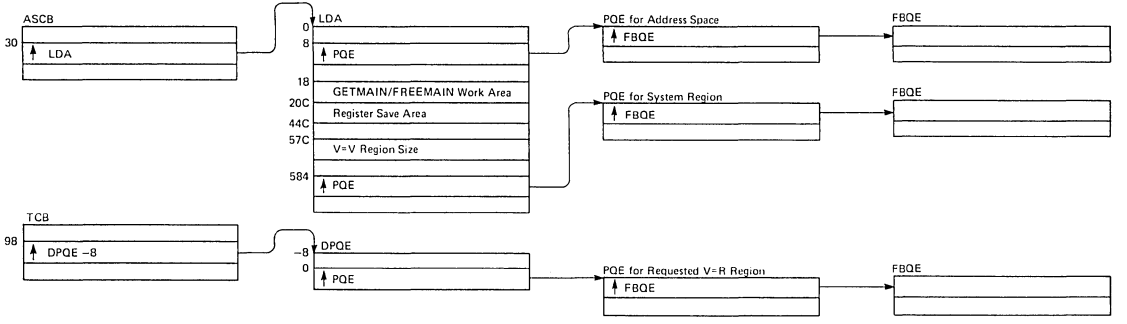


Figure 59. Virtual Storage Management Control Blocks – Local PQE Chain (V=R)

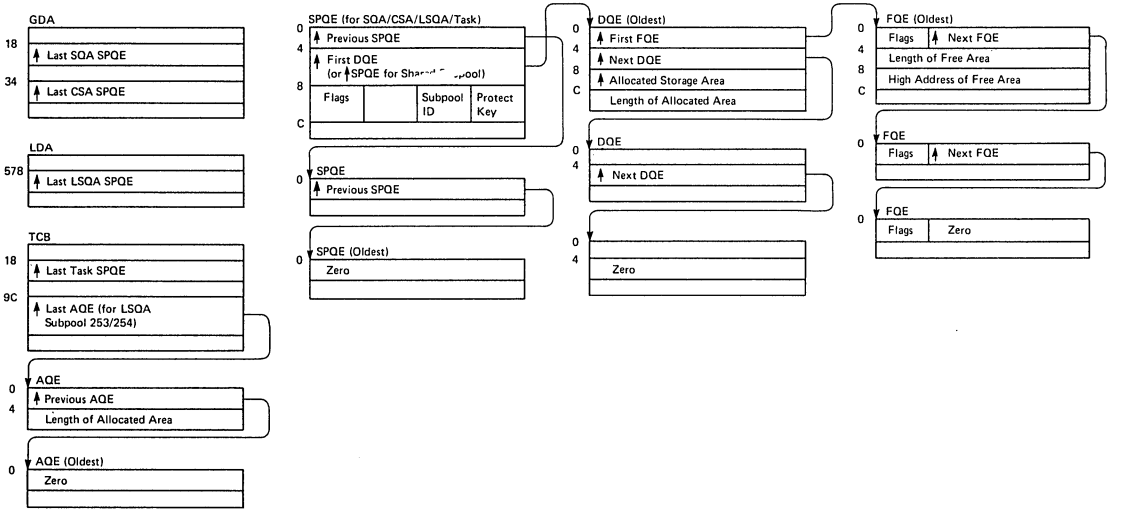


Figure 60. Virtual Storage Management Control Blocks – Subpool Queues

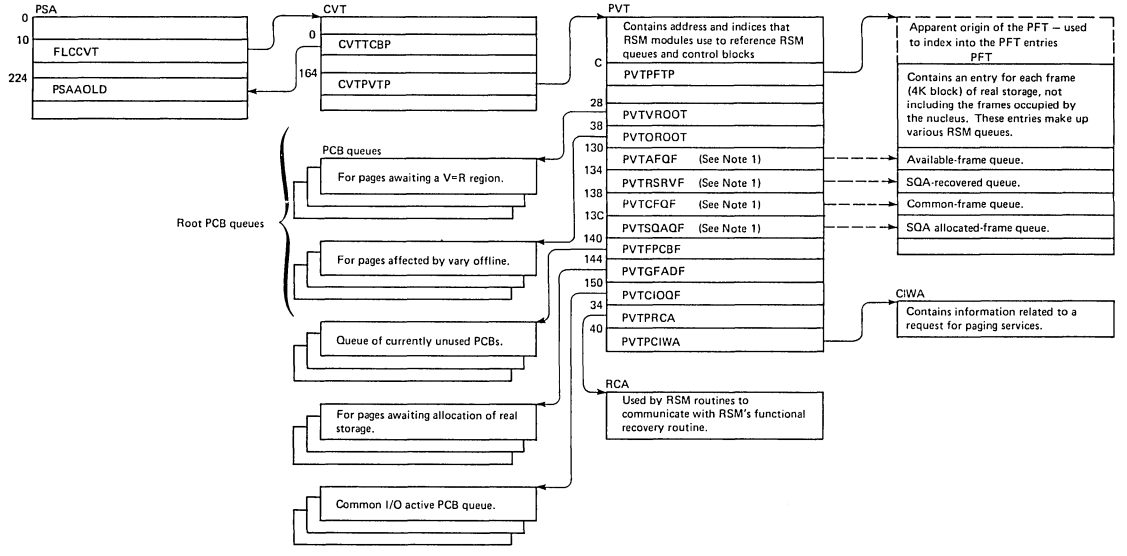


Figure 61. Real-Storage Management Control Blocks – Overview (Part 1 of 2)

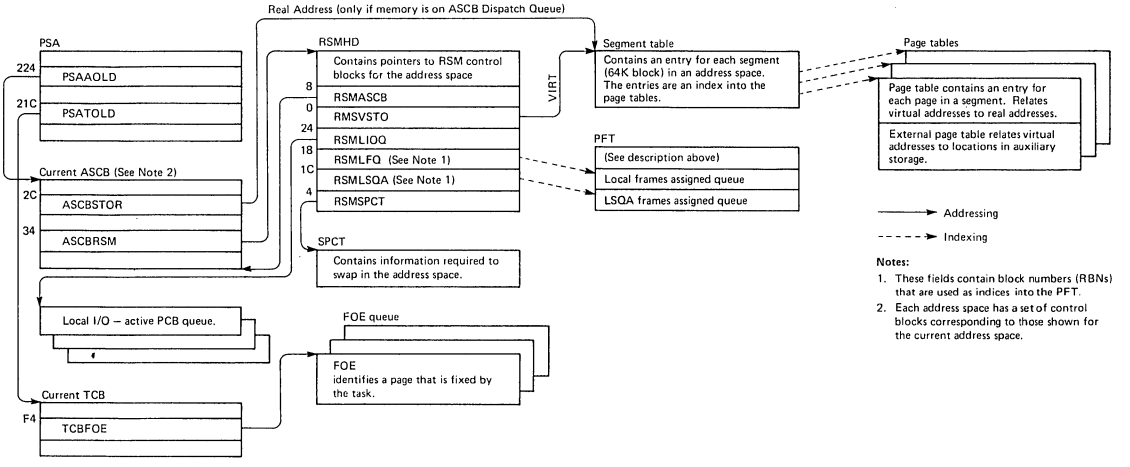


Figure 61. Real-Storage Management Control Blocks – Overview (Part 2 of 2)

