GC35-0004-5
File No. S370-34

OS/VS

Systems

System Management Facilities (SMF)

VS1 Release 3.0
VS2 Release 2.0

Sixth Edition (December 1973)

This edition is a major revision of, and makes obsolete, *OS/VS System Management Facilities (SMF),* GC35-0004-4. Major technical and editorial changes are summarized under "Summary of Major Changes." Information about Release 2 of OS/VS2 is for planning purposes only until such time as support is available.

Technical changes to the text or figures are indicated by a vertical line to the left of the change.

This edition applies to Release 3.0 of OS/VS1 and to Release 2.0 of OS/VS2. It also applies to all subsequent releases until otherwise specified in new editions or technical newsletters. To determine whether this edition is up to date, refer to *IBM System/360 and System/370 SRL Newsletter,* GN20-0360. The information contained in this publication is subject to significant change. Any such changes will be published in new editions or technical newsletters.

Requests for copies of this or other IBM publications should be made to the IBM branch office that serves you.

A form for reader's comments appears at the back of this publication. Address any additional comments concerning the publication to IBM Corporation, Programming Publications, Department 27D, Post Office Box 1900, Boulder, Colorado 80302.

Preface

This publication provides installation managers, system programmers, and operators with the information required to plan for, install, and use SMF in both VS1 and VS2. Information is common to both systems unless specifically indicated otherwise.

This publication has the following major divisions:

- "Introduction," which introduces SMF records and exits and discusses the compatibility of SMF between VS1 and MFT, VS2 and MVT, and VS1 and VS2. It also describes the basic SMF functions and the relationship of SMF to the operating system and to user written exit routines.
- "Incorporating SMF into the System," which describes procedures for incorporating SMF into an operating system.
- "System Information and Requirements," which describes storage requirements, performance, and operational considerations such as IPL and data management procedures.
- "Report Programs," which describes sorting SMF records and designing report programs.
- "Exit Routines," which describes planning, writing, and testing user written exit routines.
- "Accounting Records," which fully describes the accounting records.
- "Data Set Activity Records," which fully describes the data set activity records.
- "Volume Records," which fully describes the volume records.
- "System Use Records," which fully describes the system use records.
- "Subsystem Records," which fully describes the subsystem records.
- "Appendix A: Field to Record Cross Reference," which lists all the fields in alphabetic order and gives the record types containing each field.
- "Index," which is a subject index to this publication.

Required Publications

The following publications are required for use with the book you are now reading:

- *OS/VS Message Library: VS1 System Messages,* GC38-1001, and *OS/VS Message Library: VS2 System Messages,* GC38-1002, which contain a listing and explanation of the messages issued by SMF.
- *OS/VS1 System Data Areas,* SY28-0605, and *OS/VS2 System Data Areas,* SY28-0606, which contain additional information on the contents of SMF records.

Related Publications

The reader should be familiar with the information presented in the following publications:

- *OS/VS Assembler Programmer's Guide,* GC33-4021, which describes the ASMFCL cataloged procedure, which is used to link-edit sample exit routines.

- *OS/VS1 JCL Reference,* GC24-5099, which describes the OUTLIM parameter, which is used in conjunction with a user written exit routine.
- *OS/VS2 JCL Reference,* GC28-0692, which describes the OUTLIM parameter, which is used in conjunction with a user written exit routine.
- *OS/VS Data Management Services Guide,* GC26-3783, which describes the record descriptor word (RDW) used to write records in the SMF data set.
- *OS/VS1 System Generation Reference,* GC26-3791, and *OS/VS2 System Generation Reference,* GC26-3792, which describe the system generation program used to include SMF and associated functions into the operating system, and JES2GEN parameters that relate to SMF.
- *OS/VS Utilities,* GC35-0005, which describes the IEBUPDTE and IEBDG utility programs, which are used to enter the SMFPRMxx data set into SYS1.PARMLIB and to generate samples of standard parameter lists for exit routines.
- *OS/VS1 Job Management Logic,* SY24-5161, and *OS/VS2 Job Management Logic,* SY28-0621, which describe the OUTLIM parameter.
- *OS/VS1 Storage Estimates,* GC24-5094, and *OS/VS2 Storage Estimates,* GC28-0604, which provide information on storage requirements.
- *Operator's Library: OS/VS1 Reference,* GC38-0110, and *Operator's Library: OS/VS2 Reference,* GC38-0210, which describe the SWITCH and HALT commands.
- *OS/VS Supervisor Services and Macro Instructions,* GC27-6979, which describes step priorities.
- *OS/VS Message Library: VS1 System Codes,* GC38-1003, and *OS/VS Message Library: VS2 System Codes,* GC38-1008 which describe system completion codes.
- *OS/VS1 RES: System Programmer's Guide,* GC28-6878, which describes Remote Entry Services.
- *OS/VS1 Planning and Use Guide,* GC24-5090, which provides information on handling accounting information when SMF=BASIC is specified.
- *OS/VS1 Virtual Storage Access Method (VSAM) Logic,* SY26-3841, and *OS/VS2 Virtual Storage Access Method (VSAM) Logic,* SY26-3825, which describes the format of a record in a VSAM data set.
- *OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide,* GC26-3838, which describes how to calculate the length of VSAM catalog records.
- *OS/VS2 System Programming Library: Initialization and Tuning Guide,* GC28-0681, which describes initialization parameters for JES2 which determines when SMF is recorded by JES2.

For information on the PL/I and Sort/Merge program products, you may refer to *PL/I Language,* SC33-0009 and to *Sort/Merge, SM/1, Programmer's Guide,* SC33-4007.

Notation Conventions

The format of the parameters and instructions shown in this publication is governed by the rules of notation discussed below.

Bold Type

Information in bold type (**NONE**, **NSL**, etc.) must be entered exactly as shown.

Italic Type

Information in italics (*xx*, *register address*, etc.) indicates data to be supplied by the user. In the following examples:

BUF=*n*
VOL=SER=*vol.ser.no.*

n is replaced by a number and *vol.ser.no.* is replaced by a volume serial number.

Special Characters

Special characters are used to indicate alternative items and required blank characters. Alternative items are separated by an OR sign (|). No more than one of the items separated by OR signs may be selected. In the following example:

1 | 2 | 3

only one of the values may be selected.

A required blank character is indicated by ␣. In the following example, a blank is required between DD and DUMMY:

DD␣DUMMY

Punctuation

The punctuation used in the commands (commas, semicolons, colons, and apostrophes) must be entered as shown.

Braces

Braces { } indicate a choice of entry. You must include one, and only one, entry. If there are several choices within braces, you may enter any one of the choices. The braces themselves are never entered. In either of the following examples:

{NO | YES} **{NO}**
{YES}

you must enter either the word **NO** or the word **YES**.

Brackets

Brackets [ ] indicate an optional value. The brackets themselves are never entered. In the following example:

KEYWORD=*value* [,*value2*]

value2 is optional and need not be entered.

Underscores

Underscores indicate the value that is assumed if no value is entered. Values that are assumed are called defaults. In the following example:

[YES | NO]

if no value is given, **YES** is assumed.

Contents

Figures

- VS1 has added an additional exit IEFU83.
- VS2 has added an additional exit IEFUSO.
- VS2 has added VSAM, System Reconfiguration, and replaced record types 1 and 12 by MF/1.
- The SMF parameter list name is changed to SMFPRM*xx*. The default is SMFPRM00.
- Keywords PRM and ALT are no longer valid SMF parameters. SYS1.MANX and SYS1.MANY must be cataloged.
- The keyword SID is expanded to 4 bytes to include the MDL function, which is no longer a valid SMF parameter.
- VS2 has integrated JES2 into the system.
- Record type 20 is changed from a data set activity record to a job accounting record.

SMF (System Management Facilities) is a feature of OS/VS that provides the means for gathering and recording information that can be used for billing customers or evaluating system usage. Information is gathered and recorded by SMF data collection routines and by user written exit routines. Because the data collection and exit routines are independent of one another, they can be used in combination or separately.

Note: SMF cannot be used for system tasks. In VS2 only, SMF cannot be used for problem programs started from the console.

SMF data collection routines gather several types of information:

- Accounting information, such as CPU time and device and storage usage.
- Data set activity information, such as EXCP count and the user of the data set.
- Volume information, such as the space available on direct access volumes and error statistics for tape volumes.
- System use information, such as system wait time and I/O configuration.
- Subsystem information, such as subsystem start and stop time.

The type of data to be collected can be modified by the operator at each initial program loading (IPL).

Through user written analysis and report routines, this information can be used in a variety of ways. For example, this information can be used to prepare customer's bills. The information might also be used to measure system usage against departmental standards of efficiency and performance.

SMF is not, however, confined to after-the-fact analysis. SMF allows you to write exit routines that can monitor a job or job step at various points during its processing cycle—from control statement analysis to termination of the job. Therefore, by adding installation routines at the appropriate exits, standards of identification, priority, resource allocation, and maximum execution time can be enforced.

Here's an example of using both facilities provided by SMF. By collecting and analyzing the information obtained by the data collection routines, the installation manager determines the average time each job step uses the CPU. In general, he finds that job steps exceeding this time limit are in a loop or unending wait state. Time is being wasted and overall efficiency impaired. Therefore, the average is used to establish a time limit through an exit routine for each job or job step running on the system; a job exceeding the expected time limit will be terminated. However, there must be some way to allow a job to exceed the expected time limit. Therefore, a routine is coded for the time limit exit. This routine allows the operator to extend the run time for selected jobs, such as the inventory program at year's end.

Data Collection

Various routines within the control program format SMF records and write them to the SMF data set. At IPL you can select which of certain groups of SMF formatted records are to be recorded through the use of SMF control parameters. (See "Selecting SMF Records Using SMFPRM*xx* Parameters" in the chapter "Incorporating SMF into the Operating System.") In addition to the records

supplied by SMF, you can create records to supplement or replace SMF records in the user written routines according to your record definitions and formats. (See "SMFWTM Macro Instruction" in the chapter "Exit Routines.")

The records can be grouped according to the type of information they contain, as follows:

- Accounting records, which describe for each job and job step (1) who used the system, (2) what was used in the system, (3) how much was used, and (4) the completion code. These records describe background jobs for both VS1 and VS2. They describe foreground jobs for VS2 only.
- Data set activity records, which describe the characteristics, activity, and user of data sets. These records also contain information about the deletion and renaming of data sets.
- Volume records, which describe the space available on direct access volumes and contain error statistics for tape volumes, and describe the data spaces that are recorded in a VSAM catalog.
- System use records, which describe the configuration of the system, give system statistics such as wait time and total paging statistics, describe SMF options in effect, and record the occurrence of certain events.
- Subsystem records, which describe the activities and events of the particular subsystem. These records contain information on (1) when the subsystem is started and stopped—date and time, (2) subsystem options, and (3) the occurrence of certain subsystem events. The records describing RES are available only in VS1 and those describing JES2 are available only in VS2.

Accounting Records

Accounting records describe how much a job or job step used the system. Some of the information contained in these records includes identification fields, accounting information from the JOB and EXEC statements, priority, CPU time (the time a job or job step actually uses the CPU), SYSIN and SYSOUT usage, device usage, and job or job step completion code. This type of information can be used to bill customers for use of the system.

Figure 1 lists the records included in the group of accounting records, describes when each record is written, and lists some of the information contained in each record.

See the chapter "Accounting Records" for more information on and the complete format of each of the accounting records.

Data Set Activity Records

Data set activity records describe the characteristics, activity, and user of data sets. Some of the information contained in these records includes data set names, volume serial numbers, number of volumes, and various control block fields. This information can be used by user written routines that report the data sets used by each job or job step.

Figure 2 lists the records included in the group of data set activity records, describes when each record is written, and lists some of the information contained in each record.

See the chapter "Data Set Activity Records" for more information on and the complete format of each of the data set activity records.

Record Type	When Written	Information Contained
4	After normal or abnormal termination of a job step for background jobs.	Job identification, time of day that certain events occur during step processing, step CPU time, amount of main storage allocated and used, devices used[1], step paging activity, completion code, step priority, step accounting data, termination indicator.
5	After normal or abnormal job termination for background jobs.	Job identification, time of day that certain events occur during job processing, job CPU time, completion code, job priority, job accounting data, termination indicator.
6	After processing of a SYSOUT class or form within a class for a background job has finished.	Writer start and end times, number of SYSOUT data sets within the class and form, number of logical records processed.
20	Each time a job is initiated.	Job identification, programmer's name, user identification, number and contents of accounting fields on JOB statement.
26	As a job is purged from the system in a VS2 system.	Job identification, time of day that certain events occur during job processing, total amount of SYSOUT output for the job, accounting information.
34	Each time a LOGOFF function processes a step termination for a foreground job in a VS2 system.	Job step information, such as LOGON time, number of TGETs and TPUTs issued, job step CPU time, completion code, and main storage used. (Similar to type 4, which is produced for background jobs.)
35	Each time a LOGOFF process has been completed for foreground jobs in a VS2 system.	Job information, such as LOGON time, number of TGETs and TPUTs, session completion code, LOGON priority, LOGON enqueue time, termination indicator, and session CPU time. (Similar to type 5, which is produced for background jobs.)
40	The dynamic allocation function processes a de-allocation, concatenation, or de-concatenation request in a VS2 system.	Device class, unit type, channel/unit address, and EXCP count.

[1] In VS2, you must also consider dynamic device allocation activity in record type 40.

Figure 1. Table of Accounting Records Showing When They Are Written and the Information They Contain

Record Type	When Written	Information Contained
14	A user's data set[1] on a tape or direct access device that was opened for INPUT or RDBACK is closed or processed by EOV.	Creation and expiration dates, device type, EXCP count, volume serial numbers, number of volumes, record format and length, and pertinent portions of system control blocks.
15	A user's data set[1] on a tape or direct access device that was opened for OUTPUT, UPDAT, INOUT, or OUTIN processing is closed or processed by EOV.	Same type as record 14.
17	A user's data set[1] is scratched.	Data set name, number of volumes, volume serial numbers.
18	A user's data set[1] is renamed.	Old data set name, new data set name, number of volumes, volume serial numbers.
62	At the successful or unsuccessful opening of a VSAM component.	The name of the catalog in which the component or cluster is defined and the volumes on which the catalog and the component or cluster are stored.
64	When a VSAM component or cluster is closed, when it becomes necessary to switch to another volume to continue processing, or when no more space is available on a volume. One record is written for each component closed. If a cluster is closed, one record is written for each component in the cluster.	The condition that caused the record to be written, identifies the volume on which the component is stored, extents of the component on the volume, and statistics about processing events that have occurred since the component was opened.
68	When a VSAM cluster or component is renamed.	The name of the VSAM catalog in which the component is defined and the old and new names.

[1] Record types 14-18 are produced only for non-VSAM data sets where as types 62-68 are produced only for VSAM data sets.

Figure 2. Table of Data Set Activity Records Showing When They Are Written and the Information They Contain

Volume Use Records

Volume records describe the space available on direct access volumes, give error statistics for tape volumes, and describe data spaces in a VSAM catalog. The tape information can be used by IFHSTATR or by user written routines that address problems of volume deterioration. (See "IFHSTATR" in *OS/VS Utilities*, GC35-0005.) The direct access volume information can be used by user written routines that address space fragmentation.

Figure 3 lists the records included in the group of volume records, describes when each record is written, and lists some of the information contained in each record.

Record Type	When Written	Information Contained
19	For each direct access device on line at IPL and when a HALT or SWITCH command is processed, and for any direct access device when it is demounted.	Number of unused alternate tracks, number of unallocated cylinders and tracks, number of cylinders and tracks in the largest free extent, owner identification number.
21	When a user data set on magnetic tape is closed or processed by End-of-Volume.	Volume serial number, channel/unit address, number of read and write errors.
69	When data space is defined, extended, or deleted.	Information about the catalog in which the data space is defined, the volume on which it is allocated, the number of free data space extents on that volume, and the amount of available space.

Figure 3. Table of Volume Records Showing When They Are Written and the Information They Contain

See the chapter "Volume Records" for more information on and the complete format of each of the volume records.

System Use Records

System use records describe the system configuration and SMF options in effect, give system statistics (such as system wait time and paging statistics), and record the occurrence of certain events. Some of the information contained in these records includes system identification, SMF options, number of bytes in both virtual and real storage, system wait time, paging statistics, and the I/O configuration. Some of the SMF events whose occurrences are recorded are the beginning and ending of a dump of the SMF data set and the beginning and ending of a period of time when data is not being recorded. This type of information can be used by user written programs that report system efficiency, performance, and usage.

Figure 4 lists the records included in the group of system use records, describes when each record is written, and lists some of the information contained in each record.

Record Type	When Written	Information Contained
0	During system initialization after IPL.	Real and virtual storage size and SMF options in effect.
1	At SMF initialization and at the first job step termination following the expiration of a ten-minute interval of elapsed system time, in a VS1 system.	CPU wait time, system paging statistics accumulated during all of the ten-minute intervals that expired since the last type 1 record was written, the expiration time of the last ten-minute interval.
2	At the beginning of a dump data set.	System identification and the time and date the record was moved to the SMF buffer. (This record is the standard record header.)
3	At the end of a dump data set.	Same type as 2.
7	After any period when there was no SMF data set available for recording. This is the first record created when an SMF data set again becomes available.	Count of SMF records generated but not written and the start and end times of the period during which no records were written.
8	During system initialization after IPL.	Descriptions of each online device at IPL. (Each entry description includes the device class, unit type, and channel/unit address.)
9	During processing of the VARY operator command which results in a device being brought online.	Identification of the device added to the configuration.
10	After a device is added to the configuration.	Identification of the device made available by device class, unit type, and device address. Job requiring the allocation is identified.
11	During processing of the VARY operator command which results in a device being taken offline.	Identification of the device removed from the configuration.
12	During processing of HALT or SWITCH operator commands, in a VS1 system.	System wait time and paging statistics since the last record type 1 and the time this record was built.
13	At IPL and after each DEFINE command is processed under VS1 only.	The size of the storage range assigned to each partition.

Figure 4. Table of System Use Records Showing When They Are Written and the Information They Contain (Part 1 of 2)

Record Type	When Written	Information Contained
22	Whenever there is an IPL or a VARY of a CPU, Storage range, or Channel under VS2.	Identification, model information, and address.
31	Whenever the Terminal Input/Output Controller (TIOC) initialization routine is entered as the result of a MODIFY TCAM command under VS2 only.	Input/output control initialization information, such as the total number and size of time sharing buffers, number of buffers per user and number of buffers reserved on the free queue.
70	At the end of the collection period when the CPU activity option of MF/1 is selected under VS2.	CPU identification and wait time.
71	At the end of the collection period when the Paging activity option of MF/1 is selected under VS2.	Counts of system paging activities in different sections of the system.
72	At the end of the collection period when the Workload activity option of MF/1 is selected under VS2.	Performance group number and transaction information about the group.
73	At the end of the collection period when the Channel activity option of MF/1 is selected under VS2.	Channel identification, SIO counts, and sample counts of burst mode and channel busy with CPU in WAIT state.
74	At the end of the collection period when the Device activity option of MF/1 is selected under VS2.	Device address, UCB type, volume serial number, use count, and sample count of device being busy.