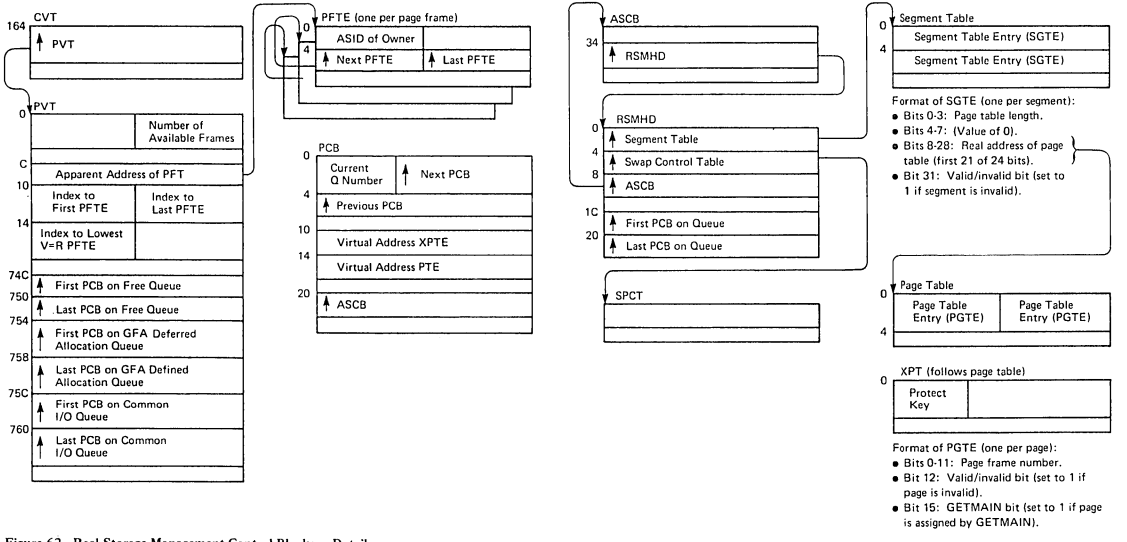


Figure 62. Real-Storage Management Control Blocks – Details

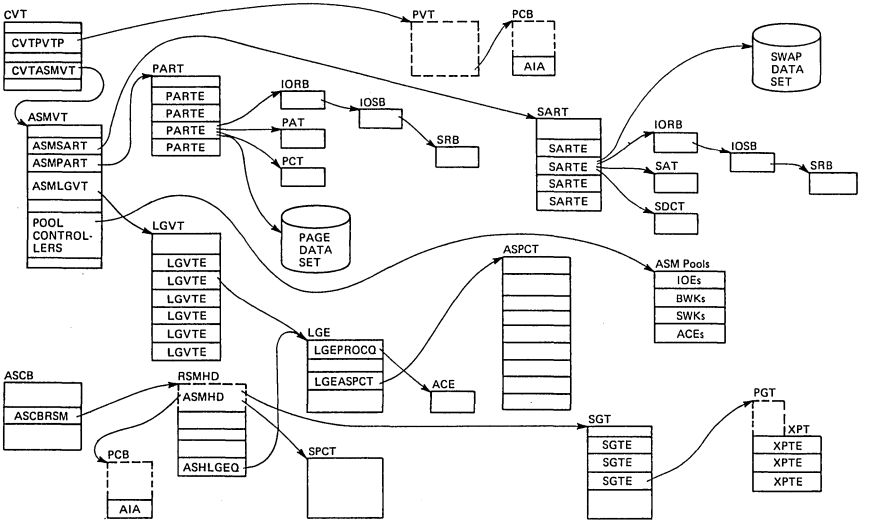


Figure 63. Auxiliary Storage Management Control Blocks

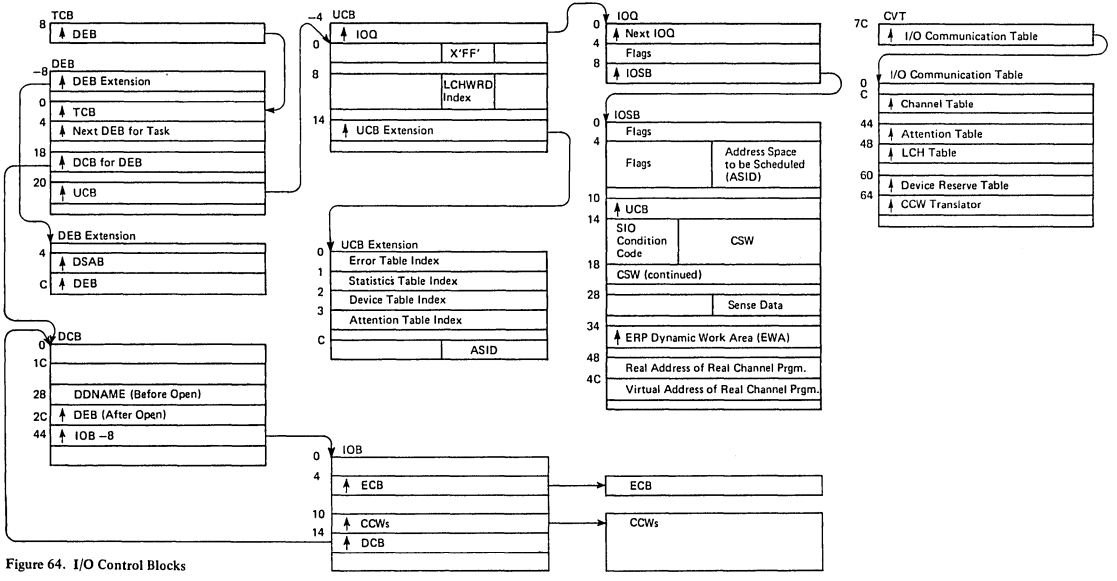


Figure 64. I/O Control Blocks

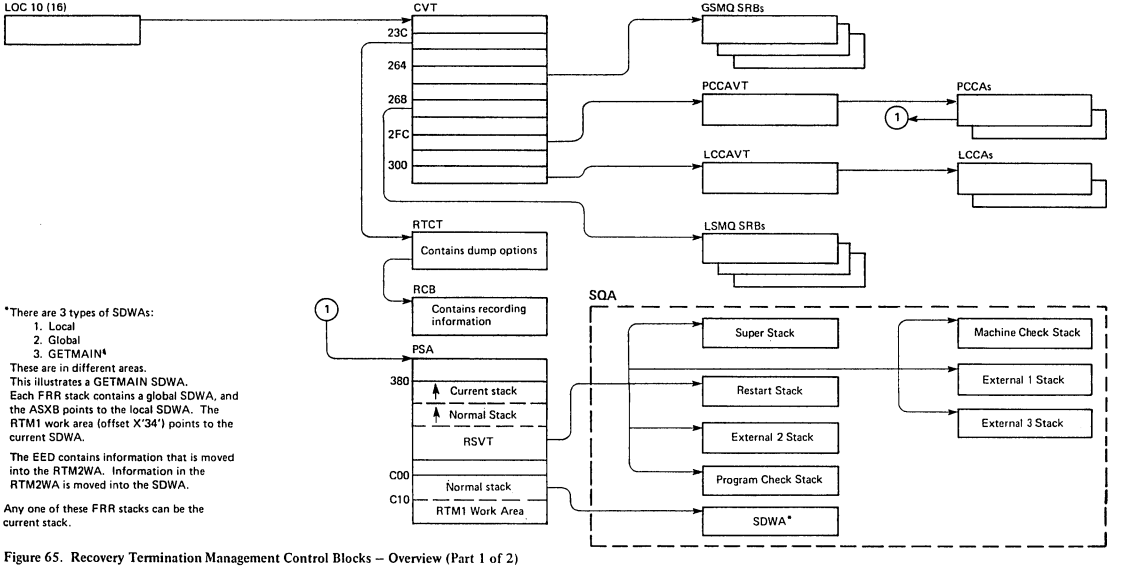
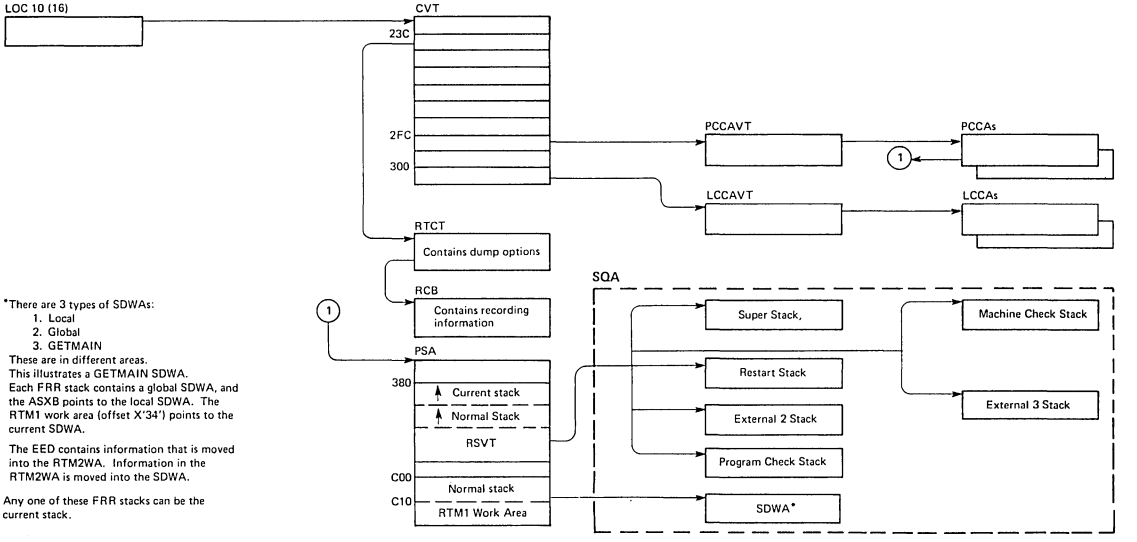


Figure 65. Recovery Termination Management Control Blocks – Overview (Part 1 of 2)



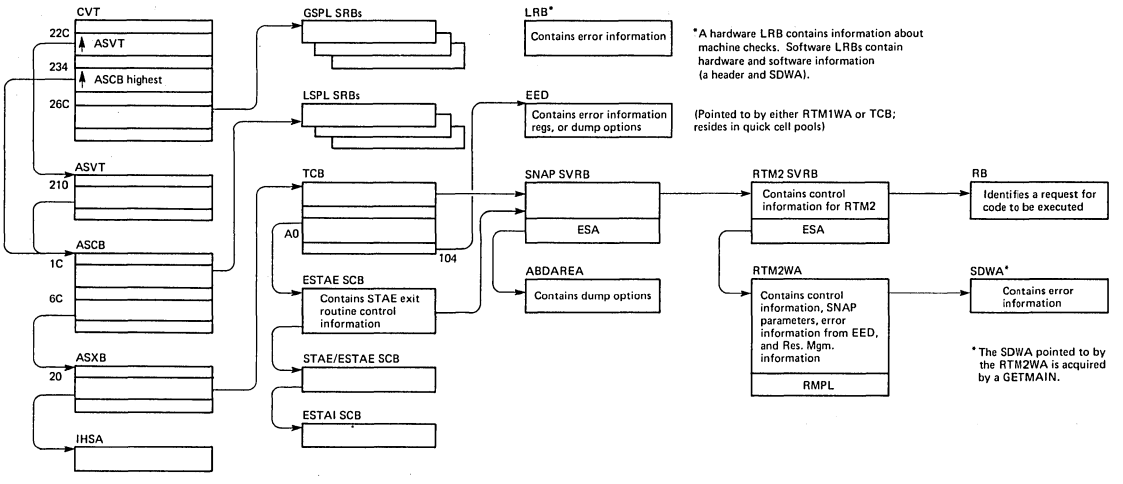
*There are 3 types of SDWAs:
1. Local
2. Global
3. GETMAIN

These are in different areas.
This illustrates a GETMAIN SDWA.
Each FRR stack contains a global SDWA, and the ASXB points to the local SDWA. The RTM1 work area (offset X'34') points to the current SDWA.

The EED contains information that is moved into the RTM2WA. Information in the RTM2WA is moved into the SDWA.

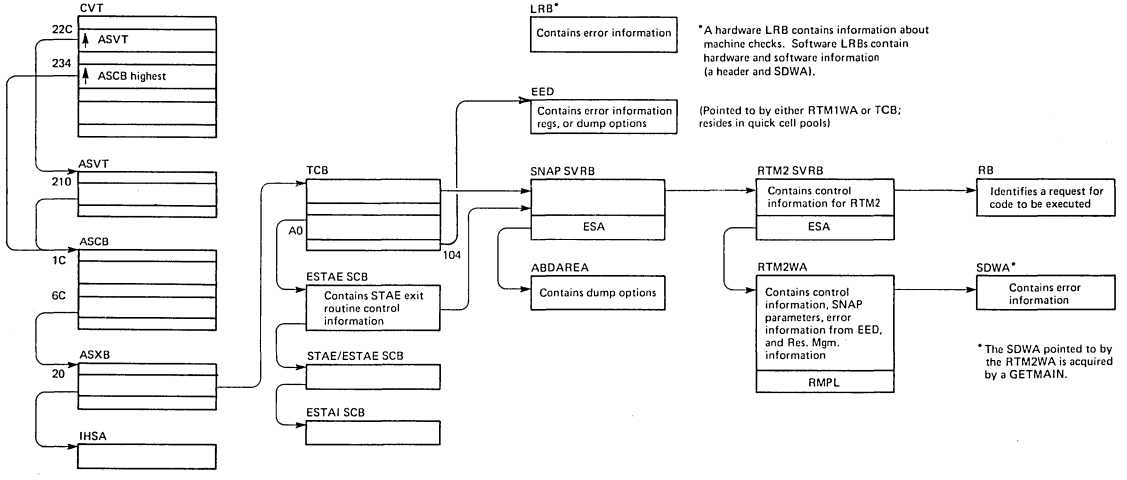
Any one of these FRR stacks can be the current stack.

Figure 65. Recovery Termination Management Control Blocks – Overview (Part 1 of 2) (Applies to MVS/System Extensions)



Note: The EED contains information that is moved into the RTM2WA. Information in the RTM2WA is moved into the SDWA.

Figure 65. Recovery Termination Management Control Blocks – Overview (Part 2 of 2)



Note: The EED contains information that is moved into the RTM2WA. Information in the RTM2WA is moved into the SDWA.

Figure 65. Recovery Termination Management Control Blocks – Overview (Part 2 of 2) (Applies to MVS/System Extensions)

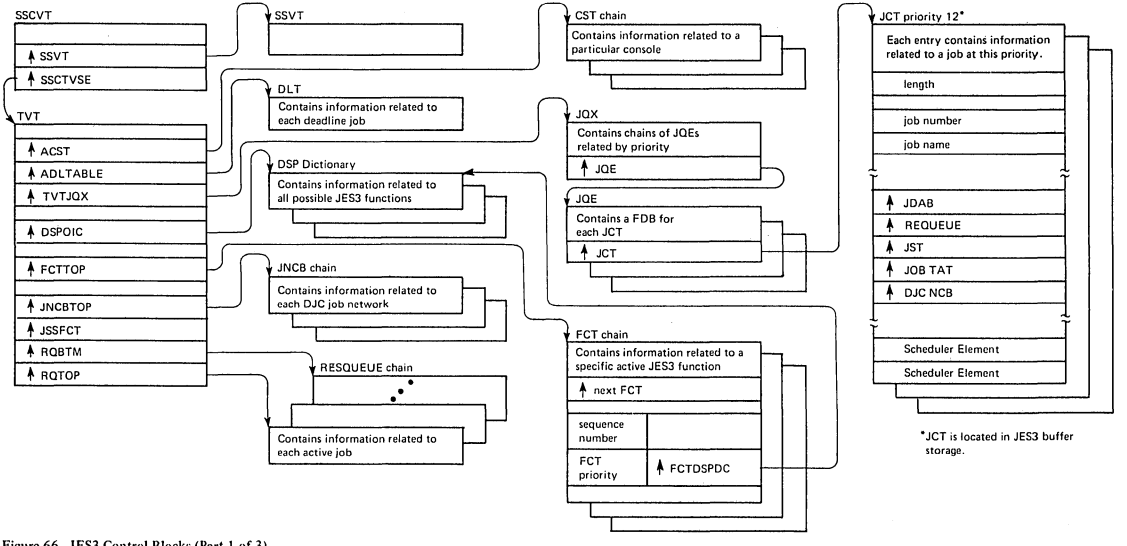


Figure 66. JES3 Control Blocks (Part 1 of 3)

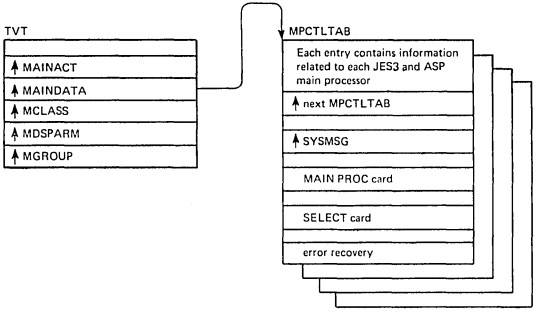


Figure 66. JES3 Control Blocks (Part 2 of 3)

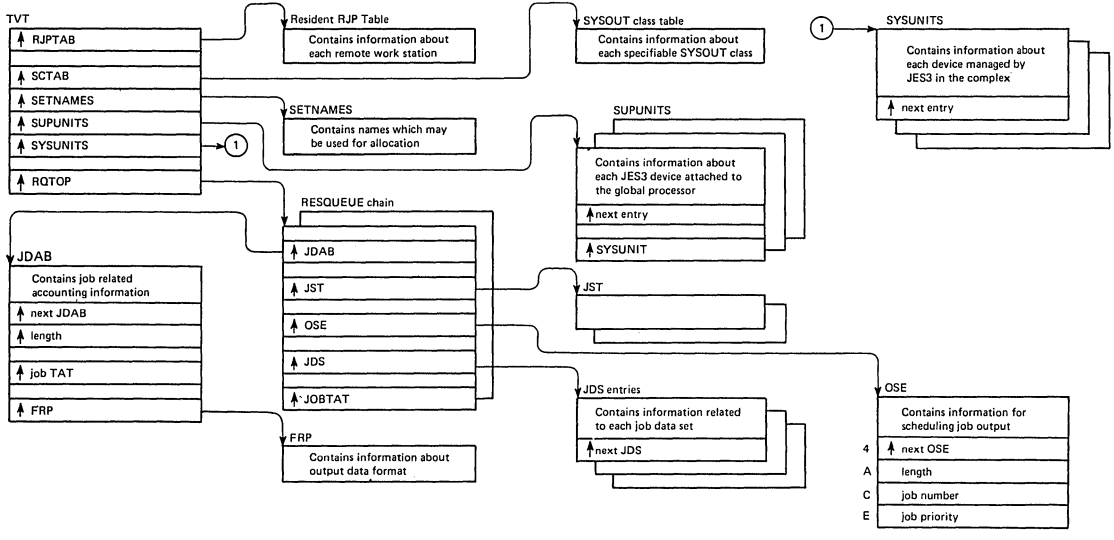
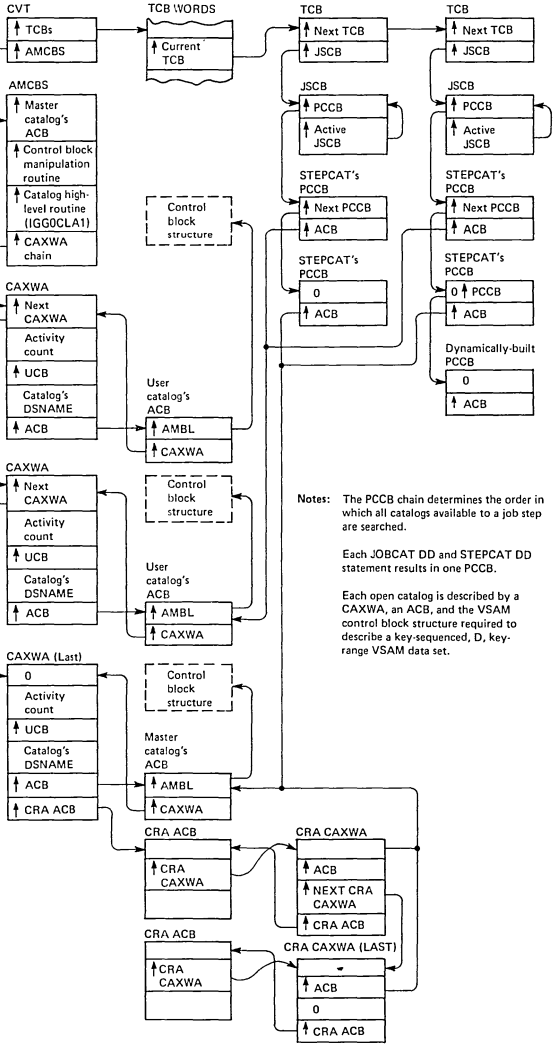


Figure 66. JES3 Control Blocks (Part 3 of 3)



Notes: The PCCB chain determines the order in which all catalogs available to a job step are searched.

Each JOBCAT DD and STEPCAT DD statement results in one PCCB.

Each open catalog is described by a CAXWA, an ACB, and the VSAM control block structure required to describe a key-sequenced, D, key-range VSAM data set.

Figure 67. Open Catalog Control Blocks

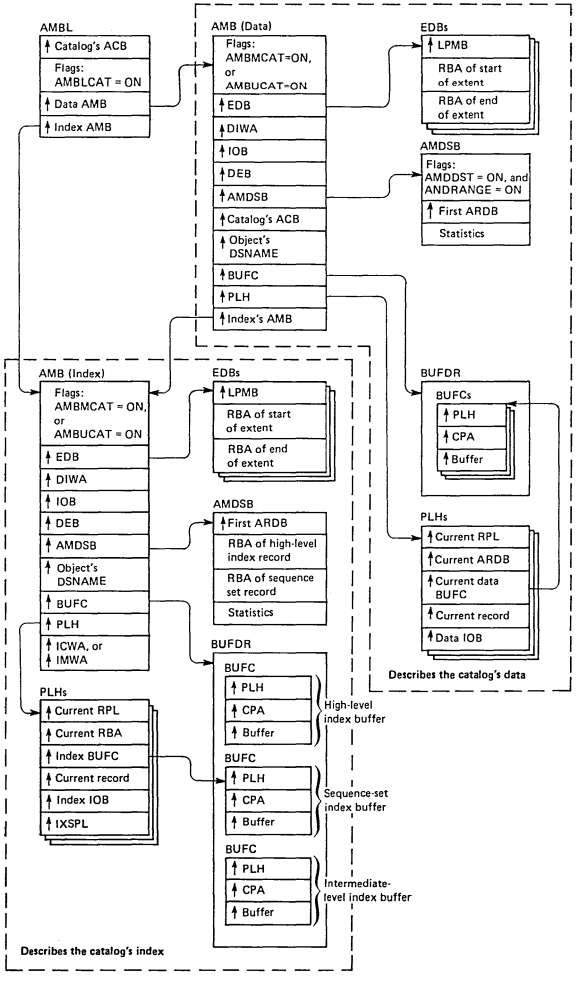


Figure 68. VSAM Control Blocks That Describe a Catalog (A Key-Sequenced Key-Range VSAM Data Set)

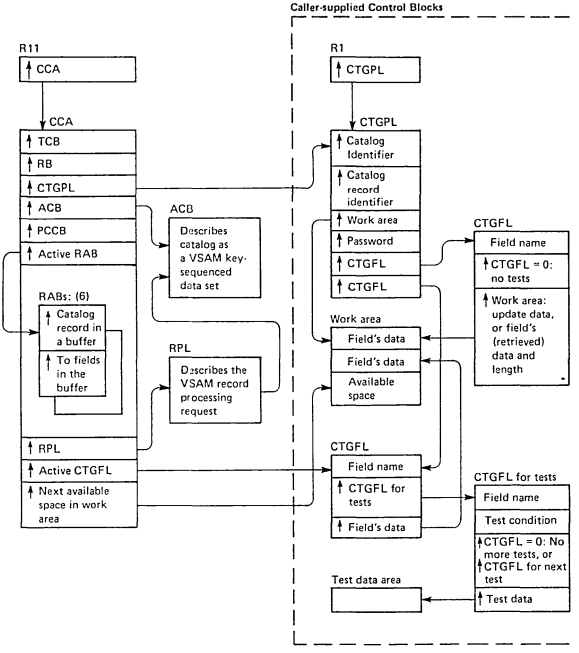


Figure 69. Catalog Management Control Blocks

VS2 System Programming Library: Debugging Handbook Volume 1 GC28-0708-1

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If comments apply to a Selectable Unit, please provide the name of the Selectable Unit_____.

If you wish a reply, give your name and mailing address:

_____

_____

_____

Please circle the description that most closely describes your occupation.

Customer	(Q) Install Mgr.	(U) System Consult.	(X) System Analyst	(Y) System Prog.	(Z) Applica. Prog.	(F) System Oper.	(I) I/O Oper.	(L) Term. Oper.						(O) Other
IBM	(S) System Eng.	(P) Prog. Sys. Rep.	(A) System Analyst	(B) System Prog.	(C) Applica. Prog.	(D) Dev. Prog.	(R) Comp. Prog.	(G) System Oper.	(J) I/O Oper.	(E) Ed. Dev. Rep.	(N) Cust. Eng.	(T) Tech. Staff Rep.		

Number of latest Newsletter associated with this publication: _____

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.)

First Class
Permit 40
Armonk
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

GC28-0708-1
File No. S370-37

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)

Printed in U.S.A.

OS/VS2 System Programming Library: Debugging Handbook

Volume 1

GC28-0708-1
File No. S370-37

Includes Selectable Units:

Scheduler Improvements	VS2.03.804
Supervisor Performance #1	VS2.03.805
Supervisor Performance #2	VS2.03.807
Data Management	VS2.03.808
IBM 3800 Printing Subsystem	VS2.03.810
TSO/VTAM	VS2.03.813
Scheduler/IOS Support	VS2.03.816
Service Data Improvements	VS2.03.817
MSS Enhancements	5752-824
3838 Vector Processing Subsystem	5752-829
3895 Device Support	5752-830
System Security Support	5752-832
Dumping Improvements	5752-833
Attached Processor Support	5752-847
MVS Processor Support	5752-851
Hardware Recovery Enhancements	5752-855
Interactive Problem Control System	5752-857
TSO/VTAM Level 2	5752-858
Data Management Support	5752-860

Includes Program Product:

MVS/System Extensions	5740-XE1

IBM Corporation, Publications Development, Dept. D58, Bldg. 706-2, PO Box 390, Poughkeepsie, New York 12602

Second Edition (November, 1978)

This is a major revision of and obsoletes GC28-0708-0 and GC28-0751-0 incorporating changes released in the following Technical Newsletters and System Library Supplements:

Scheduler Improvements	VS2.03.804	SU4	(GN28-2678)
Supervisor Performance #1	VS2.03.805	SU5	(GN28-2680)
Supervisor Performance #2	VS2.03.807	SU7	(GN28-2704)
Data Management	VS2.03.808	SU8	(GN28-2679)
IBM 3800 Printing Subsystem	VS2.03.810	SU10	(GN28-2722)
TSO/VTAM	VS2.03.813	SU13	(GN28-2655)
Service Data Improvements	VS2.03.817	SU11	(GN28-2768)
3838 Vector Processing Subsystem	5752-829	SU29	(GC28-0925-0)
3895 Device Support	5752-830	SU30	(GC28-0800-0)
System Security Support	5752-832	SU32	(GC28-0845-0)
Dumping Improvements	5752-833	SU33	(GC28-0816-0)
MVS Processor Support	5752-851	SU51	(GD23-0025-0)
Hardware Recovery Enhancements	5752-855	SU55	(GC28-0891-0)
Interactive Problem Control System	5752-857	SU57	(GD23-0096-0)
TSO/VTAM Level 2	5752-858	SU58	(GD23-0048-0)
Data Management Support	5752-860	SU60	(GD23-0076-0)
MVS/System Extensions (program product)	5752-XE1	XE1	(SD23-0001-0)

This edition with Technical Newsletters GN28-2967, GN28-2984 and GN28-4692 applies to Release 3.8 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.