Figure 4. Table of System Use Records Showing When They Are Written and the Information They Contain (Part 2 of 2)

See the chapter "System Use Records" for more information on and the complete format of each of the system use records.

Subsystem Records

Record types 43-45, and 47-49 are subsystem records. Some information contained in these records is common, such as system indicator, subsystem identification, and CPU identification. The other information contained in the records is pertinent to the record type and the subsystem for which it is written. For example, record type 43 (JES2 Start), hereafter referred to as 43H, and record type 43 (RTAM Start), hereafter referred to as 43R, contain common headings. But unique subsystem data, such as JES2 options in 43H and the RTAM start procedure in 43R, is contained in a particular subsystem record.

Some of the common data contained in these records includes the record type, date, time, system identification, and subsystem identification. Some of the events whose occurrences and data are recorded are sign-on and sign-off of a remote user, start-line, stop-line, log-on, log-off, modification type, and number of lines modified. This information can be used by the user written programs that report the activity of the subsystems.

Figure 5 lists the records included in the group of subsystem records, describes when each record is written and lists some of the information contained in each record.

Record Type	When Written	Information Contained
43H	Whenever a START JES2 command is entered under VS2 only.	JES2 information including subsystem identification and JES2 options.
45H	Whenever a STOP JES2 command is processed under VS2 only.	JES2 information including subsystem identification, date, time, and completion code.
47H	Whenever an operator enters a Start Line command or when a sign-on is received from a remote user, in VS2 only.	JES2 information including subsystem identification, subsystem event, remote name, line name, password, and signon card information.
48H	Whenever an operator enters a Stop Line command or when a sign-off is received from a remote user in VS2 only.	JES2 information including subsystem identification, subsystem event, remote name, line name, password, number of EXCPs, number of line errors, and line adapter address.
49H	Whenever a sign-on is issued with an invalid password.	JES2 information including subsystem identification, subsystem event, remote name, line name, password, and signon card information.
43R	During RTAM initialization under VS1 only.	RES information including name of RTAM start procedure, maximum numbers of readers and writers, number of entries in LINE table, number of line DCTs, number of lines to activate, line names, and unit addresses.
44R	Whenever a MODIFY RTAM command is issued under VS1 only.	RES information including name of start procedure, type of MODIFY, number of lines modified, line numbers, and unit addresses.
45R	When a STOP RTAM command is issued under VS1 only.	RTAM information including name of RTAM STOP procedure, stop status, and number of lines started when STOP was received.
47R	Whenever a valid LOGON record is received by RTAM under VS1 only.	RES information including QID entry, passback area, and LOGON record.
48R	Whenever a LOGOFF record is received by RTAM under VS1 only.	RES information consisting of the QID entry.
49R	Whenever an invalid LOGON record is received by RTAM under VS1 only.	RES information including the QID entry, passback area, and LOGON record.

Figure 5. Table of Subsystem Records Showing When They Are Written and the Information They Contain

See the chapter "Subsystem Records" for more information on and the complete format of each of the subsystem records.

SYSOUT Messages

In addition to the records written to the SMF data set, SMF writes four messages to the SYSOUT data set. These messages are assigned message numbers IEF373I, IEF374I, IEF375I, and IEF376I, and they indicate the start and end times for each job step and for each job. The text of these messages and an explanation of each is provided in *OS/VS Message Library: VS1 System Messages,* GC38-1001, and *OS/VS Message Library: VS2 System Messages,* GC38-1002.

User Written Routines

Your installation should provide two types of routines to take full advantage of the features of SMF:

- Analysis and report routines that process and format information contained in the SMF and installation defined data sets. These routines may produce billing reports, list the SMF data set, use a sort/merge program to re-order the data, or perform detailed analysis operations.
- Exit routines that periodically monitor jobs and can write user records to the SMF or installation defined data set.

SMF provides exits in the control program that can be used by user written routines. User written routines can monitor each job at specific points from the time it is encountered in the input stream to the time all spooled output has been written. The SMF writer exit is taken for each SMF record generated; it is not job related. These routines are referred to as exit routines.

Like execution time for any other part of the control program, the execution time of the exit routines is added to system overhead and will degrade system throughput. The amount of the degradation depends on the length of the routines and the number of times each is performed during processing of a job. The advantages of including exit routines must be weighed against the factors affecting system throughput when choosing which exits to use. It is possible (by the SMFPRM*xx* parameters) to specify at IPL the suppression of all exits or only step-related exits, permitting the system to operate without the exit routines.

In VS2, the job related exits are taken by jobs in either the foreground or background.

An installation can make use of any or all of these exits by providing user written exit routines and including them in the appropiate system library (See Figure 10) before system generation or in the link library SYS1.LINKLIB and SYS1.NUCLEUS for VS1 or in SYS1.LPALIB for VS2 after system generation. The installation does not need to supply dummy exits for those not being replaced.

The user written exit routines can perform functions such as: cancel jobs, write user defined records to the SMF data set, access installation defined data sets, or enforce installation standards, such as identification, priority, and resource allocation. Because these routines become part of the control program where errors can cause repeated system failure, thorough debugging is important. For more information on testing and debugging, see "Testing Exit Routines" in the chapter "Exit Routines."

The formats of the parameters passed to each exit routine are described in the chapter "Exit Routines." The procedure for adding user written routines to the system is described in the chapter "Incorporating SMF into the Operating System".

Exits Available to Both VS1 and VS2

The exits available for use under both VS1 and VS2 are:

- Job Validation (IEFUJV), which receives control from the job management routine of the control program before each job control statement (or cataloged procedure) encountered in the input stream is interpreted. One final entry is made after all of the JCL is interpreted. This exit is not taken for comment statements or, in VS2, for jobs started from the console. A return code from this exit specifies whether processing of this job is to continue.
- Job Initiation (IEFUJI), which receives control from the initiator routine of the control program when a job on the input queue is selected for initiation. A return code from this exit specifies whether the job is to be started or canceled.
- Step Initiation (IEFUSI), which receives control from the initiator before each job step is started (prior to allocation). A return code from this exit specifies whether the step is to be started or the job canceled.
- Time Limit (IEFUTL), which receives control from the timer interruption handler whenever one of the following time limits expires: the job CPU time limit from the JOB statement, the step CPU time limit from the EXEC statement or reader procedure, or the continuous wait time limit for the job from SMFPRM*xx*. A return code from this exit specifies whether the job step is to be terminated or processing continued with a new time limit.
- SYSOUT Limit (IEFUSO), which, in VS1, receives control from the I/O supervisor and in VS2 receives control from JES2, when the number of logical records written to a SYSOUT data set exceeds the output limit for the data set. A return code from this exit specifies whether the job is to be terminated or processing continued using a new limit.
- SMF Record (IEFU83), which receives control from the SVC83 routine before each record is written to the SMF data set. A return code from this exit specifies whether or not to suppress the current SMF record.
- Termination (IEFACTRT), which receives control from the terminator on the normal or abnormal termination of each job step and job. A return code from this exit specifies whether the job is to be continued or terminated (for job step entry only), and whether the SMF record is to be written or skipped. With SMF=BASIC in VS1 this routine also receives control at the beginning of allocation. There are no SMF records to be considered with SMF=BASIC.
- Job Purge (IEFUJP), which receives control from the SYSOUT writer routine for all jobs in a VS1 system or from the HASPACCT routine in a VS2 system when a job not started from the console is ready to be purged from the system (after the job has terminated and all SYSOUT output that pertains to that job has been written). This exit does not return a code to the control program in a VS1 system. In a VS2 system, this exit does return a code which specifies whether the SMF record is to be written or skipped.

Exit Available to VS1 Only

The exit available for use under VS1 only is:

- Input Stream Validation (IEFUIV), which receives control from the input stream control routine every time a new job is encountered in the input stream when VS1 is used. A return code from this exit specifies whether a job is to be accepted for processing.

SMF Operation

Figure 6 shows SMF incorporated into the operating system. The following paragraphs, which describe Figure 6, assume the user written exit routines are supplied for all SMF exits [1], all SMF-formatted records[2] are written to the SMF data set, and user written analysis and report routines are supplied. In any real application, of course, the exit routines that are supplied and the records specified to be written to the SMF data set depend on the installation's requirements.

Initializing

At IPL, an SMF initialization routine receives control and reads in the member (SMFPRM*xx*) of SYS1.PARMLIB that contains SMF control parameters. The options specified in these parameters, including such options as whether both the data collection routines and the control program exits will be active, will be typed out at the console if the operator is allowed to modify them for the current work day. (The default parameters can be permanently changed by replacing the member in SYS1.PARMLIB.) Before job processing begins, the SMF initialization routine records information about IPL, initial input/output device configuration, for VS1 only, the amount of storage assigned to each partition, and, for VS2 only, the configuration of CPUs, storage, and channels.

Job Processing

As the input stream in VS1 is read in for processing, each new job causes control to pass to an input stream validation exit. This exit routine may verify whether the job is to be accepted for processing. Before each job control statement is interpreted (VS1) or converted (VS2) control is passed to a job validation exit routine. This routine may verify any fields in the JCL statement, modify JCL, or reject jobs that do not meet installation standards. After all JCL has been interpreted, the same job validation exit routine receives control for further validity checking.[3]

Before a job is initiated, the initiator routines pass control to a job initiation exit routine. In this routine, the user can decide whether to cancel or continue job processing based on accounting parameters associated with the job. Upon return from the routine, the SMF job commencement record is created and written to the SMF data set.

Before a job step is initiated, control is passed to a step initiation exit routine. Here, processing similar to that done in the job initiation exit can be performed.

When a step within the job ends either normally or abnormally, terminator routines create a job step record and, before the record is written, pass control to an accounting exit routine. The accounting exit routine may modify the SMF records, add a record to its own accounting data set, or add records to the SMF data set. This accounting exit routine also indicates whether the job is to continue (if a job step termination is being processed) and whether the job step termination record is to be written. Upon return, the job step record is written to the SMF data set unless the exit routine specifies that it be suppressed.

---

[1] Note that one exit is available only when VS1 is used.

[2] The SMF records written under VS1 and VS2 are different in some cases.

[3] In VS2, control is passed to the job validation exit routine after all statements are processed first at the end of conversion and second at the end of interpretation.

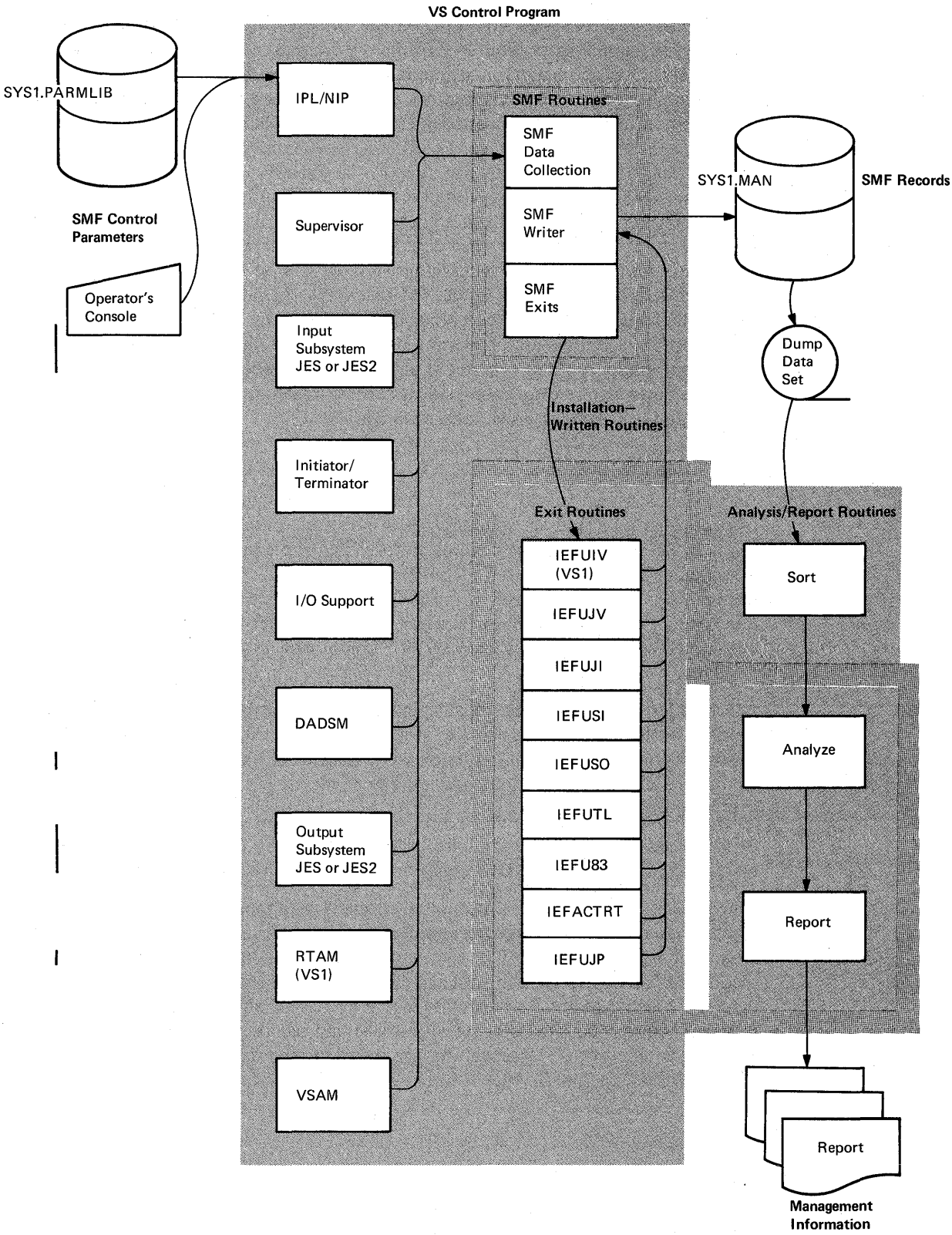

Figure 6. SMF in the Operating System

At job termination, SMF creates a job information record and again passes control to the accounting exit routine. Upon return from this routine, the SMF job termination record is written unless the exit routine specifies that it be suppressed.

After the job has terminated and all SYSOUT output that pertains to a job and all output writer records have been written, control is passed to the final exit routine, job purge. Upon return from the VS2 job purge exit routine the SMF job purge record is written.

Event Recording

Some types of information are recorded whenever particular situations arise. These types of information and the situations that cause their recording are:

- Data set information, which is recorded whenever a data set opened by a user program is scratched, renamed, closed, or processed by end of volume (EOV).
- Direct access volume information, which is recorded (1) for online, direct access devices at IPL, (2) when a volume is demounted, and (3) for online, direct access devices when a HALT or SWITCH command is issued.
- Configuration information, which is recorded when a DEFINE, or VARY command is issued and after allocation recovery.
- System statistics, which are recorded at the end of specific lengths of time, and in VS1, at the end of the day.

Just as some types of information are recorded whenever particular situations arise, some types of exits are entered whenever particular situations arise. These types of exits and the situations that cause them to be entered are:

- Time limit exit, which is entered whenever the step CPU, job CPU, or continuous wait time limits are reached.
- SYSOUT limit exit, which is entered whenever the OUTLIM limit is reached.
- SMF record exit, which is entered whenever an SMF record is ready to be written.

Dumping

Records are written to the primary SMF data set (initially SYS1.MANX) until the end of the allocated extent is reached. When the end of the allocated extent is reached, SMF opens the alternate SMF data set (initially SYS1.MANY) and continues recording. The operator is then notified to use the SMF dump program to copy SYS1.MANX to a dump data set. The data set in which recording is being done is called the active data and the other is call the inactive data set.

Note: If the operator failed to dump the requested SMF data set within a reasonable period of time, it is possible for the alternate SMF data set to also become full. When this occurs and the SMF buffer also fills, SMF will be in a data lost condition (not able to record) until dumping takes place.

Similar recording and copying operations continue throughout the work day, with SMF adding special records whenever a VARY command or allocation recovery changes the system configuration and, at the end of specific intervals, to record system statistics, such as wait time. The HALT EOD and SWITCH SMF command from the operator cause the system statistics to be recorded and the SMF buffers to be emptied into the active SMF data set. This active data set is then closed and the previously inactive data set is made active and selected for recording. The SMF dump program may then be used to copy the inactive SMF data set to the dump data set, which contains a complete history of the day's processing. The dump data set can serve as input to the user written analysis and report routines, which may be executed as ordinary problem programs under the operating system.

Post Processing

A user written routine may be used to list and total the system usage by account number. A separate analysis program can process the SMF records in the order recorded, to detect excessive system wait time or inefficient use of input/output devices. This information can lead to improved system throughput by suggesting changes in the job mixture or device allocation.

Compatibility

The compatibility between SMF under VS1 and MFT, VS2 and MVT, and VS1 and VS2 is discussed in the following sections.

SMF Compatibility Between VS1 and MFT

VS1 SMF is compatible with MFT SMF with the following exceptions:

- SYS1.MANX must not reside on tape for VS1 SMF.
- Three new exits, IEFUIV, IEFUJP, and IEFU83 have been added for VS1 SMF.
- The value received in fields such as CPU time, storage allocated and used, and EXCP counts field when operating under VS1 may differ from the value received when operating under MFT.
- Several record types have been modified for VS1 SMF. Figure 7 lists the record types and the corresponding modifications.
- The RES records 43R-45R and 47R-49R have been added for VS1.
- The VSAM records 62, 64, 68, and 69 have been added for VS1.
- Record type 20 is now a job accounting record.
- The SMF PARMLIB member is named SMFPRM*xx*.
- The SMF parameters PRM, ALT, and MDL are not supported. SID is a four-byte field.
- SYS1.MANX and SYS1.MANY must be cataloged.

Record Types	Modifications
Header	System indicator added.
0	Size of virtual storage replaces size of main storage. Size of real storage added.
1	Paging statistics and time of end of collection period added.
4	Partition size replaces allocation for hierarchy 0. Storage used replaces hierarchy 0 storage used. Reserved fields replace hierarchy 1 fields. No device entry for spooled data sets. Step termination indicators and paging statistics added.
5	In job termination indicator, the ABEND bit turns on when any step abnormally terminates. The job completion code field contains the ABEND code for the last step that abnormally terminated, regardless of normal processing by successive steps. Reserved field replaces checkpoint/restart field. A user's logon identifier has been added.
6	A user's logon identifier has been added.
12	Paging statistics and time of end of collection period added.
13	Entry size increased by 12 bytes.

Figure 7. Record Modifications from MFT for VS1

SMF Compatibility Between VS2 and MVT

VS2 SMF is compatible with MVT SMF with the following exceptions:

- SMF is standard in VS2.
- SYS1.MANX must not reside on tape for VS2 SMF.
- Two new exits, IEFU83 and IEFUJP, have been added for VS2 SMF.
- The SMF options for foreground jobs are specified in SMFPRM*xx* and not specified seperately.
- Record type 26 has been added in a VS2 system.
- The IEFUTL exit is taken for both foreground and background jobs and has an extension expressed in seconds.
- The extension of continous wait time is changed. Each step begins with the limit as specified by the SMFPRM*xx* JWT parameter. In VS2, each extension resets the limit for the step versus, only for that single wait.
- The JES2 subsystem records 43H, 45H, 47H, 48H, and 49H have been added to VS2.
- The VSAM records 62, 64, 68, and 69 have been added for VS2.
- The MF/1 records 70-74 have been added for VS2.
- The value received in fields such as CPU time, storage allocated and used, and EXCP counts field when operating under VS2 may differ from the value received when operating under MVT.
- Several record types have been modified for VS2 SMF. Figure 8 lists the record types and the corresponding modifications.
- The IFASMFR macro is multi-language (ASM and PL/S).
- Record type 20 is now a job accounting record.
- The SMF PARMLIB member is named SMFPRM*xx*.
- The SMF parameters PRM, ALT, and MDL are not supported. SID is a four byte field.
- SYS1.MANX and SYS1.MANY must be cataloged.
- Record types 1 and 12 are not supported in VS2.
- Record type 22 is added for configuration of CPUs, storage boxes, and channels.
- TSO system-related record types 30, 32, 33, 38, 41, and 42 are not produced in VS2.
- The IEFUSO exit interface does not include a DCB pointer.

Record Types	Modifications
Header	System indicator added.
0	Size of virtual storage replaces size of main storage. Size of real storage added.
4	Region size replaces allocation for hierarchy 0. Storage used is given in two parts: one from the top of the problem program area and a second from the bottom of the problem program area. Reserved fields replace hierarchy 1 fields. Step termination indicators and paging statistics added. CPU time is recorded seperately for execution under service request block versus task control block.
5	Reserved field replaces checkpoint/restart field. Service units, total transaction time, and performance group members are added. The CPU time is changed as in record type 4. SYSOUT indicators are deleted.
6	Additional JES2 fields have been added.
9 and 11	No longer have MP section.
31	Delete two fields: number of users that constitute slack time and logged-on user change.
34	Same changes as record type 4, replaced main storage occupancy time by TOD step initiated.
35	Service units, transaction active time, number of transactions, and performance group members are added. The CPU time is changed as in record type 4. SYSOUT indicators are deleted.

Figure 8. Record Modifications from MVT for VS2

SMF Compatibility Between VS1 and VS2

VS1 SMF is compatible with VS2 SMF with the following exceptions:

- For VS1, SMF is optional; for VS2, SMF is standard.
- In VS2, SMF can be used on both batch (background) jobs and time sharing (foreground) jobs entered from a terminal with the time sharing option (TSO); in VS1, SMF can be used only on batch (background) jobs.
- The IEFUIV exit is available only in VS1.
- Record types 1, 12, 13, 43R-45R, and 47R-49R are written only in VS1 systems; record types 22, 26, 31, 34, 35, 40, 43H, 45H, 47H-49H, and 70-74 are written only in VS2 systems.
- Several record types have additional fields for VS2. Figure 9 lists the record types and the corresponding modifications.
- IEFUJP exit in VS1 points to accounting information; IEFUJP exit in VS2 points to record type 26.
- The length of common exit parameter area is 72 bytes in VS1, and 36 bytes in VS2.
- The extension of continous wait time is changed. Each step begins with the limit as specified by the SMFPRM*xx* JWT parameter. In VS2, each extension resets the limit for the step versus, only for that single wait.
- SMF exits are not taken for problem programs started from the console in VS2.
- SMF data is not collected for problem programs started from the console in VS2.
- In VS2, the IFASMFR macro is multi-language (ASM and PL/S).
- In VS2, the IEFUTL exit can have time extensions expressed in seconds.
- In VS1, the IEFUSO exit interface includes a DCB pointer.

Record Types	Modifications
4	Storage used in VS2 is given in two parts: one from the top of the problem program area and a second from the bottom of the problem program area. In VS2, CPU time is recorded seperately for execution under service request block versus task control block. Additional paging statistics are added in VS2.
5	A user's logon identifier has been added in VS1. Service units, total transaction time, and performance group members are added in VS2. The CPU time is changed as in record type 4. SYSOUT indicators are deleted in VS2.
6	A user's logon identifier has been added in VS1. In VS2, additional JES2 fields have been added.

Figure 9. Record Modifications from VS1 for VS2

SMF is specified when you generate your VS1 operating system. SMF is a standard function in a VS2 system.

For VS1, SMF is an optional part of the system. To add SMF to your VS1 operating system, you must include SMF in the system generation statements.

If you are going to record data in either VS1 or VS2, you must define your use of SMF either in SMFPRM*xx* or from the operator's console, allocate direct access space for the SMF data set, catalog the SMF data sets, and add any user written exit routines you want to the control program. The information needed to add SMF to the system is given in the topics that follow.

Including Exit Routines in the System

Exit routines are an optional feature of SMF. If you decide to use exit routines, we suggest you incorporate them into system librarys before you generate your system. Otherwise, they must be link-edited into the appropriate system load module.

Note: If an exit routine is link-edited into an active system, a link-edit failure may render inoperative the load modules into which the exit was being link-edited. This is particularly important in the case of termination exits, because a failure would result in the loss of the output that indicates the cause of failure.

Figure 10 shows the distribution libraries to use to add exit routines prior to system generation. The System Release Guide for the current release has the most up-to-date information on distribution libraries. If you do not require all exit routines in your system, do not replace those you do not need.

If you wish to add or replace exit routines after system generation, you must link-edit the routines into the appropriate load module. Figure 11 shows the load module assignments. Be aware that in VS2 the load modules are a part of LPALIB and your newly link-edited modules will not be used until action occurs, such as formatting the page data set. A similar condition occurs in VS1 if the modules have been made resident.

When adding exit routines after system generation in either VS1 or VS2, refer to your system generation listing for exact load module names, aliases, and link-edit parameters.

Figure 12 shows the JCL required to add exit routines to SYS1.LINKLIB after generating a VS1 system.

Figure 13 shows the JCL required to add the exit routine IEFUTL and IEFU83 to SYS1.NUCLEUS after generating a VS1 system.

Figure 14 shows the JCL required to add exit routines to SYS1.LPALIB after generating a VS2 system.

Object module for this exit routine	Replace dummy exit in VS1 library	Replace dummy exit in VS2 library
IEFUIV	AOS00	Not applicable
IEFUJV	AOS00	AOSB3
IEFUJI	AOS00	AOSB3
IEFUSI	AOS00	AOSB3
IEFUTL	AOS00	AOSB3
IEFUSO	AOS00	AOSH1[1]
IEFU83	AOS00	AOS00
IEFACTRT	AOS00	AOSB3
IEFUJP	AOS00	AOSH1[1]

[1] There is no dummy IEFUSO or IEFUJP exit distributed in VS2.

Figure 10. Distribution Libraries for Adding Exit Routines Prior to System Generation.

Object module for this exit routine	Must be link-edited into this load module in VS1	Must be link-edited into this load module in VS2
IEFUIV	IEFJES in SYS1.LINKLIB	Not applicable
IEFUJV	IEFUJV in SYS1.LINKLIB	IEFUJV in SYS1.LPALIB
IEFUJI	IEFSD162 in SYS1.LINKLIB	IEFSD060 in SYS1.LPALIB
IEFUSI	IEFSD162 in SYS1.LINKLIB	IEFSD060 in SYS1.LPALIB
IEFUTL	IEANUC01 in SYS1.NUCLEUS	IEFSD060 in SYS1.LPALIB
IEFUSO	IEFJES in SYS1.LINKLIB	IEFUSO in SYS1.LPALIB[3]
IEFU83	IEANUC01 in SYS1.NUCLEUS	IEFU83 in SYS1.LPALIB
IEFACTRT	IEFSD161 in SYS1.LINKLIB[1] IEFW21SD in SYS1.LINKLIB[2]	IEFW21SD in SYS1.LPALIB
IEFUJP	IEFJES in SYS1.LINKLIB	IEFUJP in SYS1.LPALIB[3]

[1] IEFACTRT must be link-edited into this load module if either SMF=BASIC or SMF=FULL is specified.

[2] IEFACTRT must also be link-edited into this load module if SMF=BASIC is specified.

[3] There is no dummy IEFUSO or IEFUJP exit distributed in VS2.

Figure 11. Required Load Module Assignments for Exit Routines

Note: You must refer to your system generation listing for exact load module names, aliases, and link-edit parameters. Link-edit parameters must be specified according to the characteristics of your exits.

```
//EXITLNK JOB 123456,SMITH
//STEP1   EXEC  PGM=IEWL,PARM=(link-edit  parameters)
//SYSPRINT DD   SYSOUT=A
//SYSLMOD  DD   DSNAME=SYS1.LINKLIB,DISP=(OLD,KEEP)
//SYSUT1   DD   UNIT=SYSDA,DISP=(,DELETE),SPACE=(TRK,(20,5))
//SYSLIN   DD   *
```

(IEFUJV object deck)

```
  ENTRY     IEFUJV
  INCLUDE   SYSLMOD(IEFUJV)
  NAME      IEFUJV(R)
```

(IEFUJI and IEFUSI object decks)

```
  ENTRY     IEFSD062
  INCLUDE   SYSLMOD(IEFSD162)
  ALIAS     aliasname1,aliasname2,...
  NAME      IEFSD162(R)
```

(IEFACTRT object deck)[1]

```
  ENTRY     IEFSD061
  INCLUDE   SYSLMOD(IEFSD161)
  ALIAS     aliasnameA,aliasnameB,...
  ALIAS     aliasnameX,aliasnameY,...
  NAME      IEFSD161(R)
//STEP2  EXEC  PGM=IEWL,PARM=(link-edit  parameters)
//SYSPRINT DD   SYSOUT=A
//SYSLMOD  DD   DSNAME=SYS1.LINKLIB,DISP=(OLD,KEEP)
//SYSUT1   DD   UNIT=SYSDA,DISP=(,DELETE),SPACE=(TRK,(20,5))
//SYSLIN   DD   *
```

(IEFUSO, IEFUIV, and IEFUJP object deck)

```
  ENTRY     IEFJESCT
  INCLUDE   SYSLMOD(IEFJES)
  NAME      IEFJES(R)
/*
```

[1] When SMF=BASIC is specified, IEFACTRT must also be link-edited in load module IEFW21SD. The required statements for this additional link-edit are as follows:

```
ENTRY       IEFJESCT
INCLUDE     SYSLMOD(IEFW21SD)
ALIAS       aliasname1,aliasname2,...
NAME        IEFW21SD(R)
```

Figure 12. JCL for Adding Exit Routines to SYS1.LINKLIB After Generating a VS1 System

Note: You must refer to your system generation listing for exact load module names, aliases, and link-edit parameters. Link-edit parametes must be specified according to the characteristics of your exits.

```
//NUCLINK JOB 123456,SMITH
//       EXEC  PGM=IEWL,PARM=(link-edit  parameters)
//SYSPRINT DD  SYSOUT=A
//SYSLMOD  DD  DSN=SYS1.NUCLEUS,DISP=OLD
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSLIN   DD  *
  INSERT   IEAANIP0
  INSERT   IEAAIH00
  INSERT   IEAIOS00
```

Additional INSERT statements required[1]

(IEFUTL object deck)
(IEFU83 object deck)

```
  INCLUDE  SYSLMOD(IEANUC01)
  NAME     IEANUC01(R)
/*
```

[1] These INSERT statements are variable according to your SYSGEN; therefore, refer to your SYSGEN listing for LINK EDIT of SYS1.NUCLEUS and copy the INSERT statements as found there.

Figure 13. JCL for Adding IEFUTL and IEFU83 to SYS1.NUCLEUS After Generating a VS1 System

Note: You must refer to your system generation listing for exact load module names, aliases, and link-edit parameters. Link-edit parameters must be specified according to the characteristics of your exits.

```
//LINKEXIT    JOB     123456,JONES,REGION=300K
//STEP1       EXEC    PGM=IEWL,PARM='link-edit  parameters'
//SYSPRINT    DD      SYSOUT=A
//SYSLMOD     DD      DISP=(OLD,KEEP),DSN=SYS1.LPALIB
//SYSUT1      DD      UNIT=SYSDA,DISP=(,DELETE),
//            SPACE=(TRK,(20,5))
//SYSLIN      DD      *
  ENTRY       IEFUJV
```

(IEFUJV object deck)

```
  NAME        IEFUJV(R)
  ENTRY       IEFBB401
```

(IEFACTRT object deck)

```
  INCLUDE     SYSLMOD(IEFW21SD)
  ALIAS       aliasname1,aliasname2,...
  NAME        IEFW21SD(R)
  ENTRY       IEFSD060
```

(IEFUJI, IEFUSI, and IEFUTL object decks)

```
  INCLUDE     SYSLMOD(IEFSD060)
  ALIAS       aliasname1,aliasname2,...
  NAME        IEFSD060(R)
  ENTRY       IEFUSO
```

(IEFUSO object deck)

```
  NAME        IEFUSO[1]
  ENTRY       IEFU83
```

(IEFU83 object deck)

```
  NAME        IEFU83(R)
  ENTRY       IEFUJP
```

(IEFUJP object deck)

```
  NAME        IEFUJP[1]
/*
```

[1] There are no dummy IEFUSO or IEFUJP exits distributed in VS2.

Figure 14. JCL for Adding Exit Routines to SYS1.LPALIB After Generating a VS2 System

One system generation macro instruction, SCHEDULR, is specifically related to SMF. In either VS1 or VS2, the parameters you supply for the SCHEDULR macro instruction depend on your installation requirements. For example, if you require record type 21 to be written, the ESV parameter of the SCHEDULR macro instruction must specify SMF. You must also supply space for SMF messages written to SYSOUT.

One system generation macro instruction, JES, is related to SMF in VS1. Specifically, the IEFUSO exit (in VS1) is related to the OUTLIM parameter of this macro instruction. Similarily the IEFUSO exit is related to the JES2GEN macro in VS2.

Additional SYSGEN requirements must be fulfilled to add RES or VSAM to your system. For a complete discussion of the system generation procedure, refer to *OS/VS1 System Generation Reference,* GC26-3791, and *OS/VS2 System Generation Reference,* GC26-3792.

In VS1, you may specify in the SCHEDULR macro instruction one of the following:

- SMF=NOTSUPPLIED, which specifies that no SMF processing is to be provided. If the SMF parameter is not coded and the ESV parameter of the SCHEDULR macro does not specify SMF, NOTSUPPLIED is the default.
- SMF=BASIC, which specifies that user written accounting routines, Job Entry Subsystem (JES) accounting information, and exits IEFUSO, IEFUJP, and IEFACTRT are to be provided. No SMF records are generated. For further information on handling accounting information when SMF=BASIC is specified, refer to *OS/VS1 Planning and Use Guide,* GC24-5090.
- SMF=FULL, which specifies that the SMF routines, additional JES accounting information, and exits IEFUIV, IEFUJV, IEFUJI, IEFUSI, IEFUTL, IEFUSO, IEFU83, IEFACTRT, and IEFUJP are to be provided. If the SMF parameter is not coded and the ESV parameter specifies SMF, FULL is the default.

Figure 15 provides more detailed information about the availability of exits in VS1 when BASIC or FULL is coded.

Exit Time	Exit Name	BASIC	FULL
Input Stream Control	IEFUIV	no	yes
Interpreter	IEFUJV	no	yes
Job Initiation	IEFUJI	no	yes
Step Initiation	IEFUSI	no	yes
Timer	IEFUTL	no	yes
OUTLIM	IEFUSO	yes	yes
Record Write	IEFU83	no	yes
Allocation	IEFACTRT	yes	no
Step Termination	IEFACTRT	yes	yes
Job Termination	IEFACTRT	yes	yes
Job Purge	IEFUJP	yes	yes

Figure 15. Availability of Optional Exits Supported by SMF in VS1 When BASIC or FULL Is Coded

The OUTLIM function, which limits the number of logical records written to non-direct SYSOUT data sets, is always supported. The IEFUSO exit, which can be used to override the output limit, is supported in VS2 and if the BASIC or FULL options are selected in VS1.

Defining Use of SMF for Both VS1 and VS2 Systems

The way you intend to use SMF for both VS1 when SMF=FULL is specified and for VS2 when executing either background or foreground, is defined through SMFPRM*xx*. SMFPRM*xx* parameters can be grouped as follows:

- Required parameters, which must always be included and specify the job wait time limit and the system on which SMF is active.
- Optional parameters, which include parameters to select record types, to specify physical information about the data sets, to permit operator modification, and to specify whether exits are to be taken.

SMFPRM*xx* parameters can be specified either (1) before the first IPL of a newly generated system by adding SMFPRM*xx* as a member in SYS1.PARMLIB, (2) at each initialization of SMF by entering SMF parameters at the console during IPL, or (3) in VS1 only, by the setup and subsequent use of the Automated System Initialization facility. However, note that if a job is recovered in a warm start, the setting of some parameters (SID[1], OPT, DSV, REC, and EXT) will be the setting in effect when the job was read in, rather than the setting done during the warm start IPL. See "Entering SMFPRM*xx* into SYS1.PARMLIB" later in this chapter for information on coding and entering SMFPRM*xx* parameters.

The system is distributed with an SMFPRM00 parameter list. These parameters for VS2 are shown in Figure 16. The parameters are the same for VS1 except for SID. (In VS1, SID is equal to 155A.) You should modify this list according to your system requirements. **Note:** In a VS2 system this parameter list applies to both background and foreground jobs.

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

Figure 16. Parameters in the Sample SMFPRM00 Data Set

Required Parameters

Two of the SMFPRM*xx* parameters are required in order to run SMF in a VS2 system or in a VS1 system where SMF=FULL is specified. These two parameters, described in the topics that follow, are:

- JWT, which specifies the job's continous wait time limit.
- SID, which identifies the system on which SMF is active.

JWT Parameter

The JWT parameter is a required parameter that specifies the number of minutes a job is allowed to remain continuously in the wait state. When the specified limit has been reached, the time limit exit (IEFUTL) is entered if exits are to be taken.

[1] The SID parameter is that when the job is read in for record types 4, 5, 34, and 35.

The format of the JWT parameter is:

```
JWT=n
```

where:

n
represents a decimal number containing a maximum of three digits. It specifies the number of minutes that is the continuous wait time limit for jobs in the system. The value specified must be greater than 0.

SID Parameter

The SID parameter is a required parameter that identifies the system on which SMF is active.

The format of the SID parameter is:

```
SID=xxxx
```

where:

xxxx
represents four alphameric characters identifying the system and/or model on which SMF is active.

Optional Parameters

Seven of the SMFPRM*xx* parameters are optional. The optional parameters used to select record types are:

- MAN, which specifies the type of records (all, none, or only user) to be written to the SMF data set.
- OPT, which specifies the type of system, job, and job step information to be collected.
- DSV, which specifies the type of data set information and/or direct access volume information to be collected.
- REC, which specifies whether or not temporary data set information is to be collected.

The optional parameter used to specify physical information about the data sets is:

- BUF, which specifies the size of the SMF buffer.

The other two optional parameters are:

- OPI, which specifies whether or not the operator is presented the parameter list for his inspection and/or modification.
- EXT, which specifies whether or not exits will be taken.