The JES3 information contained in this manual is applicable only if JES3 has been integrated into your system.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

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 reader's 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. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Summary of Amendments for GC28-0708-1 As Updated by GN28-4692

This technical newsletter contains information in support of the 3800 MVS enhancements and miscellaneous technical and editorial changes.

Summary of Amendments for GC28-0708-1

General

This edition has been reorganized into a three volume publication. See the Preface and Contents for the basic design and setup.

Specific

- Volumes 1, 2, and 3 incorporate maintenance updates accumulated since the last revision. Also, the following SUs have been integrated into these volumes.

Scheduler Improvements	VS2.03.804
Supervisor Performance #1	VS2.03.805
Supervisor Performance #2	VS2.03.807
Data Management	VS2.03.808
IBM 3800 Printing Subsystem	VS2.03.810
TSO/VTAM	VS2.03.813
Service Data Improvements	VS2.03.817
3838 Vector Processing Subsystem	5752-829
3895 Device Support	5752-830
System Security Support	5752-832
Dumping Improvements	5752-833
MVS Processor Support	5752-851
Hardware Recovery Enhancements	5752-855
Interactive Problem Control System	5752-857
TSO/VTAM Level 2	5752-858
Data Management Support	5752-860

- Volume 1 incorporates program product information for MVS/System Extensions (5740-XE1) and highlights this information where applicable.
- Section 2 of Volume 2 (GC28-0709 or GC28-0752) Control Block Chains has been moved to Volume 1 (GC28-0708-1) as Section 6.
- Section 1 of Volume 2 (GC28-0709 or GC28-0752) - "How to Find Information" has been moved. This information is now contained in the description of the individual data areas. Each Volume 2 and 3 data area greater than 2 pages in length has a label-displacement list appended to it.
- The publications summary (Section 6 in GC28-0708 or GC28-0751) has been deleted and replaced by a list of applicable publications in the Preface of Volume 1 (GC28-0708-1). A complete list of MVS publications can be obtained from the MVS Release Guide.

This edition has been reorganized for a three volume publication. See the Preface and Contents for the basic design and setup.

Code	Explanation
h45	*SVC 69 (0A45) CODES*
145	An invalid DCB or ICB is used.
245	An error occurred that was not caused by invalid user input.
h4F	*SVC 79 (0A4F) CODES*
14F	The STATUS macro was used for a function other than STOP, STOP SYNCH, or START, and the user was not in supervisor key (0-7).
h51	*SVC 81 (0A51) CODES*
151	An invalid parameter list address was passed to the SETPRT macro.
251	The SETPRT macro detected an invalid DCB exit list or user supplied image in storage.
351	The SETPRT macro could not be executed.
451	The SETPRT macro detected an invalid DCB, invalid IOBs, an invalid DEB address, or a DCB exit list address.
h53	*SVC 83 (0A53) CODES*
153	An error occurred in the suspend/reset portion of SMF record processing.
253	SMF processing errors caused SMF to terminate.
h57	*SVC 87 (0A57) CODES*
157	The issuer of a DOM request is not in the same address space and is not a privileged task.
h5C	*SVC 92 (0A5C) CODES*
15C	The program issuing SVC 92 was not in supervisor state.
h5D	*SVC 93 (0A5D) CODES*
15D	An invalid data area is specified in the TGET or TPUT macro and detected by SVC 93.
h5F	*SVC 95 (0A5F) CODES*
15F	The system resources manager is invalidly invoked.
25F	A system failure destroyed the data used by the system resources manager.
35F	An address space was non-swappable when a SYSEVENT, TRANSWAP (X'0E'), was issued.
45F	A quiese or swap-out failed while a transition swap (TRANSWAP) or requested swap (REQSWAP) was pending.
55F	An error was encountered during the execution of the XMPOST SRB for an issuer of SYSEVENT, TRANSWAP or REQSWAP.
h60	*SVC 96 (0A60) CODES*
260	The STAX SVC detected an invalid user parameter.
h69	*SVC 105 (0A69) CODES*
169	The CLOSE macro passed an invalid DCB address to IMGLIB.
269	An error occurred during the execution of the IMGLIB macro.
h6B	*SVC 107 (0A6B) CODES*
16B	An invalid parameter list exists.
h6D	*SVC 109 (0A6D) CODES*
16D	An invalid ESR code for SVC 109, 116, or 122 was detected.
26D	MF/1 could not initialize, collect, or reinitialize the workload activity measurements.
36D	A communications error occurred between RMF and the system resources manager.
h6E	*SVC 110 (0A6E) CODES*
16E	The DEBCHK function was not completed. (Applies to SVC 117, not SVC 110.)

Code	Explanation
h6F	*SVC 111 (0A6F) CODES*
36F	An invalid entry code was passed to SVC 111, or DEBCHK processing found that the DEB was invalid, or the DEB was not a subsystem DEB.
56F	An error occurred during end-of-job processing for a JES2 job that was being processed by an execution batch monitor.
h71	*SVC 113 (0A71) CODES*
171	An illegal or invalid request is made for the PGFIX, PGFREE, PGLOAD, or PGOUT macro.
271	An unauthorized call is made for the PGFIX or PGFREE macro.
h72	*SVC 114 (0A72) CODE*
172	An SVC 114 was issued with one of the following conditions: the protect key was not correct (not 0 thru 7), the request was not issued in the supervisor state, or the authorization bit was not set in the JSCB.
h75	*SVC 117 (0A75) CODES*
—	(See completion code 16E)
h77	*SVC 119 (0A77) CODES*
177	Invalid input parameters were detected by TESTAUTH.
h78	*SVC 120 (0A78) CODES*
178	The RC or RU type GETMAIN macro requested more bytes of virtual storage from the SQA than were available in real storage.
378	The RC or RU type FREEMAIN macro contained an error in its subpool specification.
478	The RC or RU type FREEMAIN macro contained an error in its subpool specification.
778	An uncorrectable machine, system, or indeterminate error occurred while processing the RC or RU type GETMAIN or FREEMAIN macro.
878	Not enough virtual storage is available for the GETMAIN issuer, or the RC or RU type GETMAIN issuer specified a negative length.
978	The address of the storage area specified by the RC or RU type FREEMAIN macro was not on a doubleword boundary.
A78	An error occurred in defining the area to be freed by the RC or RU type FREEMAIN macro.
B78	The RC or RU type GETMAIN or FREEMAIN macro contained a subpool specification error.
D78	The RC or RU type FREEMAIN macro attempted to free an unowned storage area.
h79	*SVC 121 (0A79) CODES*
179	Any one of the following conditions: • PGFIX or SETLOCK fail. • The conversion to a real address (honoring a LRA instruction) fails. • The block size for track overflow is not 4k.
279	An invalid BUFC or protection check is indicated in the IOSB by IOS.
h7A	*SVC 122 (0A7A) CODES*
17A	An invalid EVENTS table address was specified.
37A	A request was issued for an EVENTS table being waited on by another program.
47A	The ENTRIES specification is incorrect (not from 1 to 32,767).
h7B	*SVC 123 (0A7B) CODES*
17B	An invalid ASID was specified when the PURGEDQ SVC was issued.
27B	A PURGEDQ received a non-zero return code from FESTAE.

SVC - Macro List

The following is a list of SVCs and associated macros.

SVC	Macro	SVC	Macro
0	EXCP/XDAP	70	GSERV
1	WAIT/WAITR/PRTOV	71	ASGNBFR/BUFINQ/
2	POST		RLSEBFR
3	EXIT	72	'NO MACRO'
4	GETMAIN	73	SPAR
5	FREEMAIN	74	DAR
6	LINK	75	DQUEUE
7	XCTL	76	IFBSTAT
8	LOAD	77	'RESERVED'
9	DELETE	78	LSPACE
10	GETMAIN/FREEMAIN (with R operand)	79	STATUS
		80	'RESERVED'
11	TIME	81	SETPRT
12	SYNCH	82	DASDR
13	ABEND	83	SMFWTM
14	SPIE	84	GRAPHICS
15	ERREXCP	85	DDRSWAP
16	PURGE	86	ATLAS
17	RESTORE	87	DOM
18	BLDL/FIND (TYPE D)	88	MOD88
19	OPEN	89	'RESERVED'
20	CLOSE	90	'RESERVED'
21	STOW	91	VOLSTAT
22	OPEN (TYPE = J)	92	TCPEXCP
23	CLOSE (TYPE = T)	93	TGET/TPUT
24	DEVTYPE	94	'SEE SVC 94'
25	TRKBAL	95	SYSEVENT
26	CATALOG/INDEX/	96	STAX
	LOCATE	97	IKJEGS9G
27	OBTAIN	98	PROTECT
28	'RESERVED'	99	DYNALLOC
29	SCRATCH	100	IKJEFFIB
30	RENAME	101	QTIP
31	FEOV	102	AQCTL
32	ALLOC	103	XLATE
33	IOHALT	104	TOPCTL
34	MGCR/QEDIT	105	IMGLIB
35	WTO/WTOR	106	'RESERVED'
36	WTL	107	MODESET
37	SEGLD/SEGWT	108	'RESERVED'
38	'RESERVED'	109	'SEE SVC109'
39	LABEL	110	'NO MACRO'
40	EXTRACT	111	'NO MACRO'
41	IDENTIFY	112	PGRLSE
42	ATTACH	113	PGFIX/PGFREE/
43	CIRB		PGLOAD/PGOUT
44	CHAP	114	EXCPVR
45	OVLYBRCH	115	'RESERVED'
46	TTIMER	116	'SEE SVC116'
47	STIMER	117	DEBCHK
48	DEQ	118	'RESERVED'
49	'RESERVED'	119	TESTAUTH
50	'RESERVED'	120	GETMAIN/FREEMAIN
51	SNAP/SDUMP	121	VSAM
52	RESTART	122	'SEE SVC122'
53	RELEX	123	PURGEDQ
54	DISABLE	124	TPIO
55	EQV	125	EVENTS
56	ENQ/RESERVE	126	MSS(ICB2SVC)
57	FREEDBUF	127	'RESERVED'
58	RELBUF/REQBUF	128	'RESERVED'
59	OLTEP	129	'RESERVED'
60	STAE/STAI-ESTAE/ESTAI	130	RACHECK
61	IKJEGS6A	131	RACINIT
62	DETACH	132	RACLIST
63	CHKPT	133	RACDEF
64	RDJFCB	134	'RESERVED'
65	'RESERVED'	135	'RESERVED'
66	BTAMTEST	136	'RESERVED'
67	'RESERVED'	137	'RESERVED'
68	SYNADAF/SYNADRLS	138	'RESERVED'
69	BSP		

SVC 0 (0A00): EXCP/XDAP macro - is type 1, gets LOCAL lock, calls module IECVEXCP (E.P. = IGC000). PLM is *OS/VS2 I/O Supervisor Logic*. GTF data is:

R15 and R0 - No applicable data.
R1 Address of the IOB associated with this request.
DDNAME cccccccc Name of the associated DD statement.
DCB xxxxxxxx Address of the DCB associated with this I/O request.
DEB xxxxxxxx Address of the DEB associated with this I/O request.

SVC 1 (0A01) WAIT/WAITR/PRTOV macro - is type 1, gets LOCAL lock, calls module IEAVSY50 (E.P. = IGC001) for WAIT/WAITR, module IGG019CL for PRTOV. PLM is *OS/VS2 System Logic Library* for WAIT/WAITR, *OS/VS2 SAM Logic* for PRTOV. GTF data is:

R15 No applicable data.
R0 Count of the number of events being waited for. If the count is zero, the wait is treated as a NOP. Bit 0 equals one indicates a long wait.
R1 If positive, the address of the ECB being used. If complemented, the address of a list of ECB addresses.
PLIST - The list is a series of fullwords, each containing the address of an ECB.