MAN Parameter

The MAN parameter is an optional parameter that specifies the type of records to be written to the SMF data set. This parameter must be specified as MAN=ALL or MAN=USER if records are to be written to the SMF data set. If MAN is equal to ALL or USER, the parameter BUF is required. If records are going to be written only to an installation-defined data set, the parameter may be specified as MAN=NONE. If MAN is equal to NONE, the OPT, DSV, and REC parameters have no function and user exits cannot write to the SMF data set.

The format of the MAN parameter is:

```
[MAN={NONE | USER | ALL}]
```

where:

NONE
specifies that no records are to be written to the SMF data set.

USER
specifies that only user records (from user written exit routines) are to be written to the SMF data set (that is, only record types 128 through 255 are permitted).

ALL
specifies that both SMF and user records are to be written to the SMF data set. If the parameter is omitted, ALL is assumed.

If MAN=NONE is specified, no records are written to the SMF data set, regardless of the values specified in the OPT, DSV and REC parameters. If MAN=ALL is specified, all SMF records are created, unless suppressed by the OPT, DSV, or REC parameters. All of the records created are written unless suppressed by a user written exit routine.

OPT Parameter

The OPT parameter is an optional parameter that specifies the type of system, job, and job step information to be collected by SMF.

The format of the OPT parameter is:

```
[OPT={1 | 2}]
```

where:

1
specifies that only system and job information is to be collected by SMF (that is, record types 4 and 34, which contain job step information, are suppressed) and that the step-related exit, IEFUSI, is not taken.

2
specifies that system, job, and job step information is to be collected by SMF. If the OPT parameter is omitted, 2 is assumed. In VS2, the I/O load balance algorithm uses EXCP counts collected when OPT=2. If OPT=1 , I/O load balancing is non-operative.

Note: If OPT=1 is specified, and if DSV=2 or DSV=3 is also specified, the value OPT=2 is used instead of OPT=1 and message IEE359I is produced.

DSV Parameter

The DSV parameter is an optional parameter that specifies the type of data set information and/or direct access volume information to be collected by SMF.

The format of the DSV parameter is:

```
[DSV={0|1|2|3}]
```

where:

0
specifies that neither data set information nor direct access volume information is to be collected by SMF (that is, record types 14, 15, 17, 18, 19, 62, 64, 68, and 69, which contain data set information and direct access volume information, are suppressed). If the parameter is omitted, 0 is assumed.

1
specifies that direct access volume information (record types 19 and 69) is to be collected by SMF and record types 14, 15, 17, 18, 62, 64, and 68, which contain data set information, are suppressed.

2[1]
specifies that data set information (record types 14, 15, 17, 18, 62, 64, and 68) is to be collected by SMF and record types 19, and 69 which contain direct access volume information, are suppressed.

3[1]
specifies that both data set information and direct access volume information (record types 14, 15, 17, 18, 19, 62, 64, 68, and 69) are to be collected by SMF.

Note: If OPT=1 is specified, and if DSV=2 or DSV=3 is also specified, the value OPT=2 is used instead of OPT=1, and message IEE359I is produced.

REC Parameter

The REC parameter is an optional parameter that specifies whether record type 17 will be written for temporary data sets.[2] This parameter is not functional unless you have specified DSV=2 or DSV=3.

The format of the REC parameter is:

```
[REC={0|2}]
```

where:

0
specifies that record type 17 is to be written for only non-temporary data sets and is to be suppressed for temporary data sets. If the parameter is omitted, 0 is assumed.

2
specifies that record type 17 is to be written for temporary data sets as well as for non-temporary data sets.

BUF Parameter

The BUF parameter is an optional parameter that specifies the size of the SMF buffer. This parameter must be specified if the MAN parameter is not specified or specified as MAN=ALL or MAN=USER. If neither SMF records nor user records are to be recorded, this parameter is not required. Buffer size requirements are

---

[1] Users of the IEHUCAT utility will specify DSV=2 or 3 to collect the utilities input data. These are non-SMF records written on the SMF data set and assigned record types 63 and 67.

[2] The system determines that a data set is *temporary* if it has a system generated name and is created within a job or job step and exists only for the duration of that job or job step. The system generates a name when the DD statement for a new data set does not include the DSNAME parameter, or when it contains a parameter of the form DSNAME=&name or DSNAME=&&name.

discussed in "System and Partition Queue Areas" in the chapter "System Information and Requirements."

The format of the BUF parameter is:

```
[BUF=n]
```

where:

n
represents a decimal number containing three to four digits. The number defines the size (in bytes) of the SMF buffer. Minimum buffer size is 400 bytes; maximum buffer size is 8,192 bytes. If the value is not a multiple of 8, it is rounded to the next *lower* multiple of 8. The buffer size is twice the block size of data written on the SMF data set.

Note: You must dump the SMF data set(s) before you reduce the buffer size from the size specified at the previous IPL; otherwise, the SMF data set cannot be retrieved successfully.

OPI Parameter

The OPI parameter is an optional parameter that specifies whether the SMFPRM*xx* parameters are presented on the console during IPL for the operators inspection and/or modification. The OPI parameter is ignored if it is entered from the console.

The format of the OPI parameter is:

```
[OPI={YES|NO}]
```

where:

YES
specifies that the parameters are presented on the console for the operators inspection and/or modification.

NO
specifies that the parameters are not presented on the console for the operators inspection and/or modification. If the parameter is omitted, NO is assumed.

EXT Parameter

The EXT parameter is an optional parameter that specifies whether the SMF exits, with the exception of IEFUSO, are to be taken. This parameter is independent of the value specified for the MAN parameter. (The IEFUSO exit is taken when the OUTLIM limit is reached for VS2 systems and for VS1 systems generated with the SCHEDULR macro instruction keyword SMF=BASIC or SMF=FULL.)

The format of the EXT parameter is:

```
[EXT={YES|NO}]
```

where:

YES
specifies that exits are to be taken. If the parameter is omitted, YES is assumed.

NO
specifies that exits are not to be taken.

If EXT=YES is specified, the exits actually taken will depend on the data-collection parameter (OPT). If OPT=2 is specified, all exits defined for the system will be taken; if OPT=1 is specified, the job step initiation exit and job step termination exit will not be taken.

Selecting SMF Records Using SMFPRM*xx* Parameters

Figure 17 summarizes the use of the SMPRM*xx* parameters[1] to select SMF records.

Keyword	Value	Meaning	Effect on SMF Records
MAN	ALL	All SMF records.	Record types 0 through 255 may be written to the SMF data set.
	NONE	No SMF records.	The SMF data set is not used.
	USER	Only user-formatted SMF records.	Only record types 128 through 255 may be written to the SMF data set.
OPT	1	System and job information.	Record types 0-3, 5-13, 20, 43R-45R, 47R-49R are created for VS1, but record type 4 is suppressed. Messages IEF375I and IEF376I are provided. Record types 0-3, 5-12, 20, 22, 26, 31, 35, 40, 43H, 45H, 47H-49H are created for VS2, but record types 4 and 34 are suppressed.
	2	System, job, and job step information.	Record type 4 (and in VS2 only 34), in addition to the above, is created, and messages IEF373I through IEF376I are provided.
DSV	0	No information for data sets or direct access volumes.	Record types 14, 15, 17, 18, 19, 62, 64, 68, and 69 are suppressed.
	1	Direct access volume information.	Record types 19 and 69 are created, but record types 14, 15, 17, 18, 62, 64, and 68 are suppressed.
	2	Data set information.[1]	Record types 14, 15, 17, 18, 62, 64, and 68 are created, record types 19 and 69 are suppressed.
	3	Data set and direct access information.[1]	Record types 14, 15, 17, 18, 19, 62, 64, 68, and 69 are created.
REC	0	No information for temporary data sets.	Record type 17 is created for non-temporary data sets only.
	2	Temporary data set information.	Record type 17 is created for temporary data sets as well as for non-temporary data sets.

[1] OPT must equal 2 if DSV is 2 or 3. If OPT equals 1 and DSV equals 2 or 3, the value OPT=2 is substituted.

Figure 17. Summary of the Use of SMFPRM*xx* Parameters to Select SMF Records

Entering SMFPRM*XX* into SYS1.PARMLIB

SMFPRM*xx* consists of a series of parameters contained in 80-character, card-image records. When you have determined the parameters to use, you must have them punched into a card deck and added as a member of SYS1.PARMLIB. The SMFPRM00 member of SYS1.PARMLIB is used during SMF initialization unless an alternate member is specified. This can be specified in VS2 with system parameters, such as SMF=01 for member SMFPRM01 or an alternate member of defaults, such as IEASYSXX. In VS1 an alternate SMF can be specified by replacing the SMF member name in the FAST procedure.

[1] Record type 21 is selected by the ESV SYSGEN parameter. In VS2, MF/1 record types 70-74 are selected by the MF/1 SYSGEN parameter and the START MF1 command. These record types can only be written when MAN=ALL.

Each parameter has the format:

```
keyword=value
```

When punching the parameters into a card deck, consecutive parameters are separated by commas and no embedded blanks are permitted. Although parameters may be included in any order, you must conform to the following restrictions in coding SMFPRM*xx* parameters:

- Parameters cannot be placed in columns 72 through 80; these columns are ignored.
- Continuation of statements must be indicated by placing a blank character after the delimiting comma and before column 72. (All records in the member except the last one indicate continuation.)
- A parameter cannot be divided between two records. Each set of keyword and value must be complete within a single record.

You should add the SMFPRM*xx* card deck as a member of SYS1.PARMLIB by executing the IEBUPDTE utility program. Figure 18 illustrates the JCL required to execute the utility program.

If your parameters change, you may replace the entire SMFPRM*xx* member with a new version by again executing IEBUPDTE. For information on the IEBUPDTE program, refer to *OS/VS Utilities,* GC35-0005.

If operator intervention is allowed (OPI=YES), you may change SMFPRM*xx* parameter values from the operator's console during IPL. If parameter errors occur, the operator is prompted for correct parameters regardless of the value specified for OPI.

```
//ENTER  JOB   123456,SMITH
//       EXEC  PGM=IEBUPDTE,PARM=NEW
//SYSPRINT  DD SYSOUT=A
//SYSUT2  DD DSNAME=SYS1.PARMLIB,DISP=(OLD,KEEP)[1]
//SYSIN   DD DATA
./   ADD  LIST=ALL,NAME=SMFPRM01,LEVEL=01,SOURCE=0
(SMFPRM01 data set)
/*
```

[1] If you want to access SMFPRM00 on the distribution package before generating your system, the data set name is SYS1.APARMLIB.

Figure 18. JCL Required for Entering SMFPRM01 into SYS1.PARMLIB Using IEBUPDTE

Note: If you do not have SMFPRM00 as a member in SYS1.PARMLIB and do not specify an alternate member, the parameters must be entered from the operator's console during IPL.

To effectively plan for and use SMF, you must have knowledge of system requirements (including main and auxiliary storage requirements), of system throughput and performance options, and of operational considerations. These topics are discussed in the remainder of this chapter.

System Requirements

SMF requires additional main storage, and it requires auxiliary storage.

Main Storage Requirements

SMF requires additional main storage space for the VS1 nucleus, the system queue area, and, if exit routines are included, space for system tasks.

Nucleus and System Link Pack Areas

The storage required for the VS1 nucleus is 2,700 bytes plus 148 bytes for each partition. An additional 1,560 byte area is required in the VS1 pageable nucleus and in the VS2 pageable system link pack area for the SMF writer routine.

System and Partition Queue Areas

Main storage is required in both VS1 and VS2 for the Timing Control Table (TCT), System Management Control Area (SMCA), the common parameter area for user written exit routines, and the SMF buffer. Figure 19 shows the areas of each system that require the additional storage.

Item	VS1	VS2
TCT	Fixed PQA	Fixed LSQA
SMCA	Fixed SQA	Fixed SQA
Common Parameter Area	Fixed PQA	Fixed LSQA
SMF Buffer	Pageable SQA	Pageable CSA

Figure 19. System Areas that Require Additional Storage

One TCT is created for each active job step. If only job accounting is required (OPT=1), the size of each TCT is 116 bytes. If job step accounting is required (OPT=2), the size of each TCT is determined by the following formula:

$$TCT = 132 + 12(a) + 8(b)$$

where:

a
is the maximum number of DD statements per job step.

b
is the number of devices allocated because of each DD statement.

Note: In VS2 only the dynamic allocation function can cause the TCT size to be increased. The increase is done in steps of 200 bytes as space is needed.

The SMCA is a permanent table occupying 172 bytes.

The area used for communication between user written exit routines is discussed in "Exit Routine Facilities and Restrictions" in the chapter "Exit Routines."

To determine the SMF buffer size required, you must understand how the buffer is used. SMF records are blocked internally in variable-length format and, if necessary, spanned before they are written to the SMF data set. To allow overlapping of blocking and writing, the buffer size defined by the BUF parameter in SYS1.PARMLIB is divided into two equal parts; while one half of the buffer is being filled, the other half can be written.

If records are to be written to the SMF data set, you must supply a buffer size in the SMFPRM*xx* member before or during IPL. The minimum BUF size is 400 bytes; the maximum is 8,192. The size specified should be twice the size of the largest record to be written to the SMF data set to eliminate the need to span records.[1] More than one record can be written in half the buffer if record sizes permit, but if a record exceeds half the buffer, it is written as segments of a spanned record. To improve performance, the SMF buffer size should be defined to eliminate the need to span records. Note, however, that if the system fails, any records in the buffer are lost. Therefore, do not make the buffer larger than necessary.

Figure 20 shows a list of buffer sizes and the corresponding number of average jobs that can be written on single track of various direct access devices. An average job is described in Figure 22. The figures have been adjusted to allow for interrecord gaps. See "SMF Data Sets" later in this chapter for a sample tabulation of some of the SMF records and their sizes. If you plan to reduce the size of the buffer during consecutive IPLs, dump the SMF data set(s) by using the SMF dump program (IFASMFDP); otherwise, the SMF data set cannot be retrieved successfully.

Buffer Size	Physical Record Length	Jobs per Track	
		2314	3330
400	200	4.5	7.5
600	300	5.0	9.0
800	400	5.5	9.5
1,000	500	5.5	10.0
1,400	700	5.5	10.5
1,800	900	6.0	10.5
2,200	1,100	5.5	11.0
2,600	1,300	6.5	11.5
3,000	1,500	6.0	12.0
4,000	2,000	6.0	12.0
6,000	3,000	6.0	12.0

Figure 20. SMF Buffer Size and Use of Direct Access Space

Auxiliary Storage Requirements

SMF requires auxiliary storage for the SMF data sets and the expansion of system libraries.

[1] The Block Descriptor Word (four bytes) and the Record Descriptor Word (four bytes) should be included in the calculation of the largest record size.

SMF Data Sets

The SMF data set must be permanently resident on a direct access device. The ability of SMF to record data in time sequence order on SYS1.MANX and SYS1.MANY across system failures, IPLs, and the dumping of full data sets is based on the assumption that the data sets are allocated to the same device types and have the same amount of space. Several factors, such as specific system configuration, amount of SMF data to be written, and report program requirements, will determine which type of device is more efficient for a particular installation.

Space must be allocated for the primary SMF data set (SYS1.MANX) and for an alternate SMF data set (SYS1.MANY). The two data sets need not be defined on the same physical device, but it is suggested that they be of the same type. Only device types on which system resident volumes are supported may be used, but, if possible, a device and channel other than those specified for SYSRES should be used.

The SMF data sets must be cataloged and, if you are going to record data, have space allocaed for them prior to IPL. The devices used for the data sets are located via the catalog and become permanently resident at IPL. The devices must be on line and ready during IPL.

Switching between the primary and alternate data sets is automatic as each becomes filled. The SWITCH or HALT command, however, can be used to switch between the data sets whenever you choose. The SMF dump program must be used to transfer a full data set to tape. When a dump is completed, the status of the data set is reset to empty by the dump program.

The space to be allocated to the SYS1.MANX and SYS1.MANY data sets depends on the average amount of data generated by each of your jobs and how often you want to dump the alternating data sets. The method for allocating space for SMF data sets on direct access devices is shown in "Data Management Procedures" later in this chapter. Figure 21 shows the size of some of the records that can appear in an SMF data set.

Figure 22 is an example of how the VS1 space requirements for an entire data set can be established, given certain assumptions. Similar calculations can be made for a VS2 system. The ESV record (type 21), the Data Lost record (type 7), and RES records (types 43R-45R and 47R-49R) are not shown in this example.

Another space consideration is for users of VSAM and the IEHUCAT utility. VSAM record types 63 and 67 are provided for catalog recovery when DSV=2 or 3. These records are very large: 63 is 1000 to 1500 bytes and 67 is 1000 to 1200 bytes in size. These records must be included in space calculations if they are written. If they are being generated but not wanted, see the IEFU83 exit routine.

System Libraries

SMF requires direct access device space for expansion of system libraries when exits are used. In VS1, the system libraries SYS1.LINKLIB and SYS1.NUCLEUS require expansion space. In VS2, the system library SYS1.LPALIB requires expansion space. Refer to *OS/VS1 Storage Estimates,* GC24-5094, and *OS/VS2 Storage Estimates,* GC28-0604, for more information.

Performance

SMF will reduce system throughput by various amounts depending on such factors as:

- SMF options selected, especially buffer size, SMF data set size, and SMF data set device.
- Execution times of user written exit routines.
- System configuration, especially the type and degree of multiprogramming.
- Job stream characteristics, such as the number of jobs, the type of jobs, and any user data set requirements. For example, a job stream consisting of short-running job steps causes more system overhead than a job stream consisting of longer-running job steps.

Category of Data	Event or Status	Use Factor Definition			Record Type No.	Record Size (in Bytes)
		MAN=	OPT=	DSV=		
Day Data	IPL	ALL			0	31
	Partition Definition (VS1)	ALL			13	16 + 22 per partition
	Devices Online at IPL	ALL			8	16 + 4 per device
	CPUs, Storage Boxes, Channels Online at IPL (VS2 only)	ALL			22	18 + 6 per element
	End of day (VS1 only)	ALL			12	34
Machine Data	Accumulated Wait Time (VS1 only)	ALL			1	34
	SMF Records Lost	ALL			7	24
	Devices Varied Online	ALL			9	16 + 4 per device
	Device Varied Offline	ALL			11	20
	Device Recovered by Allocation	ALL			10	44
	CPUs, Storage Boxes, Channels Varied online or offline (VS2 only)	ALL			22	18 + 6 per element
Auxiliary Storage Data	Space Available on DASD Volumes at IPL, HALT EOD, and when Demounted	ALL		1,3	19	64
	VSAM Data Space Defined, Extended, or Deleted	ALL		1,3	69	102
Processing Data	Initiation of a Job	ALL			20	61 + 1 per accounting data item + 1 per accounting data character
	Step Processing	ALL	2		4	117 (VS1) 137(VS2) + 8 per DD statement + 1 per accounting data item + 1 per accounting data character
	Job Processing	ALL			5	117 + 1 per accounting data item + 1 per accounting data character
	SYSOUT Processing	ALL			6	65 (VS1) 90 (VS2)
	Job Purge (VS2 only)	ALL			26	232

Figure 21. SMF Record Sizes (Part 1 of 2)

Category of Data	Event or Status		Use Factor Definition			Record Type No.	Record Size (in Bytes)
			MAN=	OPT=	DSV=		
Non-VSAM Data Set Activity Data	Closing, or EOV Processing of a Data Set	Data Set Opened for INPUT, or RDBACK.	ALL	2	2,3	14	264 + 24 per UCB + 28 for ISAM
		Data Set Opened for OUTPUT, UPDATE, INOUT, OUTIN.	ALL	2	2,3	15	264 + 24 per UCB + 28 for ISAM
	Scratching of a Data Set		ALL	2	2,3	17	88 + 8 per data set scratched
	Renaming of a Data Set		ALL	2	2,3	18	132 + 8 per data set renamed
VSAM Data Set Activity Data	VSAM Cluster or Component		ALL	2	2,3	62	138 +10 per volume
	VSAM Cluster or Component Closed or EOV		ALL	2	2,3	64	228 +26 per extent
	VSAM Catalog Entry Renamed		ALL	2	2,3	68	170
VS Data Management	Record Descriptor Word		ALL			ALL	4
	Block Descriptor Word		ALL				4

Figure 21. SMF Record Sizes (Part 2 of 2)

Event or Status	Record Type	Assumption for this Example	Example No. of Bytes per Record	Example No. of Records	Example Total	
					Job-Related Records	System-Related Records
IPL	0	Once per day	31	1		31
Partition Definition	13	Once per day (4 partitions)	104	1		104
Devices Online at IPL	8	20 devices, including 6 DASD	96	1		96
	19		64	6		384
End of day	12	Once per day	34	1		34
	19	6 DASD	64	6		384
Accumulated Wait Time	1	Once every 10 min. for 4 hours	34	24		816
Device Varied Online	9	Twice per hour for 4 hours	20	8		160
Device Varied Offline	11	Twice per hour for 4 hours	20	8		160
Device Recovered at Allocation	10	Once per hour for 4 hours	44	4		176
Scratch a Non-temporary Data Set	17	Once per 4 hours, 1 volume per data set	96	1		96
Rename a Data Set	18	Once per 4 hours, 1 volume per data set	140	1		140
Rename a VSAM catalog entry[3]	68	Once per day	1			170
		Total for these records		62		2,871
Job Processing	5	Accounting Data[1]	129	1	129	
	20		73	1	73	
	19	Demount 2 DASD volumes	64	2	128	
Step Processing	4	4 DD statements per step[2], Accounting Data[1], 3 steps per job	161	3	483	
	14	1 EOV processing and close 2 data sets per step, 3 steps per job	288	9	2,592	
	15	1 EOV processing and close 2 data sets per step, 3 steps per job	288	9	2,592	
	62	Open 2 components in 1 step of each job	168	2	336	
	64	1 EOV and 1 close per VSAM component	306	6	1,836	
SYSOUT Processing	6	2 output writers per step, 3 steps per job	65	6	390	
		Total for one job		39	8,559	
		Total for 12 jobs per hour for 4 hours (48 Jobs)		1,872	410,832	
		Total SMF Data				413,703
Record Descriptor Word			4	1,934		7,736
Block Descriptor Word		6 Records per block	4	323		1,292
		Total number of bytes for this example				422,731

[1] In this example, accounting data consists of two 5-byte items.
[2] None of the four DD statements refers to DD DATA, DD *, or SYSOUT data sets.
[3] If you have VSAM data sets and DSV of 2 or 3, VSAM record types 63 and 67 are written. These records require a large amount of space of the SMF data set. Refer to *Virtual Storage Access Method (VSAM) System Information Guide,* GC26-3835.

Figure 22. Example of Data Set Space Requirements For VS1

Operational Considerations

The system operator is responsible for operational procedures, such as the requirements for IPL and for dumping the SMF data set, and for any special procedures required by user written report and analysis routines.

IPL Procedures

SMF initialization is the final step of the system IPL procedure. The SMF initialization program checks for the existence and validity of the SMFPRM00 member, or SMFPRM*xx* member if requested. If the requested SMF has not been entered into SYS1.PARMLIB, the initialization program writes a message to the console, allowing you to enter SMF parameters from the console. If the verification program encounters an input/output error while reading SMFPRM*xx*, it writes a message to the console, allowing you to either repeat the IPL procedure or enter SMF parameters from the console.

If the initialization program finds SMFPRM*xx* and reads it without error, each parameter is checked for accuracy. If any parameters are incorrectly specified or if required parameters are omitted, messages are issued, allowing the operator to enter parameters from the console. If all parameters are correctly specified and operator intervention is allowed (OPI=YES was specified in SMFPRM*xx*), messages are issued, allowing the operator to examine and modify the parameters.

When the initialization program has completed validation of SMFPRM*xx* and if SMF recording is requested, it checks for the existence of data sets SYS1.MANX and SYS1.MANY and for the availability of the devices on which they are defined. The initialization routine checks for the availability of the specified device and the allocation of the data set on the specified direct access device.

If an error is found, a message is issued, which indicates that no recording of SMF records is allowed until the condition is corrected and the IPL procedure is repeated.

If no errors are found, the initialization routine determines which data set (SYS1.MANX or SYS1.MANY) should receive SMF records. If neither data set contains data, SYS1.MANX receives the SMF records. If one data set contains data and the other one is empty, the partially full data set receives the SMF records. If both data sets already contain data, the data set with the lesser amount of data receives the SMF records.

SMF data is maintained in chronological sequence if both SYS1.MANX and SYS1.MANY are defined on the same type of device with the same space allocation and if the data sets are dumped when called for by the control program. These conditions are not checked. However, it is a user error that occurs if data is recorded out of order when these conditions are not met.

When data set verification has been completed without error, the initialization routine writes the IPL and configuration records (SMF record types 0, 8, and in VS2 only, type 22, to the SMF data set, and gives control to the system so that processing of input streams may begin.

Data Management Procedures

The SYS1.MANX and SYS1.MANY data sets must be cataloged and, if you are going to record on them, space for them must be allocated on direct access devices before IPL. If possible, a device and channel other than those specified for SYSRES should be used. Figure 23 illustrates sample DD statements for allocating space and cataloging the SMF data set.

Note: Specification of a secondary space allocation is ignored.

```
//MANX   DD    DSNAME=SYS1.MANX,UNIT=190,VOLUME=SER=111111,
//             SPACE=(TRK,(20)),DISP=(NEW,CATLG)
//MANY   DD    DSNAME=SYS1.MANY,UNIT=191,VOLUME=SER=222222,
//             SPACE=(TRK,(20)),DISP=(NEW,CATLG)
```

Figure 23. DD Statements for Allocating Space and Cataloging SMF Data Sets on Direct Access Devices

SMF Dump Program

The SMF dump program must be used to transfer full SMF data sets to another data set (usually on tape) for analysis and to reset the dumped data set for possible reuse as the active SMF data set. Figure 24 shows the JCL required to execute the SMF dump program. The output is a non-temporary data set on a standard labeled tape. Have the operator record the volume serial number as it is needed to reference this data set from another job. The SMF dump program uses the Basic Sequential Access Method to perform a physical copy from the input data set, DUMPIN, to the output data set, DUMPOUT. In the copy process, the program creates two SMF records and writes them to the output data set: a Dump Header record (record type 2) at the beginning of the data set and a Dump Trailer record (record type 3) at the end of the data set. The SMF dump program also writes messages, as required, to SYSOUT.

```
//DUMPX      JOB   201,,MSGLEVEL=1,PRTY=12
//STEP1      EXEC  PGM=IFASMFDP
//DUMPIN     DD    DSNAME=SYS1.MANX,DISP=OLD[1]
//DUMPOUT    DD    DSNAME=SMFDATA,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSPRINT   DD    SYSOUT=A
```

[1] If the alternate SMF data set, SYS1.MANY, is being dumped to tape, SYS1.MANY appears in the DSNAME parameter.

Figure 24. JCL for Executing the SMF Dump Program

When either the SYS1.MANX or SYS1.MANY becomes full, a message is written to the console requesting a dump. In no case should a data set that is being filled be dumped. If the invoked SMF dump program attempts to dump the active SMF data set, a message is printed at the console notifying the operator of his error; in response, the operator must cancel the dump program. He may then invoke the dump program to dump the correct SMF data set.

The SWITCH command or HALT command can be used to switch between the SMF data sets so the previously active data set can be dumped. The format of the SWITCH command is:

```
{SWITCH | I} SMF
```

The format of the HALT command is:

```
{HALT | Z} EOD
```

When either the SWITCH or HALT command is entered, the following actions occur:

1. A record type 19 is written for each online direct access device if DSV=1 or DSV=3 was specified.
2. A record type 12 is written, in VS1 only.
3. The SMF buffer is written to the SMF data set.

4. The SMF data sets are switched or interchanged (that is, a different data set is designated to receive SMF records). If empty, the previously inactive data set is designated to receive SMF records. The previously active data set is unavailable for recording.

In VS1 only, both the SWITCH or HALT commands cause the CPU wait time and system statistics, accumulated up to the time of the command, to be recorded in record type 12. Thus, record type 12 contains the wait time and system statistics accumulated from the expiration of the ten-minute time interval reflected in the last record type 1 to the time of the SWITCH command or HALT command. The next record type 1 contains the wait time and system statistics accumulated from the SWITCH command or HALT command to the expiration of the next ten-minute time interval.

The HALT command also causes other system actions, which are described in *Operator's Library: OS/VS1 Reference,* GC38-0110, and *Operator's Library: OS/VS2 Reference,* GC38-0210. The SWITCH command causes only the actions described here.

You may enter jobs specifying execution of the dump program into the system and hold them on the job queue until a dump is required. You may then release the appropriate job to dump the specified data set. Another method for executing the dump is to start a reader to an input stream containing the JCL for the dump program. A high priority should be assigned to the dump job to ensure immediate initiation.

Producing a report usually requires at least two operations: sorting the SMF records and writing them in an appropriate format. Your installation's requirements will determine what further analysis of SMF data is necessary.

Sorting SMF Records

You may use the operating system sort/merge program product, number 5734-SM1, to sort SMF records. Note, however, that sort/merge will not process user records that are less than 18 bytes long. Your report format and analysis requirements will determine the fields on which to sort and will determine the sorting sequence. Sort exit E15 allows you to extract or delete selected records as the SMF dump data set is sorted, and sort exit E35 allows you to insert records into the final sorted output data set.

Sample Sort/Merge Exit Routines

Two sample sort/merge exit routines are provided in SYS1.ASAMPLIB: an E15 exit routine (called SMFE15) and an E35 exit routine (called SMFE35). The SMFE15 routine is designed for use with SMFE35. These samples assume that the input consists only of record types 0-13.

The SMFE15 routine extracts all non-job oriented SMF records (that is, records without a job log number) from the SMF dump data set. Dump header and dump trailer records (SMF record types 2 and 3) are retained in a temporary data set (DD name HDRDATA), and all other system-oriented records are retained on another temporary data set (DD name SORDATA). Only job-oriented records (that is, records having a job log number) are sorted.

The SMFE35 routine places in the sort output data set all records extracted by the SMFE15 exit routine. The system-oriented records precede the sorted job-oriented records in the output data set and are inserted in the following order: dump header records, dump trailer records, and all other system records.

An example of the JCL required to execute the sort/merge program is provided in the SYS1.ASAMPLIB member named SMFSORT. Figure 25 shows JCL that can be used to obtain a listing of sample sort exit routines and of sample sort JCL from SYS1.ASAMPLIB.

If you plan to include the sample exit routines in your sort application, you must assemble and link-edit them before executing the sort/merge program. Figure 26 shows JCL that can be used for this procedure, including one possible sort application. In this example, SMF records are to be sorted first on the job log number (major control field), then on the date and time portions of the time stamp (minor control fields). Displacements of these fields (from the beginning of the physical record) are 19, 11, and 7 bytes, respectively.

For a description of the cataloged procedure ASMFCL (assemble and link-edit), see *OS/VS Assembler Programmer's Guide*, GC33-4021. For a detailed description of the sort/merge program, see *Sort/Merge, SM/1, Programmer's Guide*, SC33-4007.

Designing a Report Program

The basic operations of a report program are formatting and printing data from SMF records. The input to a report program is normally the sorted SMF data set. Your installation's report requirements will determine the amount of data modification, analysis, and formatting your report program must perform.

A sample source report program (SMFFRMT), written in PL/I-F, is provided in SYS1.ASAMPLIB. Figure 27 illustrates sample output from the SMFFRMT program. This program can also be used to print selected types of SMF records. To do so, specify the record types, separated by commas, in the PARM field of the EXEC statement. If you want to print all record types, the PARM parameter is not required.

Before using SMFFRMT, you must compile the program using the PL/I compiler. Figure 28 shows JCL required to use SMFFRMT after it has been compiled and linkedited into SYS1.LINKLIB.

```
//PRINT  JOB    123456,SMITH
//       EXEC   PGM=IEBPTPCH
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   DSNAME=SYS1.ASAMPLIB,DISP=(OLD,KEEP),
//              UNIT=XXXX,VOLUME=SER=XXXXXX[1]
//SYSUT2   DD   SYSOUT=A
//SYSIN    DD   *
     PRINT    TYPORG=PO,MAXNAME=4,MAXFLDS=4
     MEMBER   NAME=SMFSORT
     RECORD   FIELD=(80)
     MEMBER   NAME=SMFE15
     RECORD   FIELD=(80)
     MEMBER   NAME=SMFE35
     RECORD   FIELD=(80)
/*
```

[1] The volume and unit parameters depend on your installation's request; check with your system programmer.

Figure 25. JCL for Obtaining a Listing of Sample Sort Exit Routines

```
//SMFSORT  JOB  MSGLEVEL=1
//STEP1  EXEC  ASMFCL[1]
//ASM.SYSIN DD *
         E15  SOURCE  DECK
/*
//LKED.SYSLMOD DD DSNAME=SMF1.EXITS,UNIT=2314,[2]
//        DISP=(NEW,KEEP),SPACE=(TRK,(10,5,1)),
//        VOLUME=SER=231400
//LKED.SYSIN DD *
  NAME   E15(R)[3]
/*
//STEP2  EXEC  ASMFCL[1]
//ASM.SYSIN DD *
         E35  SOURCE  DECK
/*
//LKED.SYSLMOD DD DSNAME=SMF1.EXITS,DISP=(OLD,KEEP),[2]
//        UNIT=2314,VOL=SER=231400
//LKED.SYSIN DD *
  NAME   E35(R)[3]
/*
//SORTSTEP EXEC PGM=SORT,REGION=100K[4]
//SYSOUT DD SYSOUT=A
//SORTLIB  DD  DSNAME=SYS1.SORTLIB,DISP=SHR
//EXITLIB  DD  DSNAME=SMF1.EXITS,DISP=(OLD,KEEP),[5]
//        UNIT=2314,VOL=SER=231400
//SORTIN   DD  UNIT=2400,VOLUME=SER=SYSMAN,DISP=OLD,[6]
//        LABEL=(,NL),DCB=(RECFM=VBS,LRECL=600,BLKSIZE=200)[7]
//SORTWK01 DD UNIT=2314,SPACE=(TRK,(50),,CONTIG)[8]
//SORTWK02 DD UNIT=2314,SPACE=(TRK,(50),,CONTIG)[8]
//SORTWK03 DD UNIT=2314,SPACE=(TRK,(50),,CONTIG)[8]
//SORTOUT  DD UNIT=2400,DSNAME=SMF1.SORTOUT,LABEL=(,NL),[9]
//        DISP=(,KEEP),DCB=(RECFM=VBS,LRECL=600,BLKSIZE=200)[7]
//SORDATA  DD  UNIT=SYSDA,SPACE=(CYL,(1,1)),[10]
//        DCB=(RECFM=VBS,LRECL=600,BLKSIZE=200)[7]
//HDRDATA  DD  UNIT=SYSDA,SPACE=(TRK,(5,5)),[10]
//        DCB=(RECFM=VBS,LRECL=600,BLKSIZE=200)[7]
//SYSIN  DD    *
 SORT  FIELDS=(19,16,A,11,4,A,7,4,A),FORMAT=BI,SIZE=E4000[11]
 MODS  E15=(E15,700,EXITLIB,N),E35=(E35,1500,EXITLIB,N)[11]
 END
/*
```

1 EXEC statement for cataloged procedure ASMFCL (assemble and link-edit).
2 The sample sort exit routines will be link-edited into data set SMF1.EXITS.
3 Linkage editor control statements specifying that E15 and E35 will be the load module names of the exit routines.
4 EXEC statement for the sort/merge program.
5 Data set SMF1.EXITS is specified as the library in which sort exit routines may be found.
6 Input to the sort program is the SMF dump data set, contained on a tape having a volume serial number of SYSMAN.
7 The LRECL value may be larger than the BLKSIZE value because records may be spanned. The LRECL value must be as large as the longest SMF record being created plus four bytes for the RDW. The BLKSIZE must be equal to one half the SMF parameter BUF. You are expected to modify these parameters according to your buffer size and the longest record collected.
8 Three sort work units are defined as being direct access devices.
9 The sort output data set is to be written on tape.
10 Two data sets required by the sample sort exit routines are defined on direct access devices.
11 The sort/merge control statements define the sort control fields and exit routines to be used in this sort application.