SVC 2 (0A02) - POST macro - is type 1, gets LOCAL and SALLOC locks, calls module IEAVSY50 (E.P. = IGC002). PLM is: *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.
R0 For POST: The completion code to be placed in the ECB.
R1 For POST: The address of the ECB to be posted or (if the high-order bit is 1), the address of a parameter list as follows:
Bytes:
0-3 Address of the ECB.
4-7 Address of the ASCB for the address space that contains the ECB
8-11 Address of the ERRET routine.
12 Contains the storage protection key of the ECB if the high-order list of R0 is on and the high-order bit of R1 is on.

SVC 3 (0A03) - EXIT macro - is type 1, gets LOCAL lock, calls module IEAVEOR (E.P. = IGC003). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15, R0, and R1 - No applicable data.

28-35 Entry point name for EP; blank or zeros for EPLOC or DE specification.

36 Reserved.

37-39 JSCB address.

40 Task ID—AUTH ONLY.

41-43 Address of the parameter list for the STAI or the ESTAI routine.

44 Flag bytes:

....	..00	Purge = QUIESCE specified.
....	..01	Purge = HALT specified.
....	..10	Purge = NONE specified.
....	.1..	ASYNCH = YES.
....	1...	ESTAI was specified.
...1	....	TERM = YES was specified.
000.	....	Reserved.

45-47 Address of the routine to get control if the subtask ABENDs.

48 Reserved.

49-51 Address of the opened TASKLIB DCB.

52 Indicators:

0...	....	Word 56 contains a subpool number (AUTH ONLY) or zero.
1...	....	Word 56 contains the address of a list of subpools—AUTH ONLY.
.1..	....	Reset APF.
..00	0000	Reserved.

53 Reserved.

54 Length of the parameter list in bytes.

56 A subpool number or a list of subpool numbers. See byte 52—AUTH ONLY.

Length of PLIST (1 byte)

PLIST - PLIST up to 40 bytes of parameter list passed to a program. The parameter list is a series of 4-byte entries. Each entry has its high-order byte reserved and an address in the low-order three bytes.

Note: Contents of register 1 at the time SVC 42 is issued are passed to the attached program.

SVC 43 (0A2B) CIRB macro - is type 1, gets LOCAL lock, calls module IEAVEF00 (E.P. = IGC043). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Entry point address of the user's asynchronous exit routine. When the routine is dispatched it will get control at this entry point.

R1 The meanings of the bytes of the register are as follows:

Byte 1

0100	0...	A normal IRB is being created.
1000	0...	An SIRB is being created. This is used only by IOS to run ERP routines.
....	.1..	Problem program key.
....	.0..	Supervisor key.
....	..1.	Problem program state.
....	..0.	Supervisor state.
....	...1	Save area for registers requested.
....	...0	No save area requested.

Byte 2

0000	...0	Reserved - always zero.
....	1...	Indicates that the IQE's are going to schedule the routine.
....	0...	Indicates that the RQE's are going to schedule the routine
....	.1..	Return the IQEs at exit if the IRB has a work area and the RBUSIOE flag is not on.
....	.0..	Do not return the IQE's at exit.
....	..1.	Indicates that the RB will be freed when the exit issues an SVC 3.
....	..0.	Indicates that the RB will not be freed when the exit issues an SVC 3.

Byte 3

Reserved.

Byte 4

Indicates the size in doublewords of the work area to be acquired. CIRB will unconditionally request space from subpool 253. The maximum size is 255 doublewords.

SVC 44 (0A2C) CHAP macro - is type 2, gets LOCAL lock, calls module IEAVECH0 (IGC044). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Signed value to be added to the dispatching priority of the specified task; negative value will be in two's-complement form.

R1 Address of the area containing the address of TCB whose priority is to be changed; or zeros. Zeros indicates that the active task's priority is to be changed.

CHAP TCB hhhhhhhh Address of the TCB whose priority is to be changed. Must be a subtask of the current task.

SVC 45 (0A2D) OVLYBRCH macro - is type 2, gets no lock, calls module IEWSUOVR (E.P. = IGC045). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 Address of the entry-table entry that caused the SVC to be issued.

R0 and R1 - No applicable data.

PLIST - 12 bytes long; format is:

Bytes	
0-3	Branch instruction to SVC 45.
4-7	Address of the referred-to symbol.
8	'To' segment number.
9-11	Previous caller or zero.

SVC 46 (0A2E) TTIMER macro - is type 3, gets no locks, calls module IEAVRT00 (E.P. = IGC0004F). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Address of the doubleword where time in microseconds is to be returned.

R1 Low-order byte has code determining the type of request and the format of the returned value.

Code		
....	.000	Time remaining in the current task's time interval is to be in register 0 in timer units; the time interval is not to be canceled.
....	...1	Current task's time interval is to be canceled.
....	..1.	Same as .....000 except the interval remaining is returned to the specified address in microseconds.
....	.1..	Routine specified by the ERRET parameter gets control on an environmental error.

SVC 47 (0A2F) STIMER macro - is type 3, gets no locks, calls module IEAVRT00 (E.P. = IGC0004G). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.

R0 Contents:

Bytes			
0	STIMER option byte as follows:		
	0000	....	TUINTVL option.
	0001	....	BINTVL option.
	0010	....	MICVL option.
	0011	....	DINTVL option.
	0110	....	GMT option.
	0111	....	TOD option.
	....	.000	Interval to be decreased only when the task is active.
	....	.001	Decrease the interval continuously and put the task in the wait state until the interval expires.
	....	.011	Decrease the interval continuously.
	....	1...	ERRET bit: Control is returned on errors with register 15 set to 8.
1-3	Exit address.		

R1 Address of the time value.

PLIST - 4 or 8 bytes depending on option in force:

a. DINTVL,TOD,MICVL, and GMT - 8 bytes; represents the time value.

b. BINTVL and TUINTVL - 4 bytes; represents the time value.

SVC 60 (0A3C) STAE/STAI-ESTAE/ESTAI macro - is type 3, gets LOCAL lock, calls module IEAVSTA0 (E.P. = IGC00060). PLM is *OS/VS2 System Logic Library*. GTF data is:

FOR STAE/STAI REQUESTS

R15 No applicable data.

R0 Contents:

- 00 Create.
- 04 Cancel.
- 08 Overlay.

R1 Address of the parameter list. The high-order bit is set to 1 if the XCTL=YES parameter was coded.

PLIST 12 bytes long; format is:

Bytes			
0	Flag byte:		
	1...	....	TCB address is supplied.
	....	.1..	Allow asynchronous exit scheduling.
	....	..10	Do not purge I/O operations.
	....	..01	Purge I/O operations with the halt option.
	....	..00	Purge I/O operations with the quiesce option.
	.xx1	x...	Reserved and set to zero.
1-3	If zero, the CANCEL operand is in effect; otherwise, the address of the STAE/STAI exit routine.		
4-7	Address of the exit routine parameter list; if zero, no exit routine parameter list exists.		
8-11	TCB address for a STAI request.		

FOR ESTAE/ESTAI REQUESTS

R15 No applicable data.

R0 Contents:

- 00 A new ESTAE parameter list is to be created.
- 02 Previous STAI/ESTAI exits are to be propagated from the originating task.
- 04 Cancel the most recent STAE request.
- 84 Cancel the most recent ESTAE request.
- 94 Branch enter to cancel the most recent ESTAE request.
- 08 Overlay the previous ESTAE parameter list with the parameters passed in this request.

R1 Address of the parameter list. The high-order bit is set to 1 if the ESTAE macro is not to be canceled when an XCTL is issued, and to 0 if the ESTAE macro is to be canceled when an XCTL is issued.

PLIST 16 bytes long; format is:

Bytes			
0	Flag byte:		
	1...	....	TCB address is supplied.
	.1..	....	ESTAR indicator.
	...1	....	ESTAE/ESTAI/ESTAR parameters.
	....	.1..	Allow asynchronous exit scheduling.
	....	..10	Do not purge I/O operations.
	....	..01	Purge I/O operations with the halt option.
	....	..00	Purge I/O operations with the quiesce option.
	..0.	0...	Reserved and set to zero.
1-3	Address of the user exit routine.		
4-7	Address of the user parameter list.		
8-11	TCB address if an ESTAI request, otherwise zero.		
12	Flag byte:		
	.1..	....	Request for termination processing.
	..1.	....	Request for error recording.
	....	.1..	Request branch entry to SVC 60.
13-15	Reserved.		

SVC 61 (0A3D) IKJEGS6A macro - is type 3, gets LOCAL lock, calls module IGC0006A. PLM is *OS/VS2 TSO Command Processor Logic: Volume III (TEST)*. GTF data is:

R15 and R0 - No applicable information.

R1 Contains:

Zeros if the routine is being entered from the overlay supervisor.

Address of the DCB used to fetch the module if the routine is being entered from the contents supervisor.

SVC 62 (0A3E) DETACH macro - is type 2, gets LOCAL lock, calls module IEAVEED0 (E.P. = IGC062). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the fullword containing the address of the subtask TCB to be detached. If bit 0 = 1, STAE=YES was specified. This affects the abend code with which an incomplete subtask is abended; If STAE=YES the code is 33E, otherwise it is 13E.

DETACH TCB hhhhhhhh Address of the subtask TCB to be detached.

Note: If R1 contains zeros, the DETACH TCB field is meaningless, and the issuer of SVC 62 will be abended with code 23E.

SVC 63 (0A3F) CHKPT macro - is type 4, gets LOCAL and CMS locks, calls module IHJACP00 (E.P. = IGC0006C). PLM is *OS/VS2 Checkpoint Restart Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Contents:

a. Address of the parameter list.

b. Zero if for a CANCEL request.

PLIST

8 bytes long; format is:

Bytes		
0	00	Check the ID address provided in the second parameter of CHKPT macro instruction.
	80	No check ID address is provided.
1-3	Address of the checkpoint DCB.	
4	00	Check ID address is provided.
	01 to 10	Check ID length is provided via the third parameter of the CHKPT macro instruction.
	FF	'S' specified as the third parameter of the CHKPT macro instruction; the system-generated check ID is to be placed at the address specified in bytes 5-7.
5-7	Address for storing the system-generated check ID or the address of the user provided check ID.	

SVC 64 (0A40) RDJFCB macro - is type 3, gets LOCAL lock, calls module IGC0006D. PLM is *OS/VS2 OPEN/CLOSE/EOV Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the parameter list.

PLIST four to 40 bytes of the RDJFCB parameter list, which has a maximum of 1020 bytes. The list is a series of 4-byte entries, each containing a DCB address. The high-order byte has bit 0 set to one to indicate the last entry.

SVC 65 (0A41) Reserved.

SVC 66 (0A42) BTAMTEST macro - is type 4, gets no lock, calls module IGC0006F. PLM is *OS/VS2 BTAM Logic*. GTF data is:

R15 and R0 - No applicable data.

R1 Address of the IOB when the SVC was issued.

IOBERINF Address of the RFT message, inserted by the channel end appendage (IGG019MB).

IOBERNIF+4 Address of the parameter list, inserted by the terminal test control (IGG019MR).

SVC 67 (0A43) Reserved.

SVC 74 (0A4A) DAR macro - is type 3, gets LOCAL lock, calls module IGC0007D. PLM is *OS/VS2 Graphics Access Method Logic*. GTF data is:

R15 and R0 - No applicable information.

R1 Address of the parameter list.

PLIST up to 40 bytes. It is a series of 4-byte entries. First entry has the format:

Bytes

0-1 Reserved.

2-3 Number of words in the parameter list.

Each additional entry contains the GACB address specified by the DAR macro instruction.

SVC 75 (0A4B) DQUEUE macro - is type 3, gets LOCAL lock, calls module IGC0007E. PLM is *OS/VS2 Graphics Access Method Logic*. GTF data is:

R15 No applicable data.

R0 Address of next the IQE on the IRB active list for the attention routine when ATTNINQ has specified the clear mode; otherwise, contains zeros.

R1 content:

Bytes

0 Unit index to identify a particular 2260 display station; or 00 for a 2250 station.

1-3 GACB address.

IQE When ATTNINQ specifies clear mode this field contains the first 3 words of the IQE pointed to by R0:

Bytes

0-3 Address of the next IQE in the chain, or zeros.

4-7 No applicable data.

8-11 Address of the IRB associated with the IQE. N/A will appear in this field whenever the ATTNINQ macro instruction did not specify the clear mode.

SVC 76 (0A4C) No macro - is type 3, gets no lock, calls module IFBSVC76 (E.P. = IGC0007F). PLM is *OS/VS2 SYS1.LOGREC Error Recording Logic*. APF protected. GTF data is:

R15 No applicable data.

R0 If positive, contains the function indicator in byte 3:

00 Indicates that the EOD recording is requested.

04 Indicates that the EREP entry to record statistical information in SYS1.LOGREC is requested.

08 Indicates that an IPL recording is requested.

0C Indicates entry to update date and time values in the SYS1.LOGREC time-stamp record.

If negative (complemented), contains the length in bytes of a record to be placed in the SYS1.LOGREC data set.

R1 If R0 is positive, R1 contains no applicable data. If R0 is negative, R1 contains the address of the record to be written.

SVC 77 (0A4D) Reserved.

SVC 78 (0A4E) No macro - is type 3, gets LOCAL lock, calls module IGC0007H. PLM is *OS/VS2 DADSM Logic*. GTF data is:

R15 No applicable data.

R0 Address of the associated UCB.

R1 SMF indicator and/or the message buffer address as follows:

Bytes

0 SMF indicator (caller must be in protect key 0 or authorized to specify either SMF indicator).

X'80' - Build SMF record type 19.

X'40' - LSPACE should test if the SMF volume information is requested before building the SMF record type 19.

1-3 zero or the address of a 30-byte message buffer.

CUU cccc unit address in channel-unit format.

SVC 79 (0A4F) STATUS macro - is type 1, gets LOCAL, CMS, CMSEQDQ, SALLOC, and DISP locks, plus the local and global intersect, calls module IEAVSETS (E.P. = IGC079). PLM is *OS/VS2 System Logic LIbrary*. GTF data is:

The two low-order bytes of register 0 contain a STATUS function code. Depending on the code, registers 15 and 1 contain other information as shown.

Register 0		Function	Register 1	Register 15
0-1	2-3			
0000	0001	MCSTEP	N/A	N/A
MASK	0003	NDSTEP	N/A	ASID(XM status only for reset/start)
MASK	0004	NDSYS	N/A	N/A
N/A	0005	NDTCB	†TCB	ASID
0000	0006	STOP	0 or †TCB	N/A
0000	0007	START	0 or †TCB	N/A
ASID	0008	SDSTEP	N/A	MASK
N/A	0009	SDSYS	N/A	N/A
ASID	000A	SDTCB	†TCB	MASK
ASID	000B	SDETCB	†TCB	MASK
MASK	000C	NDETCB	†TCB	ASID(XM status only for reset or start)
0000	000D	SRBS	N/A	ASID(XM status only for reset or start)
0000	000E	SYNCH	N/A	N/A
0000	000F	Caller, SD	†TCB	MASK
0000	000F	Caller, SD	N/A	N/A (for MVS/System Extensions)
0000	0010	Caller, ND	†TCB	MASK
0000	0010	Caller, ND	N/A	N/A(for MVS/System Extensions)
0000	0011	SRBs only	N/A	ASID(XM status only for reset/start - for MVS/System Extensions)

Note: The sign bit of register one indicates:
0 = set(stop).
1 = reset(start).
not applicable to codes 6, 7, 14, 15, 16.

SVC 80 (0A50) Reserved.

SVC 81 (0A51) SETPRT macro - is type 4, gets no lock, calls module IGC0008A. PLM is *OS/VS2 SAM Logic*. GTF data is:

R15 and R0 - No applicable data.
R1 Address of the parameter list.
DDNAME cccccccc Name of the DD statement associated with the data set being printed.
PLIST parameter list of up to 14 words being passed to SVC 81.
Bytes
0-3 address of the DCB.
4-7 EBCDIC UCS image ID.
8 LOAD MODE indicator; bit settings are:
Bits
.0.. .... No fold.
.1.. .... Fold.
x.xx xxxx Reserved.
9 Verification indicator; bit settings are:
Bits
...1 .... Display the image on the printer for verification.
...0 .... Do not display the image on the printer for verification.
xxx. xxxx Reserved.

10 Data check indicator; bit settings are:

Bits		
10..	....	Block data checks.
01..	....	Unblock data checks.
00..	....	Data checks the DCB specifies.
..10	....	Schedule SYSOUT data segment for printing now.
..01	....	Do not schedule SYSOUT data segment for immediate printing.
....	10..	Unfold 3203 or 3211 UCS.
....	01..	Fold 3203 or 3211 UCS.
..xx	..x.	Reserved.
....	...1	SETPRT parameter list is extended to at least 48 bytes in length.

11-14 EBCDIC FCB image ID.

15 FCB parameter options; bit settings are:

Bits		
1...	....	Verify the FCB.
....	...1	Align.
.xxx	xxx.	Reserved.

16 SPPFLAG1 Flag indicators; bit settings are:

Bits		
0...	....	BURST=N, thread continuous forms stacker.
1...	....	BURST=Y, thread burster-trimmer-stacker.
.1..	....	REXMIT=Y, retransmission-only change COPIES, FLASH and starting copy number.
..1.	....	INIT=Y, initialize the printer.
...1	....	PRTMSG=N, suppress error messages in the printer.
....	1...	Bypass the "load forms overlay" message and status display.
....	.1..	Bypass the stacker setup message and status display.
....	..1.	Bypass WCGM overflow message (JES2 only).
....	...1	Load the requested FDB (JES2 and JES3 only).

17 SPPFLAG2 Flag indicators; bit settings are:

Bits		
1...	....	MODIFY is specified as an address.
0...	....	MODIFY is not specified or is specified as a name.
.1..	....	First character arrangement table is specified as an address.
.0..	....	First character arrangement table is specified as a name or is not specified.
..1.	....	Second character arrangement table is specified as an address.
..0.	....	Second character arrangement table is specified as a name or is not specified.
...1	....	Third character arrangement table is specified as an address.
...0	....	Third character arrangement table is specified as a name or is not specified.
....	1...	Fourth character arrangement table is specified as an address.
....	0...	Fourth character arrangement table is specified as a name or is not specified.
....	.1..	FCB is specified as an address (3800 only).
....	.0..	FCB is specified as a name or is not specified.
....	..xx	Reserved.

18 Number of copies to be printed on this transmission.

19 Starting copy number.

20 Length of the parameter list.

22 Number of copies to be forms flashed on this transmission.

23 Table reference character for copy modification.

24 The last 4 bytes of a module name or a pointer to the copy modification control record.

28 The 4 character name of a forms overlay frame.

32 The last 4 bytes of a member name or a pointer to the first character arrangement table module.

36 The last 4 bytes of a member name or a pointer to the second Character arrangement table module.

40 The last 4 bytes of a member name or a pointer to the third character arrangement table module.

44	The last 4 bytes of a member name or a pointer to the fourth character arrangement table module.
48	Address of the message communication area for error information.
52	Address of the DCB for a user library to load 3800 setup modules.

SVC 82 (0A52) DASDR macro - is type 4, gets no lock, calls module IGC0008B. PLM is *OS/VS2 Utilities Logic*. APF protected. GTF data is:

R15 and R0 - No applicable data.
R1 Address of the parameter list.
VOLSER cccc volume serial number.
DA-ADDR hhhhhhhh hhhhhhhh field displayed depends on the options in effect for the SVC routine.
Option Direct access address content.
Analyze or format 6-byte track address.
Post UCB 8-byte track address.
Address of alternate
track CCHH N/A.
Unlabeled volume 8-byte track address.
New volume N/A.
PLIST parameter list up to 16 bytes long pointed to by R1. First 4 bytes include a flag byte defining the function to be performed, and a 3-byte UCB address. The fourth, eighth, and twelfth bytes, when present, will contain a flag indicating the last element (4-bytes) in the list. Last flag bit settings are:
Bytes
0 Function byte as follows:
8F New volume.
1F Address of the alternate track CCHH.
00 ANALYSE or FORMAT.
08 POST UCB.
88 Unlabeled volume.
F8 Delete the DEB analyze of label.
1-3 Address of the UCB.
The contents of the remaining bytes are determined by the function specified in byte 0.
(Function 8F)
4 80 Flag byte in the last element.
5-7 Address of DCB.
(Function 1F)
4 80 Flag byte in the last element.
5-7 Address of the alternate track CCHH.
(Function 00)
4-7 Address of the alternate track CCHH.
8 80 Flag byte in the last element.
9-11 Address of alternate track information.
(Function 08)
4-7 Address of the serial number.
8 80 Flag byte in the last element.
9-11 Address of the VTOC.
(Function 88)
4-7 Address of the serial number.
8-11 Address of the VTOC.
12 80 Flag byte in the last element.
13-15 Address of the DEB.
(Function F8)
4-7 Address of the serial number.
8-11 Address of the VTOC.
12 80 Flag byte in the last element.
13-15 Address of the DEB.

SVC 83 (0A53) SMFWTM macro - is type 3, gets no lock, calls module IEEMB830. PLM is *OS/VS2 System Logic Library*. APF protected. GTF data is:

R15 and R0 - No applicable data.
R1 The address of an SMF record that is to be written to an SMF data set.

SVC 107 (0A6B) MODESET macro - is type 6, gets no lock, calls module IEAVMODE (E.P. = IGC107). PLM is *OS/VS2 System Logic Library*. APF protected. GTF data is:

R15 and R0 - No applicable data.
R1 Parameter list:

Bytes			
0-2	Reserved (must be zero).		
3	Indicator bits:		
	0000	....	No action.
	0001	....	Invalid.
	0010	....	Place the TCB key in the RBOPSW field of the RB.
	0011	....	Set the RBOPSW key to zero.
	....	0000	No action.
	....	0100	Turn on the state bit in RBOPSW field of the RB (problem state).
	....	1000	Invalid.
	....	1100	Turn off the state bit in RBOPSW field of the RB (supervisor state).

SVC 108 (0A6C) Reserved.

SVC 109 (0A6D) ESR (type 4) SVC - is type 4.

Routes control to types 3 and 4 extended supervisor service routines based on the routing code in register 15.

Code	Macro	Description
00		Reserved.
01		Reserved.
02		Reserved.
03		Reserved.
04		Reserved.
05		Reserved.
06	VSAM MSS Support	SVC - calls module IGX00006. PLM is SY26-3825.
07	MFSTART(RMF)	Authorization required - calls module IGX00007.
08		Reserved.
09		Reserved.
0A		Reserved.
0B		Reserved.
0C		Reserved.
0D	MFSTART (MF/1)	Authorization required - calls module IGX00013. PLM is *OS/VS2 System Logic Library*.
0E	MFDATA (MF/1)	Authorization required - calls module IGX00014. PLM is *OS/VS2 System Logic Library*.
0F		Task termination - calls module IGX00015. PLM is *OS/VS2 System Logic Library*.
10		Reserved.
11		Reserved.
12		Reserved.
13		Reserved.
14		Reserved.
15		Reserved.
16	MFDATA(RMF)	Authorization required - calls module IGX00022.
17		Reserved.
18		Reserved.
19		SMF transaction count calls module IGX00025.
1A		Reserved.
1B		Reserved.
1C		Reserved.
1E		Reserved.
1F		Reserved.
20		Reserved.
21		Reserved.