Figure 26. Sample Sort Procedure

```
RECORD            HEADER/RECORD                                                                    DATE        720117
 TYPE                                                                                              PAGE             1
 02      0102  00681590  0072017F  C2C2F4F5                                    *.. .... .... BB45                *

 08      0108  0066D2E5  0072017F  C2C2F4F5                                    *.. ..KV .... BB45                *
         002E  08230009080200OD0808000E0808000F2008023020080231200802          *.. ..............................*
         32200802332008023420080235200802 36                                   *.................                 *

 01      0101  0066D2E8  0072017F  C2C2F4F5                                    *.. ..KY .... BB45                *
         0000602C  0066D2E8  00000000  00000000  00000000                       *.... ..KY .... .... ....         *

 13      010D  0066D32F  0072017F  C2C2F4F5                                    *.. ..L. .... BB45                *
         002E  00004000000140404040404040404040404040C1  01020000             *.. .. ...                  A ....*
         00034040404040404040404040404040C1D1D4                                *..              AJM              *

 10      010A  00672333  0072017F  C2C2F4F5                                    *.. .... .... BB45                *
         4040404040404040  00000000  00000000  4040404040404040  0006        *          .... ....            ..*
           0801000C                                                            * ....                            *

 05      0105  00677297  0072017F  C2C2F4F5                                    *.. .... .... BB45                *
         C1C1D1E2F5F6C1F0  00672FFE  0072017F  0000000000000000  01           *AAJS56A0 .... .... ........ . *
         006769F5  0072017F  00000000  0000  06  00673036  0072017F           *...5 .... .... .. . .... .... *
         00  8000000000  00  0801  C1  00  00000000000000000000000000         *. ..... . .. A . ..............*
         000000000000  2E  D7C5D5C3C540D1E64040404040404040404040404040      *..... . PENCE JW                 *
         000173  01  15  F7F6F0F1F0F1F3F1F6F2F0F2F7D5F0F2F2D7D9D6C4           *... . . 7601013162027N022PROD *

 06      0106  006774D1  0072017F  C2C2F4F5                                    *.. ...J .... BB45                *
         C1C1D1F2F5F6C1F0  00672FFE  0072017F  0000000000000000  C1           *AAJS56A0 .... .... ........ A *
         00677408  0072017F  00000008  00  01  40404040                       *.... .... .... . .               *

 05      0105  0067AE1F  0072017F  C2C2F4F5                                    *.. .... .... BB45                *
         C1C1D1E2F5F6C2F0  00677F2F  0072017F  0000000000000000  01           *AAJS56B0 .... .... ........ . *
         00678298  0072017F  00000000  0000  06  00677F68  0072017F           *.... .... .... .. . .... .... *
         00  8000000000  00  0801  C1  00  00000000000000000000000000         *. ..... . .. A . ..............*
         000000000000  2E  D7C5D5C3C540D1E64040404040404040404040404040      *..... . PENCE JW                 *
         00031C  01  15  F7F6F0F1F0F1F3F1F6F2F0F2F7D5F0F2F2D7D9D6C4           *... . . 7601013162027N022PROD *

 06      0106  0067B073  0072017F  C2C2F4F5                                    *.. .... .... BB45                *
         C1C1D1E2F5F6C2F0  00677F2F  0072017F  0000000000000000  C1           *AAJS56B0 .... .... ........ A *
         0067AF96  0072017F  00000008  00  01  40404040                       *.... .... .... . .               *

 12      010C  0067B81A  0072017F  C2C2F4F5                                    *.. .... .... BB45                *
         00009B2F  0067B7FF  0000051A  000005BF  00000002                      *.... .... .... .... ....         *

 03      0103  0068161A  0072017F  C2C2F4F5                                    *.. .... .... BB45                *
```

Figure 27. Sample Output from SMFFRMT Program

```
//FORMAT    JOB  123456,SMITH
//FRMT      EXEC PGM=SMFFRMT,PARM='1,2,3,5,6,8,9,10,11,12,13'
//SYSPRINT  DD SYSOUT=A
//REPORT    DD SYSOUT=A,DCB=(RECFM=VBA,BLKSIZE=3500)
//SMFDATA   DD DISP=(OLD,KEEP),LABEL=(,NL),VOL=SER=XXXXXX,[1]
//             UNIT=2400,DCB=(RECFM=VBS,BLKSIZE=1000)[2]
```

[1] The tape volume serial number replaces the X's.

[2] The BLKSIZE is one-half the value specified as the SMF IPL parameter BUF. For this example, BUF is equal to 2,000.

Figure 28. Sample JCL to Run SMFFRMT

This chapter contains information for planning and writing exit routines. It describes:

- Exit routine facilities and restrictions, including information on communication among user written exit routines; parameters passed to exit routines; the SMFWTM macro instruction, which can be used by exit routines to write a record to the SMF data set; and the IFASMFR macro instruction, which can be used to symbolically address fields in SMF records.
- Exit routines, including a full description of each of the sample SMF exit routines.

Figure 29 shows the exits available in each system.

Exit	VS1	VS2
IEFUIV	Yes	No
IEFUJV	Yes	Yes
IEFUJI	Yes	Yes
IEFUSI	Yes	Yes
IEFUTL	Yes	Yes
IEFUSO	Yes	Yes
IEFU83	Yes	Yes
IEFACTRT	Yes	Yes
IEFUJP	Yes	Yes

Figure 29. SMF Exits Available in Each System

Exit Routine Facilities and Restrictions

The functions performed by your exit routines are determined solely by the requirements of your installation. The following paragraphs describe exit routine restrictions, formats of parameters passed to each exit, and the required return codes. Sample Assembler language exit routines for some exits are provided in a member (SMFEXITS) of SYS1.ASAMPLIB. Figure 30 shows the JCL required to retrieve a listing of these sample routines.

```
//PRINT     JOB  123456,SMITH
//          EXEC PGM=IEBPTPCH
//SYSPRINT  DD   SYSOUT=A
//SYSUT1    DD   DSNAME=SYS1.ASAMPLIB,DISP=(OLD,KEEP),
//               UNIT=XXXX,VOLUME=SER=XXXXXX[1]
//SYSUT2    DD   SYSOUT=A
//SYSIN     DD   *
      PRINT      TYPORG=PO,MAXNAME=1,MAXFLDS=1
      MEMBER     NAME=SMFEXITS
      RECORD     FIELD=(80)
/*
```

[1] The volume and unit parameters depend on your installation's request; check with your system programmer.

Figure 30. JCL for Obtaining a Listing of Sample Exit Routines

User written exit routines in VS2 *must* be written in re-enterable code. Of the user written exit routines in VS1, IEFU83 and IEFUTL, and any other routine made resident, must be written in re-enterable code. It is recommended that all exits be re-entrant. All user written exit routines receive control via a BALR. User written exit routines must save registers when they receive control and restore registers before returning to the control program. Register 13 contains the address of the register save area, register 14 contains the return address, and register 15 contains the entry point address. If an exit issues a WTOR in VS2, it should specify the LONG WAIT option.

Under VS2, all SMF exit routines except IEFU83 and IEFUSO can access installation-defined data sets. Under VS1, all exit routines except IEFUSO and IEFUTL can access installation-defined data sets.

The exits are supported in VS1 for problem programs started from the console but are not supported in VS2.

User written exit routines can communicate with each other via the user-communication field and the user-identification field. The user-communication field can be used to communicate between exits taken for different jobs. In VS1, the user-communication field returned from IEFUIV is placed in the job management record for the next entry to IEFUIV, hence enabling communication between jobs at this exit. This process is similar in VS2, except the exit involved is IEFUJV and only for entry codes 0-16. The user-identification field can be used to communicate between exits of the same job. These fields are passed to every exit routine except IEFU83 and are logically part of each job. The user-communication field is initialized to binary zeros at reader initialization (starting a reader or restoring a transient reader). The user-identification field is initialized to blanks as each new job is read.

Any exit routine may obtain an additional work area by issuing a GETMAIN macro instruction that specifies an appropriate subpool in system queue area. Figure 31 shows the characteristics of the subpools that can be specified. The address of the work area can be placed in the user-communication field. (You must consider the storage required by this work area when estimating the size of system queue area.)

The VS2 user must be aware that different exits receive control in different keys. IEFUSO and IEFUJP receive control in storage key 1 and IEFUJV is entered in key 1 for entry codes 0-16 and in key 0 for entry code 32. All other exits have a storage key 0.

Note: Communication areas obtained by exits are not maintained if the system is restarted.

Subpool Number	Storage Is Allocated	Storage Attributes
231 (VS2 only)	In common service area.	Explicitly freed, pageable, fetch protected, user's key.
241 (VS2 only)	In common service area.	Explicitly freed, pageable, not fetch protected, user's key.
245	In system queue area.	Explicitly freed, by issuing a FREEMAIN macro instruction.
253	In partition queue area.	Automatically freed at end of task.
254	In partition queue area.	Automatically freed at end of step.
255	In partition queue area.	Explicitly freed, by issuing a FREEMAIN macro instruction.

Note: Only routines having a protection key of 0 can obtain the storage in system queue area (VS1 or VS2) or high in a partition (VS1). In VS2, exits receive control in different keys, so use subpool number 241 for read access from all exits. Subpool number 231 is to be used for exit communication only among exits of the same key.

Figure 31. Characteristics of Subpools in System Queue Space

If an exit routine cancels a job during or after job initiation, a job termination record (record type 5) is written to the SMF data set if the writing of records is permitted. If you require job cancellation information in the System Output Message Data Set, you may pass a message to module IEFYS from the termination exit routine IEFACTRT. IEFACTRT is the only SMF exit that can write to the System Output Message Data Set, and only by passing a message to IEFYS. At job termination time, you can determine if an IEFUJV (VS1 only), IEFUJI, IEFUSI, or IEFACTRT exit routine canceled a job, by examining the job termination indicators in record type 5. At step termination time, you can determine if an IEFUSO or IEFUTL exit routine canceled a job, by examining the step completion code field in record type 4. The VS1 system does no recording of jobs canceled before job initiation. In VS1, the only exit before job initiation is IEFUIV. The installation may, however, write a record to the SMF data set from this exit when a job is not accepted for processing prior to job initiation. For jobs cancelled before job initialization in VS2, only record types 6 and 26 are generated.

Your routines can use the SMFWTM macro instruction to write to the SMF data set. If you want to use your own data sets, you must define them for VS1, as follows:

- A data set used by exit routine IEFUIV requires a DD statement in the reader cataloged procedure.
- A data set used by exit routines IEFUJV, IEFUJI, IEFUSI, IEFU83, and IEFACTRT requires a DD statement in the initiator cataloged procedure.
- A data set used by exit routine IEFUJP requires a DD statement in the writer cataloged procedure.

You cannot write to your own data sets from the exit routines IEFUSO and IEFUTL.

If you want to use your own data sets, you must define them for VS2, as follows:

- A data set used by exit routines IEFUJV and IEFUJP requires a DD statement in the JES2 cataloged procedure.
- A data set used by exit routines IEFUJI, IEFUSI, IEFUTL, and IEFACTRT requires a DD statement in the initiator cataloged procedure.

You cannot write to your own data sets from the IEFU83 and IEFUSO exits.

Note: In either VS1 or VS2, user data sets cannot be allocated to SYSOUT.

Exit Routine Parameters

When an exit routine receives control, register 1 points to a list of four-byte addresses. The first entry in the list is common to all exit routines except IEFU83. The first entry points to a parameter area that is 72 bytes long for VS1 when SMF=FULL is specified or 36 bytes long for VS2.

Figure 32 describes the format of the parameter area for VS1. Note that the fields in the parameter area are filled in chronologically; therefore, not all fields are meaningful for all exits. The first 36 bytes shown in Figure 32 describe the format of the parameter area for VS2.

Figure 33 summarizes the information available to each exit (in addition to common exit parameters, described in Figure 32), when each exit is called, and the return from each exit to the control program. The names in parentheses are mandatory entry-point names that must be assigned.

Displacement from Pointer	Field Size	Data Format	Description
0	8	EBCDIC	Job name
8	4	binary	Time, in hundredths of a second, that the reader recognized the JOB card for this job
12	4	packed decimal	Date that the reader recognized the JOB card for the job, in the form 00YYDDDF, where F is the sign
16	4	EBCDIC	System identification from SMFPRMXX SID parameter
20	8	EBCDIC	User identification field. SMF places this data in all subsequent records for this job. This field is initialized to EBCDIC blanks for each job.[1]
28	1	binary	Number of the step being processed
29	1	binary	Eight indicators of SMF options that are selected by the user. A bit setting of 1 indicates the related option was selected. *Bit* *Option* 0 System and job accounting 1 Step accounting 2 Dynamic exits 3 Data set accounting 4 Volume accounting 5 Reserved 6 Temporary data set scratch records 7 0—Background job 1—Foreground job[2]
30	1	binary	Reserved (VS1) *Bit* *Meaning When Set* (VS2) 0 Step restart 1 Checkpoint/restart 2 Continue restart 3 Reserved 4 Warm start 5-7 Reserved
31	1	binary	Reserved
32	4	binary	User communication field. This field is intended to be a user exit routine communication field. This field is initialized to zeros *only* when the reader is started or a transient reader is restored in VS1 and when the converter is started in VS2.[1]
The following fields apply to only VS1:			
36	4	binary	Real time in reader, in hundredths of a second
40	4	binary	Number of lines read
44	1	binary	Job priority
45	1	binary	Reserved
46	1	EBCDIC	Job class
47	1	EBCDIC	Reserved
48	4	binary	Real time to print, in hundredths of a second[3]
52	4	binary	Number of SYSOUT lines printed[4]
56	4	binary	Real time to punch, in hundredths of a second
60	4	binary	Number of SYSOUT lines punched
64	4	binary	Real time for tape, in hundredths of a second
68	4	binary	Number of SYSOUT lines written to tape

[1] These two fields are the only fields provided for user modification.
[2] Because TSO does not operate under VS1, bit 7 is always 0 when VS1 is used.
[3] The SYSOUT fields are valid only at the IEFUJP exit.
[4] This includes job related JOBLOG lines if JESPARMS parameter is JOBLOG=YES.

Figure 32. Common Exit Parameter Area

Exit Routine	Parameters Passed	When Entered	Type of Return
Input Stream Validation—VS1 only (IEFUIV)	JCL image of JOB statement.	When a JOB card is encountered by input-stream control.	Continue or cancel.
Job Validation (IEFUJV)	JCL image, JCL statement type.	Each JCL card (excluding comment statements), after all JCL has been interpreted.[1]	Continue or cancel.
Job Initiation (IEFUJI)	Programmer name, priority, account field.	Job initiation.	Continue or cancel.
Step Initiation (IEFUSI)	Step program name, step name, accounting fields.	Step initiation.	Continue or cancel.
SYSOUT Limit—(IEFUSO)	DCB (VS1 only).	OUTLIM limit exceeded.	Continue with new limit or cancel.
Time Limit (IEFUTL)	Entry type.	Job CPU time limit exceeded, step CPU time limit exceeded, continuous wait time limit for the job exceeded.	Continue with new time limit or cancel.
SMF Record—(IEFU83)	SMF record to be written.	When a record is to be written to the SMF.	Write or don't write SMF record to SMF.
Termination (IEFACTRT)	Programmer name, job CPU time, job accounting fields, step CPU time, step accounting fields, completion code, SMF record.	Step termination, job termination.	Continue or cancel; write or skip SMF record.
Job Purge—VS1 only (IEFUJP)	Programmer name, job CPU time, number of accounting fields in the JOB statement, address of the accounting information.	When a job is ready to be purged from the system.	None.
Job Purge—VS2 only (IEFUJP)	SMF record to be written.	When a job is ready to be purged from the system.	Write or skip SMF record.

[1] In VS2, IEFUJV is also entered after all JCL has been processed by the converter.

Figure 33. Exit Routine Characteristics

SMFWTM Macro Instruction

You may use the SMFWTM macro instruction in any routine that has a protection key of 0 except IEFU83, or in VS2 an APF authorization, to write a record to the SMF data set. The macro is supplied on SYS1.MACLIB.

Note: Record types 128 through 255 are available for user written records.

The format of the SMFWTM macro instruction is:

```
[label] SMFWTM {record address | (r)}
```

where:

record address
is the symbolic address of the record to be written.

(r)
is a register containing the address of the record. You may use either the absolute register number or a symbolic designation. In either case you must enclose the value in parentheses; for example, (2) or (REG2).

The record to be written should include a standard SMF record header and a record descriptor word (RDW). See the first fourteen bytes of record type 6 in the chapter "Accounting Records" for the header format. For a discussion of the RDW, refer to *OS/VS Data Management Services Guide,* GC26-3783.

For record types 0-127, the issuer of the SMFWTM macro specifies only the record type field in the 14-byte header and the remainder is supplied by the macro. An exception to this is record types 4, 5, 34, and 35 which are passed to exits before the macro is issued. For user records, types 128-255, the issuer supplies the entire header.

Note: The sort/merge program product will not process records less than 18 bytes long.

The SMFWTM macro instruction returns a code in register 15, which indicates the disposition of the user record as follows:

- 0, which indicates that the record was written without error.
- 4, which indicates that the record was truncated, because it would not completely fit in an empty SMF data set.
- 8, which indicates that the record was not written, because the specified length was less than five bytes in VS1, or less than 18 bytes in VS2.
- 12, which indicates that the record was not written, because the routine was not authorized to write to the SMF data set. (The requesting routine had a non-zero protection key and was not authorized.) This return code does not exist for VS2.
- 16, which indicates that the record was not written, because (a) the writing of records to the SMF data set is prohibited (that is, MAN=NONE was specified in SMFPRM*xx*), or (b) the writing of records is allowed but the SMF data set was full. If the SMF data set is full, it must be dumped before additional SMF records can be written. (See "Operational Considerations" in the chapter "System Information and Requirements" for the procedure for executing the SMF dump program.)
- 20, which indicates that the record was not written, because the IEFU83 exit specified that the record should not be written.

IFASMFR Macro Instruction

You may use the IFASMFR macro instruction in exit routines (or in any problem program application) to symbolically address SMF record fields. The macro is supplied on SYS1.AMODGEN.

The format of the IFASMFR macro instruction is:

```
[label] IFASMFR [n]
```

where:

n
is the record type to be defined. If more than one record type is specified, the record types must be enclosed in parentheses and separated by commas. The values of *n* for VS1 can be 0-15, 17-21, 43-45, 47-49, 62, 64, 68, or 69. The values of *n* for VS2 can be 0, 2-11, 14, 15, 17-22, 26, 31, 34, 35, 40, 43, 45, 47-49, 62, 64, 68-74.

Because of the similarity of record types 14 and 15, whenever record type 15 is specified, record type 14 is defined. Therefore, record types 14 and 15 cannot both be specified in one program using the IFASMFR macro instruction. If both are specified, a multiple definition of record type 14 is produced.

Note: You must enter at least one of the parameters with the IFASMFR macro instruction. If required, a CSECT or DSECT statement must be supplied ahead of the macro instruction.

VS1 and VS2 Exit Routines

The exits available for user written routines in both VS1 and VS2 are fully described in the topics that follow.

IEFUJV—Job Validation Exit

The IEFUJV exit can be taken from both the background and the foreground. IEFUJV receives control, in VS1, just before each JCL 80-character image (other than comment statements) for a job is interpreted and after all JCL images have been interpreted.

In VS2, IEFUJV receives control once before each JCL 80-character image. It also receives control twice more: once after all JCL has been processed by the converter and second after all JCL has been processed by the interpreter. However, IEFUJV does not receive control of comment statements nor for any statement for console started tasks.

A user written routine entered at this exit might do any or all of the following:

- Validate any account fields included in the JOB and EXEC statements by comparison with a standard list.
- Validate or assign the priority.
- Validate or assign the REGION request.
- Validate or assign job time and job step time parameters.
- Control output stream data by assigning a SPACE parameter to SYSOUT statements.
- Check for authorization to use data sets.
- Create installation-defined accounting records.
- Assign the user-identification field to be included in SMF termination records and the SYSOUT records for the job.

Depending on the processing you want to perform, it may be more efficient to check JOB and EXEC statement accounting fields at the exits provided for job initiation and the first job step initiation, respectively. The fields are passed as parameters to those routines, making a statement scan routine unnecessary. You may assign user identification at either of these exits and may write messages to the system output message data set from the IEFACTRT exit routine.

Note: In VS2, the user-communication field is initialized to binary zeros when the converter is started. Data placed in this field by the VS2 IEFUJV exit routine (for entry codes 0-16) will become part of each job, will be accessible by exit routines (except IEFU83) entered in processing each job and will be the same the next time this exit routine is entered. In VS1, the user-communication field is similarly related to the IEFUIV exit.

At entry to the routine, register 1 points to a list of four-byte addresses, as follows:

1. The address of an exit parameter area. (See Figure 32.)
2. The address of an 80-character JCL statement image (in EBCDIC). JCL statements are identical to those listed in the SYSOUT data set; control statements containing only comments, however, are not made available. If a cataloged procedure is being executed, it is expanded before this exit routine receives control; the

sequence of statements is JOB, EXEC PROC=..., EXEC PGM=..., followed by the other statements of the procedure. Override statements immediately precede the statement being overridden.