SVC 110 (0A6E) No macro - is type 4, gets no lock, calls module IEE00110. PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 Zeros indicate the first entry into the SVC routine.
Nonzeros indicate a request to FREE the CSCB pointed by the address in register 1.
R0 No applicable data.
R1 Address of the CSCB.

SVC 111 (0A6F) No macro - is type 2, gets LOCAL and CMS locks, calls module IGC111. PLM is *OS/VS2 JES2 Logic*. GTF data is:

R15 No applicable data.
R0 Contains the function indicator in the low-order byte; refer to the HASPSSSM (SVCHAM) for JES2 or IATDMEB for JES3 program listing for an interpretation.
R1 If positive, contains the address of the RPL. If negative (complemented), contains the address of the ACB.

SVC 112 (0A70) PGRLSE macro - is type 1, gets LOCAL and SALLOC locks, calls module IEAVPSI (E.P. = IGC112). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.
R0 Starting address of the virtual area to be operated on.
R1 End address of that area plus 1.

SVC 113 (0A71) PGFIX/PGFREE/PGLOAD/PGOUT macro - is type 1, gets LOCAL and SALLOC locks, calls module IEAVPSI (E.P. = IGC113). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 If the high-order bit of register 1 is off, contains the second word of the virtual subarea list (VSL).
R0 If positive, contains the address of the ECB.
R1 If the high-order bit is on, contains the address of the VSL.
If high-order bit is off, contains the first word of the VSL; register 15 will contain the second word.

Virtual Subarea List

Byte 0 Flags:

Bit 0	(1... ....)	This bit indicates that bytes 1-3 are a chain pointer to the next VSL entry to be processed; bytes 4-7 are ignored, but the checking of this bit is subject to the setting of byte 4, bit 1. This feature allows several parameter lists to be chained as a single logical parameter list.
Bit 1	(.1.. ....)	PGFIX is to be performed; reserved, set by macro instruction.
Bit 2	(..1. ....)	PGFREE is to be performed; reserved, set by macro instruction.
Bit 3	(...1 ....)	PGLOAD is to be performed; reserved, set by macro instruction.
Bit 4	(.... 1...)	PGRLSE is to be performed; reserved, set by macro instruction.
Bit 5	(.... .1..)	Reserved.
Bit 6	(.... ..1.)	Long-term PGFIX is to be performed; reserved, set by macro instruction.
Bit 7	(.... ...1)	Reserved.

Start Address:
The virtual address of the origin of the virtual area to be processed.

Byte 4 Flags:

Bit 0	(1... ....)	This flag indicates the last entry of the list. It is set in the last doubleword entry in the list.
Bit 1	(.1.. ....)	When this flag is set, the entry in which it is set is ignored. This bit takes precedence over byte 0, bit 0.
Bit 2	(..1. ....)	Reserved.
Bit 3	(...1 ....)	This flag indicates that a return code of 4 was issued from a page service function other than PGRLSE.
Bit 4	(.... 1...)	Reserved.
Bit 5	(.... .1..)	PGOUT is to be performed; reserved, set by macro instruction.
Bit 6	(.... ..1.)	KEEPREAL option of PGOUT is to be performed; reserved, set by macro instruction.
Bit 7	(.... ...1)	Reserved.

End Address + 1:
The virtual address of the byte immediately following the end of the virtual area.

SVC 122 (0A7A) ESR(type2)SVC - is type 2, gets LOCAL lock.

Routes control to type 2 extended supervisor service routines based on a routing code in register 15.

Code	Macro	Description
00		Reserved.
01		Reserved.
02		Reserved.
03		Reserved.
04		Reserved.
05	EVENTS	Calls module IEAVEVT0 (E.P. = IEAVET1). PLM is *OS/VS2 System Logic Library*.

R0 Bytes have meaning as follows:

Bytes	Meaning
0	Flag bits
	1... .... ENTRIES=n (create request); delete is requested if FC=5.
	.111 1111 Reserved.
1	Reserved.
2-3	Number of ENTRIES requested or zero.

R1 Address of the EVENT table if a delete is requested.

06	Reserved.
07	Reserved.
08	Reserved.
09	Reserved.

SVC 123 (0A7B) PURGEDQ macro - is type 2, gets LOCAL and DISP locks (for non-MVS/System Extensions), get only the DISP lock (for MVS/System Extensions), calls module IEAVEPD0 (E.P. = IGC123). APF protected. GTF data is:

R15 No applicable data.
R0 Parameter to be passed to the RMTR if the SRB is purged.
R1 Address of the parameter list.

SVC 124 (0A7C) TPIO macro - is type 1, gets LOCAL locks, calls module ISTAPC22. PLM is *OS/VS2 VTAM Logic*. GTF data is:

R15 No applicable data.
R0 Bytes have meaning as follows:

Byte	Meaning		
0	Flag bits		
	x...		Reserved.
	.1..		on LCPB indicates.
	Bits 2-7	**Code**	**Meaning**
		C0	Specific request.
		C4	Any request.
		C8	Open.
		CC	TPPOST.
		0F	CLOSE ACB.
		10	Session control request.
1-3	DEB address.		

R1 Work element address.

SVC 125 (0A7D) EVENTS macro - is type 1, gets LOCAL lock, calls module IEAVEVT0 (E.P. = IGC125). PLM is *OS/VS2 System Logic Library*. GTF data is:

R15 No applicable data.
R0 Bytes have meaning as follows:

Bytes	Meaning	
0	Flag bits	
	1...	WAIT=YES.
	.1..	WAIT=NO.
	..1.	ECB= address.
	...1 1111	Reserved.
1-3	Address of LAST= ENTRY or, address of ECB if ECB= specified.	

R1 Address of the EVENT table.

SVC 126 (0A7E) MSS Interface - is type 3, gets local and CMS locks, branches to ICB2SIOF, ICBVPI00, ICBVQM00, and ICBVSC00. PLM is *OS/VS2 MSSC Logic*. APF protected. GTF data is:

R15 and R0 No Applicable data.
R1 Contains address of the request block.

SVC 127 (0A7F) Reserved.

SVC 128 (0A80) Reserved.

SVC 129 (0A81) Reserved.

SVC 130 (0A82) RACHECK macro - is type 2, gets no lock, calls module ICHRCK00. GTF data is:

R15 and R0 No applicable data.
R1 Address of the parameter list.

SVC 131 (0A83) RACINIT macro - is type 2, gets no lock, calls module ICHRIN00. APF protected or RACF authorized. GTF data is:

R15 and R0 No applicable data.
R1 Address of the parameter list.

SVC 132 (0A84) RACLIST macro - is type 2, gets no lock, calls module ICHRSV00. APF protected or RACF authorized. GTF data is:

R15 and R0 No Applicable data.
R1 Address of the parameter list.

SVC 133 (0A85) RACDEF macro - is type 2, gets no lock, calls module ICHRDF00. APF protected or RACF authorized. GTF data is:

R15 and R0 No Applicable data.
R0 Address of the parameter list.

SVC 134 (0A86) Reserved.

SVC 135 (0A87) Reserved.

SVC 136 (0A88) Reserved.

SVC 137 (0A89) ESR macro (MVS/System Extensions only) - is type 6, gets no locks, calls module, IGC137. The routing code in register 15 identifies the type 6 SVC routine to be executed.

Code	Macro	Description
00	CALLDISP	Dispatcher call - calls module, IEAVEDS0.
01		Reserved.
02		Reserved.
03		Reserved.

SVC 138 (0A8A) Reserved.

SYSEVENT 27 HEX (REQPGDAT): Issued by SMF during step termination to obtain user paging data.

Note: This SYSEVENT is intended for use only by SMF because the related data fields in the OUSB and OUXB are in certain cases reset to 0.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID). Byte 3 contains the SYSEVENT code. **In addition for MVS/System Extensions Release 2**, byte 2, bit 0 contains 0 if this is the end of a job step (the paging data fields are reset); contains 1 if this is the end of an accounting interval (the paging data fields are not reset).

Register 1: Bytes 0-3 contain the address of an area where the paging data is to be stored.

SRM returns the following information:

Register 15: Byte 3 contains X'04' if data is lost due to the accumulation control block error; otherwise it contains X'00'.

Register 1: Contains the same address as input register 1.

Return area: Word 1 contains a count of non-VIO page-ins; word 2 contains a count of non-VIO page-outs; word 3 contains a count of non-VIO reclaims; word 4 contains a count of VIO page-ins; word 5 contains a count of VIO page-outs; word 6 contains a count of VIO reclaims; word 7 contains a count of pages swapped in; word 8 contains a count of pages swapped out; word 9 contains a count of swap-outs; word 10 contains a count of common area page-ins; word 11 contains a count of common area reclaims; word 12 contains a count of pages stolen; words 13 and 14 contain a count of processor page seconds **(not applicable for MVS/System Extensions)**. For MVS/System Extensions, word 13 contains a count of LPA page-ins; word 14 contains a count of LPA page reclaims; words 15 and 16 contain a count of processor page-seconds.

SYSEVENT 28 HEX (COPYDMDT): Issued to notify SRM that a DISPLAY command with the keyword DMN has been entered.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain ASID (or zero indicating current ASID); byte 3 contains the SYSEVENT code.

Register 1: Contains a pointer to a data area large enough to contain the maximum size domain table. ***Exception:*** For MVS/System Extensions Release 2, register 1 contains a pointer to the COPYD parameter list (mapped by the IHACOPYD macro).

SRM returns the following information: Register 1: contains the same address as input register 1.

Return area: word 1 contains a count of the domains in bytes 0-1; the remainder of the area contains a copy of the domain descriptor table. ***Exception:*** For MVS/System Extensions Release 2, the return area contains the initialized COPYD area.

SYSEVENT 29 HEX (DONTSWAP): Issued to notify SRM that the issuing address space must not be swapped out until an OKSWAP or an INITDET or an INITATT SYSEVENT has occurred.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID of the issuing address space or zero; byte 3 contains the SYSEVENT code.

SRM returns the following information:

Register 1: Byte 3 contains: X'08' if the request is not authorized or if the outstanding count of DONTSWAP requests has reached its maximum value; X'04' if the request is not for the current address space; X'00' if the request to mark the address space as "not swappable" was honored.

SYSEVENT 2A HEX (OKSWAP): Issued to notify SRM that the issuing address space may again be considered for swapping.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID of the issuing address space or zero; byte 3 contains the SYSEVENT code.

SRM returns the following information:

Register 1: Byte 3 contains: X'08' if the request is not authorized; X'04' if the request is not for the current address space; X'00' if the request to mark the address space as swappable was honored.

SYSEVENT 2B HEX (REQSWAP): Causes an address space to be swapped out. Issued, for example, when VARY storage wants to swap out the address space that occupies the storage to be taken offline.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of the ECB to be posted if a dependency exists on the requested swap (or zero if no dependency exists on the swap). An ECB can be specified only if the request is for the current address space.

SRM returns the following information:

Register 1: Byte 3 contains X'00' if the swap-out request was honored; X'04'if address space is nonswappable; X'0C' if the request was ignored because the address space was in the process of being swapped.

Post codes:

X'00' - Requested swap-out is complete

X'04' - Address space became non-swappable before it could be swapped out.

SYSEVENT 2C HEX (BRINGIN): Issued when a CANCEL command has been issued for a job. BRINGIN ensures that if a job has been swapped-out because of a resource shortage, the cancellation can take effect without waiting until the shortage is relieved.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

SRM returns the following information:

Register 1: Byte 3 contains: X'00' if the swap-in request was honored; X'08' if the request was ignored because the address space was in the process of being swapped.

SYSEVENT 2D HEX (WKLDINIT): Issued by measurement facility routines to request that SRM begin collecting workload activity data.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of a global, fixed data collection buffer.

SRM returns the following information:

Register 1: Bytes 0-3 contain X'00' if the data collection buffer was accepted.