3. The address of a one-byte area that indicates the type of JCL statement being presented to the exit routine. The indicator will have one of the following binary values:

   0, which indicates a null statement.

   1, which indicates a JOB statement.

   2, which indicates an EXEC statement.

   4, which indicates a DD statement.

   8, which indicates a PROC statement (for symbolic parameter definition).

   16, which indicates that no statement is being presented and that all JCL images have been passed to the exit. In VS2, this entry code indicates that all JCL has been processed by the converter.

   32 (VS2 only), which indicates that no statement is being presented and all JCL images have been processed by the interpreter.

The job validation routine may test and modify any of the operand fields in the job control statements and indicate, through a return code passed to job management, whether processing of this job is to continue. If the user modifies a job control statement, the modified statement is passed to the interpreter for processing. The modified statement appears in the SYSOUT listing.

Editing of the job control statements must not result in additional job control statements or continuation cards. Operand fields being added must not precede the first operand of any JCL statement image.

Before the IEFUJV exit routine returns to the control program, it must place a return code in register 15. A value of 0 indicates that processing of the job should continue; a binary value of 4 indicates that the job is to be canceled.

Sample IEFUJV Routine

The sample IEFUJV exit routine provided in the member SMFEXITS of SYS1.ASAMPLIB checks the validity of a continued JOB statement and of values supplied for REGION, PRTY, TIME, and accounting parameters in JOB statements. Characters from the account number are used to index a table that contains allowable values for these parameters. If any value is found to be invalid, the job is terminated.

IEFUJI—Job Initiation Exit

The IEFUJI exit can be taken from both the background and the foreground. IEFUJI receives control just before each job is initiated.

The job initiation exit may be used to validate job accounting information and in VS1 only to change job priority.

At entry to the routine, register 1 points to a list of four-byte addresses, as follows:

1. The address of an exit parameter area. (See Figure 32.)
2. The address of a 20-byte area containing the programmer's name (in EBCDIC) from the JOB statement. This area is aligned left and, if necessary, padded with blanks.

3. The address of a one-byte area indicating the requested job priority.
4. The address of an area containing accounting information from the JOB statement. If the JOB statement contains no accounting information, the area is one byte of zeros.

The accounting fields are placed in a formatted list for easy access. Figure 34 shows the format of the accounting information that is available to IEFUJI.

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	Number of accounting fields in statement (0 for no fields)
1	1			Consecutive accounting fields[1]

[1] Each entry for an accounting field contains the length of the field (one byte, binary), followed by the field (variable length, EBCDIC). The entry for a null accounting field contains a length of zero. (Null accounting fields are indicated by consecutive commas in the accounting field of a JOB statement.)

Figure 34. Format of Accounting Information

If the installation uses major and minor account numbers with several fields, this exit is easier to use for account number processing than IEFUJV is because of the formatted list.

Before the IEFUJI exit routine returns to the control program, it must place a return code in register 15. A value of 0 indicates that processing of the job is to continue; a binary value of 4 indicates that the job is to be canceled.

Sample IEFUJI Routine

The sample IEFUJI exit routine provided in the member SMFEXITS of SYS1.ASAMPLIB determines how long a job has been in the input job queue before it is initiated. This value and the job priority are written to the SMF data set as a user record.

IEFUSI—Step Initiation Exit

The IEFUSI exit can be taken from both the background and the foreground. IEFUSI receives control just before each job step is initiated, prior to allocation. (If OPT=1 was specified in the SMFPRM*xx* member or entered from the console at IPL time, this exit is not taken.)

The step initiation exit may be used to validate job step accounting information and to write to an installation data set.

At entry to the routine, register 1 points to a list of four-byte addresses, as follows:

1. The address of an exit parameter area. (See Figure 32.)
2. The address of an eight-byte area containing the job step name (in EBCDIC) from the EXEC statement. This area is aligned left and, if necessary, padded with blanks.
3. The address of an eight-byte area containing the program name (in EBCDIC) from the EXEC statement. This area is aligned left and, if necessary, padded with blanks.
4. The address of an area containing step accounting information from the EXEC statement. This area has the format shown in Figure 34.

Before the IEFUSI exit routine returns to the control program, it must place a return code in register 15. A value of 0 indicates that processing of the job should continue; a binary value of 4 indicates that the job is to be canceled.

No sample IEFUSI exit routine is provided in SYS1.ASAMPLIB.

IEFUTL—Time Limit Exit

The IEFUTL exit can be taken from both the background and the foreground. IEFUTL receives control when one of the following time limits expires:

- The job CPU time limit (from the JOB statement).
- The step CPU time limit (from the EXEC statement or default from input subsystem).
- The continuous wait time limit for the job (from SMFPRM*xx*).

If a job time limit is specified on the JOB statement, the limit for each step will be set to job step time limit (selected from the TIME value coded on the EXEC statement or time limit from the input subsystem) or the remaining job time, whichever is smaller.

If no time limit is specified on the JOB statement, each job step is timed individually by using the TIME value from the EXEC statement or the time limit value from the input subsystem.

Specifying TIME=1440 on the JOB statement eliminates all timing for the job. Specifying TIME=1440 on the EXEC statement without a JOB time limit specification eliminates job step timing for the step.

The time limit exit can be used to control and record time expirations. For example, you may use this exit to inform the operator that a job has exceeded the continuous wait time limit and request a reply to cancel the job or extend the time limit.

In VS2, CPU time is collected in two catagories: execution under TCBs and SRBs. The limiting function and exit interfaces only apply to TCB time.

The meaning of an extension to continous wait time differs between VS1 and VS2. Each step begins with the limit as specified by the SMFPRM*xx* JWT parameter. In VS2, each extension resets the limit in effect for the step versus in VS1, the extension only affects the wait currently in process. The new limit for the step in VS2 is set equal to the extension value.

In VS1, the asynchronous exit interface routine sets a step must-complete status before SMF installation exit IEFUTL is given control. Consequently, the initiator will abnormally terminate if the exit enqueues on a resource already enqueued on by the job step task or any of its subtasks. This enqueue can come from within SVCs, for example, the SMFWTM and WTO macros.

At entry to the routine, register 1 points to the address of an exit parameter area. (See Figure 32.) Register 0 will have one of the following binary values:

- 0, which indicates that the job CPU time limit expired.
- 4, which indicates that the job step CPU time limit expired.
- 8, which indicates that the continuous wait time limit for the job expired.

Before the IEFUTL exit routine returns to the control program it must place a return code in register 15. A value of 0 indicates that the job is to be canceled; a binary value of 4 indicates that the job is to continue processing with an additional time allocation (in timer units). In VS2, a binary value of 8 indicating that the job is to continue processing with an additional time allocation specified in seconds. The additional time must be placed in register 1. The number of timer units is determined by the following algorithm:

1 second=38400 timer units

The exit routine should control the number of extensions for a given job to prevent looping. It may record the expiration in the SMF data set or as a message to the console. It may not record the expiration in an installation-defined data set in VS1, but can in VS2.

Note: Time can be extended only within a step. When the step is completed, the next step will never be started if the total job CPU time used is greater than the job CPU time limit including the extension. Also, be sure to return control in the supervisor state , and in VS1, with a protect key of 0.

Note: In VS1 with extended timer support, the smallest extension granted is 2^{20} micro seconds or 1.048576 seconds.

Note: In VS2, the detection of timing out is changed to perodic checking each second to see if the limit has been exceeded. Note that the CPU used field is updated only at task switch. The expected error is from 0 to 1 second but if no other task gains control this can be exceeded.

Sample IEFUTL Routine

The sample IEFUTL exit routine provided in the member SMFEXITS of SYS1.ASAMPLIB causes a job to be terminated if the job CPU time limit or job step CPU time limit has been exceeded. If the continuous wait time limit for the job has been exceeded, the limit is extended twice; on the third entry for exceeding the continuous wait time limit for the job, the job is canceled. Each time the routine is entered for exceeding the continous wait time limit, it writes a record to the SMF data set describing the action taken.

IEFUSO—SYSOUT Limit

The OUTLIM function limits output to spooled data sets; it does not apply to direct SYSOUT data sets. The function is described in *OS/VS1 JCL Reference,* GC24-5099, *OS/VS2 JCL Reference,* GC28-0692, and *OS/VS1 Job Management Logic,* SY24-5161. The IEFUSO exit can be taken from background and foreground jobs. In a VS1 system specifying SMF=FULL or SMF=BASIC the IEFUSO exit receives control when the output limit is reached. The output limit is specified by the OUTLIM parameter on the DD statement or defaults to a SYSGEN value. The SYSGEN default for OUTLIM is specified in the JES macro.

In VS2 systems, the IEFUSO exit can be taken only for data sets having an OUTLIM value specified on the DD statement. The JOB's SYSOUT is limited by a job limit specified on the JES2 macro or overridden by the job's spooled output estimate field, but the IEFUSO exit is not taken when that limit is reached.

Note: The IEFUSO exit is not controlled by the EXT parameter.

The SYSOUT limit exit may monitor the amount of output written to spooled data sets. It cannot write to installation-defined data sets. Unless you install an IEFUSO exit jobs are canceled when the OUTLIM limit is reached.

At entry to IEFUSO, register 1 points to a list of four-byte addresses, as follows:

1. The address of an exit parameter area. (See Figure 32.) (When SMF=BASIC is specified, this field contains zeros.)
2. The address of the DCB for the data set (in VS1 only).

Before the IEFUSO exit routine returns control to the control program, it must place a return code in register 15. A value of 0 indicates that the job step is to be terminated; a binary value of 4 indicates that the output limit is to be increased by the value placed in register 1 and processing is to continue.

Note: If the exit indicates extension of OUTLIM (register 15=4), then, unless the output limit has been increased (register 1 is greater than 0), the exit to the user written routine will again be taken when the next record is written to this SYSOUT data set.

No sample IEFUSO exit routine is provided in SYS1.ASAMPLIB.

IEFU83—SMF Record

IEFU83 receives control when each SMF record is ready to be written to the SMF data set. This exit can be used to select the records to be written or to act on the occurrence of a given record. An example of the latter is asking the operator the reason for an IPL whenever an IPL record is to be written.

At entry to the routine, register 1 points to a four-byte address, which points to the RDW of the SMF record to be written.

Before IEFU83 returns control to the control program, it must place a return code in register 15. A value of 0 indicates that the record is to be written to the SMF data set; a value of 4 indicates that the record is not to be written.

Output from IEFU83 may be directed to the console. In VS2, it cannot be directed to an installation-supplied data set.

Sample IEFU83 Routine

The sample IEFU83 exit routine supplied in the member SMFEXITS of SYS1.ASAMPLIB examines the record to be written. If the record to be written is an IPL record, IEFU83 writes to the operator with a reply request for the record types to be written. If the record is not an IPL record, the return code depends upon the records currently requested.

The sample IEFU83 routine has a special macro definition for "write to operator with reply" so that output normally directed to the operator is suppressed and a standard reply is assumed for testing with TESTEXIT. This macro should be removed if you want the message printed at the console. The sample routine also has special macro definitions for "write to operator" and "wait", which generate no-op instructions.

Note: For VS2 release 1.6 and later releases, you will want to suppress writing of VSAM record types 63 and 67 to the SMF data set or truncate them, if you do not plan to use them for VSAM catalog recovery, or by the IEHUCAT utility. The reason is that these records take up a considerable amount of space on the SMF data set.

IEFACTRT—Termination Exit

The IEFACTRT exit can be taken from both the background and the foreground. IEFACTRT receives control when execution of a job or job step is terminated.[1] (If OPT=1 was specified in the SMFPRM*xx* member or entered from the console at IPL time, this exit is taken only at job termination.)

The termination exit may be used to perform your own unique accounting functions.

At entry to the routine, register 1 points to a list of four-byte addresses, as follows:

1. The address of an exit parameter area. (When SMF=BASIC is specified, the address is that of the jobname.)
2. The address of an eight-byte area containing the job step name (in EBCDIC). This area is aligned left and, if necessary, padded with blanks. At job termination the address is zero.
3. The address of a 20-byte area containing the programmer's name (in EBCDIC). This area is aligned left and, if necessary, padded with blanks.
4. The address of a four-byte area that contains, in the first three bytes, accumulative job CPU[2] time in hundredths of a second (a binary value). The last byte contains the number (binary) of accounting fields in the JOB statement.
5. The address of an area that contains accounting information from the JOB statement. This area has the format described earlier in Figure 34, excluding the first field shown (the number of accounting fields). If the JOB statement contains no accounting information, the area contains one byte of zeros.
6. The address of a four-byte area that contains, in the first three bytes, step CPU[2] time in hundredths of a second (a binary value). The last byte contains the number (binary) of accounting fields in the EXEC statement. At job termination the address is zero.
7. The address of an area that contains accounting information from the EXEC statement. This area has the format described earlier in Figure 34, excluding the first field shown (the number of accounting fields). If the EXEC statement contains no accounting information, the area contains one byte of zeros. At job termination the address is zero.
8. The address of a two-byte area. The first byte is an indicator: if bit 7 is set to 1 when the exit routine is entered, the job has been canceled; if the exit routine sets bit 7 to 1, the job will be canceled. The second byte contains the number of the job step currently being processed. At job termination, this second byte contains the number of steps in the job.
9. The address of a two-byte area containing the termination status (condition or completion code) of the job or job step. (When SMF=BASIC is specified, this field does not exist.)
10. The address of an area containing a four-byte record descriptor word (RDW) immediately followed by the job step termination record (record type 4) or job termination record (record type 5) to be written to the SMF data set. (When SMF=BASIC is specified, this field does not exist.)

At entry to the routine, register 0 contains a binary code indicating the reason for entry;a value of 8 indicates job step allocation (SMF=BASIC only in VS1), a value of 12 indicates job step termination, and a value of 16 indicates job termination.

Output from IEFACTRT may be directed to the console or to SYSOUT. Under SMF, installation information may be written either to the SMF data set or to an installation-supplied data set.

If your IEFACTRT exit routine writes messages for system output, the contents of register 12 must be the same as when the routine was entered, and register 13 must contain the address of a 45-word work area. Figure 35 shows the technique that must be used when IEFACTRT writes to the System Output Message Class.

---

[1] IEFACTRT also receives control at allocation time when SMF=BASIC is specified, in VS1.

[2] In VS2, this is the TCB CPU time.

```
         .
         MVC     36(4,12),MSGADDR    MOVE MESSAGE ADDRESS AND
         MVC     42(2,12),MSGLEN     LENGTH TO SYSTEM TABLE
         L       REG15,VIEFYS        BRANCH AND LINK TO MESSAGE
         BALR    REG14,REG15         ROUTINE
         .
MSGADDR  DC      A(MSG)
MSG      DC      C'message text'
MSGLEN   DC      H'xx'               MESSAGE LENGTH
VIEFYS   DC      V(IEFYS)
```

Figure 35. Writing System Output Messages from IEFACTRT

Before the IEFACTRT exit routine returns to the control program, it must place return codes in registers 1 and 15, as follows:

- If register 1 contains a value of 4, the termination record is not to be written to the SMF data set; if it contains a value other than 4, the termination record is to be written.
- If register 15 contains a value of 4, the remaining job steps are to be canceled; if it contains a value other than 4, processing is to continue.

Sample IEFACTRT Routine

The sample IEFACTRT exit routine provided in the member SMFEXITS of SYS1.ASAMPLIB changes the SMF job termination and job step termination records (unless the job step is flushed) to user records and attempts to write them to the SMF data set. If the data set is full, a message indicating lost SMF records is written to the console. At job termination a record containing the job name, programmer's name, and account number is written to the SYSOUT device.

IEFUJP—Job Purge Exit

IEFUJP receives control when a job, background or foreground, is ready to be purged from the system, that is, after the job has terminated and all the SYSOUT output that pertains to the job has been written.

The VS1 job purge exit can be used, for example, to write additional data statistics found in the common exit parameter area to the SMF data set by using the SMFWTM macro instruction. In VS2, the job purge exit may be used to summarize a job's activities in the system.

In VS1, at entry to the routine, register 1 points to a list of four-byte addresses, as follows:

1. The address of the 72-byte exit parameter area. (See Figure 32.) (When SMF=BASIC is specified, this field contains zeros.)
2. Reserved.
3. The address of a 20-byte area containing the programmer's name (in EBCDIC). This area is aligned left and, if necessary, padded with blanks.
4. The address of a four-byte area that contains, in the first three bytes, job CPU time in hundredths of a second (a binary value). The last byte contains the number (binary) of accounting fields in the JOB statement.
5. The address of the job accounting information. This area has the format described in Figure 34, excluding the first field.

In VS2 at entry to the routine, register 1 points to a list of four-byte addresses, as follows:

1. The address of an exit parameter area. (See Figure 32.)

2. The address of an area containing the job purge record (record type 26) to be written to the SMF data set.

In VS1, IEFUJP does not return a code to the control program.

In VS2, before the IEFUJP exit routine returns to the control program, it must place a return code in register 15. A value of other than 4 indicates that the purge record is to be written to the SMF data set; a value of 4 indicates that the purge record is not to be written.

No sample IEFUJP routine is provided in SYS1.ASAMPLIB.

VS1-Only Exit Routine

The exit available for user written routines in only VS1 is fully described in the topic that follows.

IEFUIV—Input Stream Validation Exit

IEFUIV receives control when the input-stream control routine encounters each 80-character logical record of a JOB statement. IEFUIV is available only when VS1 is used. This exit is not given control for problem programs entered from the console.

The input stream validation exit may verify that the JOB statement is correct and acceptable for processing. This routine may also modify any of the operand fields in the JOB statement.

At entry to the routine, register 1 points to a list of four-byte addresses, as follows:

1. The address of the 72-byte exit parameter area. (See Figure 32.)

2. The address of an 80-character logical record of a JOB statement.

Before the IEFUIV exit routine returns to the control program, it must place a return code in register 15. A value of 0 indicates that the job is to be accepted for processing; a binary 4 indicates that the job is not acceptable for processing.

Note that the SMF data set will not contain any record of jobs not accepted by this exit routine unless the user written exit routine writes it.

The user-communication field in VS1 can be used for inter-job communication only at this exit. The field is initialized to binary zeros when the reader is started or a transient reader is restored. Data placed in this field by this exit routine will become part of each job, will be accessible by exit routines (except IEFU83) entered in processing each job, and will be the same the next time this exit routine is entered, unless a new reader is started or a transient reader is restored.

No sample IEFUIV routine is provided in SYS1.ASAMPLIB.

Testing Exit Routines

Because the exit routines provided by your installation will become a part of the control program, you must test them thoroughly. A test procedure (TESTEXIT) is provided in SYS1.ASAMPLIB to aid in your testing.

Special Requirements for Testing Exit Routines

For testing purposes only, your exit routines must conform to the following requirements.

- A user subpool (0-129) must be specified in GETMAIN macro instructions. (When testing is completed, one of the subpools shown in Figure 31 must be specified for the area used to communicate between exit routines.)

- If the SMFWTM macro instruction is used in any of your routines, you must provide a special macro definition in the routine. The special macro definition writes the TESTEXIT data set defined by the DD card having the DDNAME of MANX. (With the normal SMFWTM macro instruction, the data is written to SYS1.MANX or SYS1.MANY.) You can then process the data without accessing the system accounting data on SYS1.MANX and SYS1.MANY. When testing is completed the macro definition must be removed.