Register 15: Byte 3 contains: 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.

SYSEVENT 2E HEX (WKLDCOLL): Issued by measurement facility routines at the end of a reporting interval to collect workload activity.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain an ASID (or zero, indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of a fixed buffer, into which the collected workload activity measurements are to be copied.

SRM returns the following information:

Register 1: Bytes 0-3 are unchanged.

Register 15: Byte 3 contains: 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 (for MVS/System Extensions Release 2, X'04' may also be returned if previously started workload activity data collection has been stopped because of an installation control specification change); X'40' if the data collection buffer has not been established.

LOGON should use the UADs and the VERIFYPG SYSEVENT for performance group number verification.

Note: This sysevent is issued only when the user has installed MVS/System Extensions Release 2 (5740-XE1) and TSO Command Package (5740-XT6).

SYSEVENT 3D HEX (NEWICS) applies to MVS/System Extensions Release 2: When the SET command is issued with the installation control specification option, IEEMB812 issues this SYSEVENT as part of establishing the new installation control specification.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain the ASID (or zero indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Bytes 0-3 contain the address of the new installation control specification tables to be used.

SRM returns the following information:

Register 1: Bytes 0-3 contain the address of the installation control specification tables to be freed.

Register 15: Byte 3 contains X'00' if the new installation control specification has been properly set; X'24' if a recoverable error occurred and the new installation control specification is not in effect.

SYSEVENT 3F HEX (CMDSTART) applies to MVS/System Extensions Release 2: Issued by the terminal monitor program whenever a TSO command is invoked.

Locks held on entry to SRM: Local.

Register 0: Bytes 0-1 contain the ASID(or zero indicating the current ASID); byte 3 contains the SYSEVENT code.

Register 1: Contains the address of a 12 byte parameter list in fixed storage. Byte 1 of this parameter list contains X'80' indicating that the command came from and "in-storage list" or X'00' if the command was entered from a terminal; the bytes 2-4 are reserved; and bytes 5-12 contain the command name.

Note: This sysevent is issued only when the user has installed MVS/System Extensions Release 2 (5740-XE1) and TSO Command Package (5740-XT6).

SYSEVENT 40 HEX (CMDEND) applies to MVS/System Extensions Release 2: Issued by the terminal monitor program (TMP) when a TSO command has completed processing and has returned control to the TMP.

Locks held on entry to SRM: Any locks may be held.

Register 0: Byte 3 contains the SYSEVENT code.

Register 1: Byte 0 contains X'80' if this command put the next command on an "in storage" list.

Note: This sysevent is issued only when the user has installed MVS/System Extensions Release 2 (5740-XE1) and TSO Command Package (5740-XT6).

Module Summary

For each module name prefix, (first three characters in the name of an object module), this summary identifies the corresponding component and program logic manual (PLM). Where the module name prefix is also a system message number prefix, the appropriate message manual is also identified.

- Components are identified by component number (5752-SC1xx for VS2 SCP components, 5744-Axx for emulators). Refer to the component summary (immediately following the module summary) for component names, component microfiche order numbers, and primary PLM order numbers.
- Publications are identified by order number.

AHL Components: 5752-SC1xx, xx=11. PLMs: SY28-0643. Message manual: GC38-1002.

AKJ Components: 5752-SC1xx, xx=T5. PLMs: SY28-0652.

AMA Components: 5752-SC1xx, xx=12,16. PLMs: SY28-0643. Message manual: GC38-1002.

AMD Components: 5752-SC1xx, xx=DE,D0,11,12,15,18. PLMs: SY26-3832, SY28-0643. Message manual: GC38-1002.

HEW Components: 5752-SC1xx, xx=04,05. PLMs: SY26-3814, SY26-3815. Message manual: GC38-1007.

HMA Components: 5752-SC1xx, xx=30. PLMs: SY28-0685. Message manual: GC28-0673.

HMB Components: 5752-SC1xx, xx=14. PLMs: SY28-0643. Message manual: GC38-1002.

IAS Components: 5752-SCIxx, xx=BH,B2. PLMs: SY24-6000.

IAT Components: 5752-SC1xx, xx=BA. PLMs: SY28-0612.

IBC Components: 5752-SC1xx, xx=I0,I1. PLMs: SY35-0005. Message manual: GC38-1005.

ICA Components: 5752-SC1xx, xx=I2. PLMs: SY35-0005.

ICB Components: 5752-SC1xx, xx=DR,DP. PLMs: SY35-0013. Message manual: GC38-1000.

ICG Components: 5752-SC1xx, xx=DQ,DT,CC. PLMs: SY35-0014, -0016. Message manual: GC38-1000.

IDA Components: 5752-SC1xx, xx=DA,DE,D6. PLMs: SY26-3825 SY26-3826 SY26-3834. Message manual: GC38-1000.

IDC Components: 5752-SC1xx, xx=DU, DK, SY35-0010. PLMs: SY35-0013. Message manual: GC38-1000.

IDD Components: 5752-SC1xx, xx=DG. PLMs: SY26-3834.

IDF Components: 5752-SC1xx, xx=22. PLMs: GY30-2000.

IEA Components: 5752-SC1xx, xx=B4, CH, CJ, CK, CL, CM, CP, CR, CU, CV, C3, C4, C5, C8, C9, D1, 02, 10. PLMs: SBOF-8210, SY26-3827, SY26-3834, SY28-0623, SY28-0678. Message manual: GC38-1002.

IEB Components: 5752-SC1xx, xx=SC1DN,UA,UG,UH,UJ,UK,U6,U7,U8,U9. PLMs: SY35-0005. Message manual: GC38-1005.

IEC Components: 5752-SC1xx, xx=CA,CC,C3,C6,D0,20. PLMs: SY26-3823, SY26-3832, SY26-3834, SY27-7246. Message manual: GC38-1002.

IED Components: 5752-SC1xx, xx=T3,T8,21. PLMs: SY28-0651, SY30-2040. Message manual: GC38-1002.

IEE Components: 5752-SC1xx, xx=B4,B5,B8,B9,CK,CM,CV,CX,CZ,C4,C5,T4,00,20. PLMs: SBOF-8210, SY28-0623, SY28-0651. Message manual: GC38-1002.

IEF Components: 5752-SC1xx, xx=B2,B3,B4,B5,B6,B7,B8,B9,DE,T4,00,01. PLMs: SBOF-8210, SY26-3825, SY28-0623. Message manual: GC38-1002.

IEH Components: 5752-SC1xx, xx=UC,UD,UF,UY,U0,U2,U3. PLMs: SY35-0005. Message manual: GC38-1005.

IEI (Message number prefix). Message manual: GC38-1002.

IEW Components: 5752-SC1xx, xx=C2,C7. PLMs: SBOF-8210. Message manual: GC38-1007.

IEZ Components: 5752-SC1xx, xx=B9. PLMs: SBOF-8210.

IFA Components: 5752-SC1xx, xx=02. PLMs: SBOF-8210. Message manual: GC38-1002.

IFB Components: 5752-SC1xx, xx=CD. PLMs: SY28-0678. Message manual: GC38-1045.

IFC Components: 5752-SC1xx, xx=CD. PLMs: SY28-0678. Message manual: GC38-1045.

IFD Components: 5752-SC1xx, xx=06. PLMs: SY28-0676. Message manual: GC38-1006.

IFF Components: 5752-SC1xx, xx=G0,07. PLMs: SY27-7241, SY27-7242, SY27-7260. Message manual: GC27-6974.

Technical Newsletter

This Newsletter No. GN28-4692
Date January 15, 1980
Base Publication No. GC28-0708-1
File No. S370-37
Prerequisite Newsletters/ Supplements GN28-2967 GN28-2984

OS/VS2 System Programming Library:
Debugging Handbook Volume 1

This newsletter contains replacement pages for *Debugging Handbook (Vol. 1)*.

Before inserting any of the attached pages into *Debugging Handbook (Vol. 1)*, read ***carefully*** the instructions on this cover. They indicate when and how you should insert the pages.

Pages to be Removed	Attached Pages to be Inserted*
Cover - Edition Notice	Cover - Edition Notice
xi - xii	xi - xii
4-21 - 4-22	4-21 - 4-22
5-3 - 5-4	5-3 - 5-4
5-21 - 5-22	5-21 - 5-22
5-29 - 5-30	5-29 - 5-30
5-33 - 5-36	5-33 - 5-36
5-47 - 5-48	5-47 - 5-48
5-51 - 5-52	5-51 - 5-52
5-59 - 5-60	5-59 - 5-60
5-61.1 - 5-62	5-61.1 - 5-62

*If you are inserting pages from different Newsletters/Supplements and ***identical*** page numbers are involved, always use the page with the latest date (shown in the slug at the top of the page). The page with the latest date contains the most complete information.

A change to the text or to an illustration is indicated by a vertical line to the left of the change.

Summary of Amendments

This technical newsletter contains information in support of the 3800 MVS enchancements and miscellaneous technical and editorial changes.

Note: *Please file this cover letter at the back of the base publication to provide a record of changes.*

IBM Corporation, Publications Development, Department D58, Building 706-2, PO Box 390, Poughkeepsie, New York 12602

Printed in U.S.A.

OS/VS2 System Programming Library: Debugging Handbook

Volume 1

GC28-0708-1
File No. S370-37

Includes Selectable Units:

Scheduler Improvements	VS2.03.804
Supervisor Performance #1	VS2.03.805
Supervisor Performance #2	VS2.03.807
Data Management	VS2.03.808
IBM 3800 Printing Subsystem	VS2.03.810
TSO/VTAM	VS2.03.813
Scheduler/IOS Support	VS2.03.816
Service Data Improvements	VS2.03.817
MSS Enhancements	5752-824
3838 Vector Processing Subsystem	5752-829
3895 Device Support	5752-830
System Security Support	5752-832
Dumping Improvements	5752-833
Attached Processor Support	5752-847
MVS Processor Support	5752-851
Hardware Recovery Enhancements	5752-855
Interactive Problem Control System	5752-857
TSO/VTAM Level 2	5752-858
Data Management Support	5752-860

Includes Program Product:

MVS/System Extensions	5740-XE1

IBM Corporation, Publications Development, Dept. D58, Bldg. 706-2, PO Box 390, Poughkeepsie, New York 12602

Second Edition (November, 1978)

This is a major revision of and obsoletes GC28-0708-0 and GC28-0751-0 incorporating changes released in the following Technical Newsletters and System Library Supplements:

Scheduler Improvements	VS2.03.804	SU4	(GN28-2678)
Supervisor Performance #1	VS2.03.805	SU5	(GN28-2680)
Supervisor Performance #2	VS2.03.807	SU7	(GN28-2704)
Data Management	VS2.03.808	SU8	(GN28-2679)
IBM 3800 Printing Subsystem	VS2.03.810	SU10	(GN28-2722)
TSO/VTAM	VS2.03.813	SU13	(GN28-2655)
Service Data Improvements	VS2.03.817	SU11	(GN28-2768)
3838 Vector Processing Subsystem	5752-829	SU29	(GC28-0925-0)
3895 Device Support	5752-830	SU30	(GC28-0800-0)
System Security Support	5752-832	SU32	(GC28-0845-0)
Dumping Improvements	5752-833	SU33	(GC28-0816-0)
MVS Processor Support	5752-851	SU51	(GD23-0025-0)
Hardware Recovery Enhancements	5752-855	SU55	(GC28-0891-0)
Interactive Problem Control System	5752-857	SU57	(GD23-0096-0)
TSO/VTAM Level 2	5752-858	SU58	(GD23-0048-0)
Data Management Support	5752-860	SU60	(GD23-0076-0)
MVS/System Extensions (program product)	5752-XE1	XE1	(SD23-0001-0)

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*, GA20-0001, for the editions that are applicable and current.

The JES3 information contained in this manual is applicable only if JES3 has been integrated into your system.

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 reader's 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.