Figure 36 shows the SMFWTM macro instruction that is required for testing.

```
         MACRO
&NAME    SMFWTM  &MSGAD
         AIF     ('&MSGAD' EQ '').E1
         AIF     ('&MSGAD' EQ '(1)').BAL
         AIF     ('&MSGAD'(1,1) EQ '(').REGA
         AGO     .LODIT
.E1      MNOTE   '*** NO OPERAND SPECIFIED ***'
         MEXIT
.BAL     ANOP
         CNOP    0,4
&NAME    BAL     15,*+8
.LIST    DC      V(TSMFWTM)
         L       15,0(15)
         BALR    14,15
         MEXIT
.REGA    ANOP
&NAME    LR      1,&MSGAD(1)
         CNOP    0,4
         BAL     15,*+8
         AGO     .LIST
.LODIT   ANOP
&NAME    LA      1,&MSGAD
         CNOP    0,4
         BAL     15,*+8
         AGO     .LIST
         MEND
```

Figure 36. SMFWTM Macro Definition Required for Testing

TESTEXIT Procedure

Included in the test procedure is an Assembler language source program (also named TESTEXIT). This source program attaches the data generator utility program (IEBDG) to create sample parameter lists; TESTEXIT then calls each exit routine being tested, passing the appropriate parameter list to it. Figure 37 illustrates the input/output and control flow of the TESTEXIT routine.

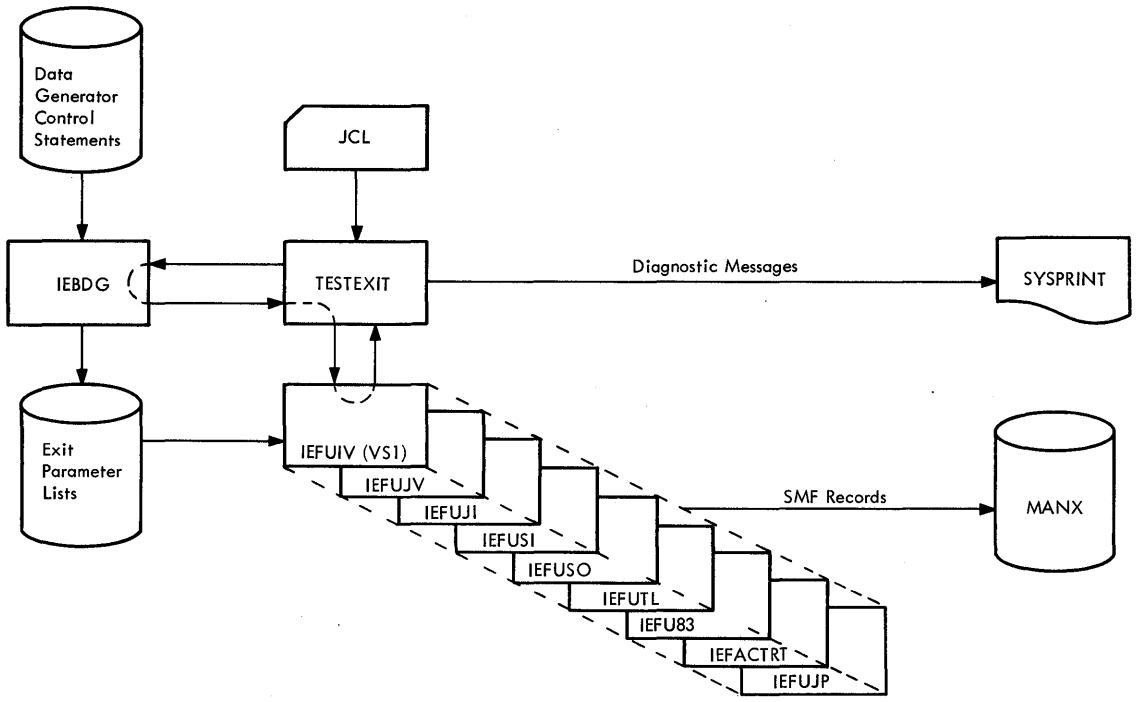

Figure 37. TESTEXIT Input/Output and Control Flow

Figure 38 is an example of JCL that invokes TESTEXIT in an unmodified system. Following is a summary of the operations performed by the procedure shown in Figure 38 :

- The TESTEXIT job assembles the TESTEXIT routine (not illustrated in the figure) and link-edits it with the exit routines being tested. The exit routines must reside in a partitioned data set (EXITLIB).
- The DATAGEN job, using the IEBUPDTE utility program, creates a partitioned data set (DGINPUT) containing control statements for the IEBDG utility program, which will be attached by the TESTEXIT program.
- The TESTING job includes the execution of the TESTEXIT program.

| Reference to IEFUIV is to be ignored in Figure 38 for VS2 systems.

```
//TESTEXIT     JOB  MSGLEVEL=1
//TEST         EXEC ASMFCL
//ASM.SYSIN    DD   *
```

(TESTEXIT Source Module)

```
/*
//LKED.SYSLMOD DD   DSNAME=TESTLIB,VOLUME=SER=231400,
//             UNIT=2314,SPACE=(TRK,(5,2,1)),
//             DISP=(NEW,KEEP)
//LKED.EXITS   DD   DSNAME=EXITLIB,VOLUME=SER=231400,
//             UNIT=2314,DISP=OLD
//LKED.SYSIN   DD   *
 INCLUDE EXITS(IEFUIV,IEFUJV,IEFUJI,IEFUSI,IEFUTL,IEFUSO,     C
               IEFU83,IEFACTRT,IEFUJP)
 ENTRY TESTEXIT
 NAME TESTEXIT
/*
//DATAGEN      JOB  MSGLEVEL=1
//            EXEC PGM=IEBUPDTE,PARM=NEW
//SYSUT2       DD   DSNAME=DGINPUT,UNIT=2314,DISP=(,KEEP),
//            VOLUME=SER=231400,SPACE=(TRK,(10,5,1)),
//            DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//SYSPRINT     DD   SYSOUT=A
//SYSIN   DD  SYSOUT=A
./  ADD      NAME=UIV
```

(IEBDG Control Statements for IEFUIV)

```
./  ADD      NAME=UJV
```

(IEBDG Control Statements for IEFUJV)

```
./  ADD      NAME=UJI
```

(IEBDG Control Statements for IEFUJI)

```
./  ADD      NAME=USI
```

(IEBDG Control Statements for IEFUSI)

```
./  ADD      NAME=USO
```

(IEBDG Control Statements for IEFUSO)

```
./  ADD      NAME=UTL
```

(IEBDG Control Statements for IEFUTL)

```
./  ADD      NAME=U83
```

(IEBDG Control Statements for IEFU83)

```
./  ADD      NAME=ACT
```

(IEBDG Control Statements for IEFACTRT)

Figure 38. TESTEXIT Procedure JCL (Part 1 of 2)

```
./  ADD       NAME=UJP
```
(IEBDG Control Statements for IEFUJP)
```
./  ENDUP
/*
//TESTING      JOB  MSGLEVEL=1
//JOBLIB       DD   DSNAME=TESTLIB,VOLUME=SER=231400,
//             UNIT=2314,DISP=(OLD,KEEP)
//             EXEC PGM=TESTEXIT,
//  PARM='UIV=3,UJV=25,UJI=8,USI=8,USO=5,UTL=5,U83=12,ACT=2,UJP=2'
//INUIV        DD   DSNAME=DGINPUT(UIV),DCB=(LRECL=80,
//             BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),
//             UNIT=2314,VOLUME=SER=231400
//INUJV        DD   DSNAME=DGINPUT(UJV),DCB=(LRECL=80,
//             BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),
//             UNIT=2314,VOLUME=SER=231400
//INUJI        DD   DSNAME=DGINPUT(UJI),DCB=(LRECL=80,
//             BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),
//             UNIT=2314,VOLUME=SER=231400
//INUSI        DD   DSNAME=DGINPUT(USI),DCB=(LRECL=80,
//             BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),
//             UNIT=2314,VOLUME=SER=231400
//INUSO        DD   DSNAME=DGINPUT(USO),DCB=(LRECL=80,
//             BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),
//             UNIT=2314,VOLUME=SER=231400
//INUTL        DD   DSNAME=DGINPUT(UTL),DCB=(LRECL=80,
//             BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),
//             UNIT=2314,VOLUME=SER=231400
//INU83        DD   DSNAME=DGINPUT(U83),DCB=(LRECL=80,
//             BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),
//             UNIT=2314,VOLUME=SER=231400
//INACT        DD   DSNAME=DGINPUT(ACT),DCB=(LRECL=80,
//             BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),
//             UNIT=2314,VOLUME=SER=231400
//INUJP        DD   DSNAME=DGINPUT(UJP),DCB=(LRECL=80,
//             BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),
//             UNIT=2314,VOLUME=SER=231400
//OUTUIV       DD   DSNAME=UIV(OUT),UNIT=2314,DISP=(,PASS),
//             SPACE=(TRK,(10,5,1)),VOLUME=SER=231400,
//             DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//OUTUJV       DD   DSNAME=UJV(OUT),UNIT=2314,DISP=(,PASS),
//             SPACE=(TRK,(10,5,1)),VOLUME=SER=231400,
//             DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//OUTUJI       DD   DSNAME=UJI(OUT),UNIT=2314,DISP=(,PASS),
//             SPACE=(TRK,(10,5,1)),VOLUME=SER=231400,
//             DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//OUTUSI       DD   DSNAME=USI(OUT),UNIT=2314,DISP=(,PASS),
//             SPACE=(TRK,(10,5,1)),VOLUME=SER=231400,
//             DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//OUTUSO       DD   DSNAME=USO(OUT),UNIT=2314,DISP=(,PASS),
//             SPACE=(TRK,(10,5,1)),VOLUME=SER=231400,
//             DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//OUTUTL       DD   DSNAME=UTL(OUT),UNIT=2314,DISP=(,PASS),
//             SPACE=(TRK,(10,5,1)),VOLUME=SER=231400,
//             DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//OUTU83       DD   DSNAME=U83(OUT),UNIT=2314,DISP=(,PASS),
//             SPACE=(TRK,(10,5,1)),VOLUME=SER=231400,
//             DCB=(LRECL=130,BLKSIZE=130,RECFM=FB)
//OUTACT       DD   DSNAME=ACT(OUT),UNIT=2314,DISP=(,PASS),
//             SPACE=(TRK,(10,5,1)),VOLUME=SER=231400,
//             DCB=(LRECL=180,BLKSIZE=180,RECFM=FB)
//OUTUJP       DD   DSNAME=UJP(OUT),UNIT=2314,DISP=(,PASS),
//             SPACE=(TRK,(10,5,1)),VOLUME=SER=231400,
//             DCB=(LRECL=130,BLKSIZE=130,RECFM=FB)
//MANX         DD   UNIT=2314,VOLUME=SER=231400,DSN=MANX,
//             SPACE=(TRK,(3,1)),DISP=(NEW,KEEP),
//             DCB=(BLKSIZE=200,LRECL=196)
//SYSPRINT     DD   SYSOUT=A,DCB=(BLKSIZE=136,LRECL=132)
//DGPRINT      DD   SYSOUT=A
//SYSABEND     DD   SYSOUT=A
/*
```

Figure 38. TESTEXIT Procedure JCL (Part 2 of 2)

Using TESTEXIT

To use the TESTEXIT procedure you must do the following:

- Place your exit routines in a partitioned data set.
- Obtain a punched deck of TESTEXIT from SYS1.ASAMPLIB.
- Modify the procedure to meet your testing requirements.
- Execute the three jobs in the procedure.
- Print the MANX data set if it has data.

The procedure provided in SYS1.ASAMPLIB, without modification, can be used to link-edit the sample exit routines (also in SYS1.ASAMPLIB), generate sample parameter lists, and test the sample exit routines. You should consider linkage-editor and data-generator modifications in adapting the procedure to your testing requirements. These modifications are discussed in the topics that follow.

Linkage Editor Modifications

The linkage editor step of the first job (TESTEXIT), shown in Figure 38, link-edits the TESTEXIT program with the exit routines. You must substitute an INCLUDE control statement specifying the names of the exit routines you are testing.

Data Generator Modifications

The second job (DATAGEN), shown in Figure 38, creates a partitioned data set containing control statements for the IEBDG utility program. The control statements supplied with the procedure will generate samples of standard parameter lists. You should omit control statements and their associated ADD statements for any exit routines you are not testing. If you are testing for special conditions or require additional test parameters, you must make appropriate modifications and additions to the control statements.

Figure 39 shows the JCL that can be used to place user written routines into EXITLIB, a partitioned data set. Reference to IEFUIV is to be ignored in Figure 39 for VS2 systems.

Figure 40 shows the JCL that can be used to obtain a punched deck of TESTEXIT.

Note that you must provide control statements in such an order that the records subsequently generated by the IEBDG utility will be grouped as complete parameter lists that conform in length and format to the exit parameters previously defined in this chapter. (The entry code passed to exits IEFACTRT and IEFUTL in register 0 must be included as a one-byte parameter at the end of the parameter lists for those exits.) For detailed information on the use of IEBDG control statements, refer to *OS/VS Utilities,* GC35-0005.

```
//UPDTE         JOB  MSGLEVEL=1
//             EXEC PGM=IEBUPDTE,PARM=NEW
//SYSUT2       DD   DSNAME=EXITLIB,VOLUME=SER=231400,
//             UNIT=2314,SPACE=(TRK,(10,3,1)),
//             DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//SYSPRINT     DD   SYSOUT=A
//SYSIN        DD   DATA
./  ADD         NAME=IEFUIV
(IEFUIV object deck)
./  ADD         NAME=IEFUJV
(IEFUJV object deck)
./  ADD         NAME=IEFUJI
(IEFUJI object deck)
./  ADD         NAME=IEFUSI
(IEFUSI object deck)
./  ADD         NAME=IEFUTL
(IEFUTL object deck)
./  ADD         NAME=IEFUSO
(IEFUSO object deck)
./  ADD         NAME=IEFU83
(IEFU83 object deck)
./  ADD         NAME=IEFACTRT
(IEFACTRT object deck)
./  ADD         NAME=IEFUJP
(IEFUJP object deck)
./  ENDUP
/*
```

Figure 39. JCL for Entering Exit Routines into EXITLIB

```
//PUNCH    JOB    MSGLEVEL=1
//         EXEC   PGM=IEBPTPCH
//SYSPRINT DD     SYSOUT=A
//SYSUT1   DD     DSNAME=SYS1.ASAMPLIB,DISP=(OLD,KEEP),
//         UNIT=XXXX,VOLUME=SER=XXXXXX[1]
//SYSUT2   DD     UNIT=2540-2
//SYSIN    DD     *
      PUNCH    TYPORG=PO,MAXNAME=1,MAXFLDS=1
      MEMBER   NAME=TESTEXIT
      RECORD   FIELD=(80)
/*
```

[1] The volume and unit parameters depend on your installation's request; check with your system programmer.

Figure 40. JCL for Obtaining a Punched Deck of TESTEXIT

TESTEXIT Execution Modifications

The third job (TESTING), shown earlier in Figure 38, includes execution of the TESTEXIT program. Values for the PARM parameter of the EXEC statement specify which exit routines are to be tested and the number of times each is to be tested. This parameter has the format PARM='xxx=nnn,...,xxx=nnn' where nnn is the number of times an exit routine is to be tested (the maximum value is 255), and xxx is an exit routine identifier. The DD statements to be included depend upon the exit routines to be tested.

Figure 41 shows the exit-routine identifiers, specified on the EXEC statement, and the DD statements that must be included for each exit routine to be tested. DD statements for any other data sets used by your exit routines must be included in the TESTEXIT JCL.

Exit Routine	Identifier	DD Statements
IEFUIV	UIV	INUIV, OUTUIV
IEFUJV	UJV	INUJV, OUTUJV
IEFUJI	UJI	INUJI, OUTUJI
IEFUSI	USI	INUSI, OUTUSI
IEFUTL	UTL	INUTL, OUTUTL
IEFUSO	USO	INUSO, OUTUSO
IEFU83	U83	INU83, OUTU83
IEFACTRT	ACT	INACT, OUTACT
IEFUJP	UJP	INUJP, OUTUJP
Any		MANX, SYSPRINT, DGPRINT, SYSABEND

Figure 41. Parameters and DD Statements for Executing TESTEXIT

Accounting Records

Accounting records describe for each job and job step the user of the system, the resources used, and the completion status of the job or job step. The following record types, which constitute the group of accounting records, are described in this chapter:

- Record type 4—Step Termination.
- Record type 5—Job Termination.
- Record type 6—Output Writer.
- Record type 20—Job Commencement
- Record type 26—Job Purge (VS2 only)
- Record type 34—TS-Step Termination (VS2 only).
- Record type 35—Logoff (VS2 only).
- Record type 40—Dynamic DD (VS2 only).

Fields in these records marked "Reserved" are reserved for use by SMF and are not available for your use.

Record Type 4—Step Termination

Record type 4 is written at the normal or abnormal termination of a job step or when a job step is flushed during or after job initiation. The length is 117 bytes (in VS1) and 137 bytes (in VS2) plus (1) 8 bytes for each device entry and (2) the length of the step accounting fields.

The job step is identified by job log number (job name and time and date that the reader recognized the JOB card for the job), step name, the number of the job step within the job, the user identification field (which may be initialized by the installation to facilitate subsequent sorting of records), and the program name. If accounting numbers (which can be alphameric) were specified in the EXEC statement, they are included.

The record provides operating information such as the time the job step was started and completed, the CPU time, the amount of storage allocated and used, the termination status, the number of records in DD DATA and DD * data sets for the step, the time that device allocation started, the time that the problem program was loaded, and the storage protect key. In VS2, the CPU time is separated into two fields: execution under TCBs and SRBs. Input/output activity is recorded for each non-spooled data set for which the user has a DD statement. Each of these entries lists the EXCP count for the data set, the device class, type, and address.

Data sets are recorded in the order of the step DD statements; they are not identified by name. A user written exit routine can record this order as each statement is validated, or he can do data set recording, if a report on data set activity is needed. See the section 'Data Set Activity Records'.

The EXCP count appears in SMF record types 4, 14, 15, 34, and 40. It indicates the input/output activity required by the job. The EXCP count includes direct EXCPs, program controlled interruptions (PCIs), and channel-end and abnormal-end EXCP returns. When chained scheduling is used, the EXCP count may vary from run to run for the same job. This system function is designed to optimize input/output activity, and, therefore, the number of EXCPs required will depend on system and program interaction at the time the input/output is performed. The variation due to chained scheduling will be reflected in the counts for any data set using chained scheduling.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, of end of step
6	6	4	packed	Date of end of step, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that the reader recognized the JOB card for this job[1]
26	1A	4	packed	Date that the reader recognized the JOB card for this job[1]
30	1E	8	EBCDIC	User identification field from common exit parameter area

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
38	26	1	binary	Step number (first step=1, etc.)
39	27	4	binary	Step initiation time, in hundredths of a second, which is the time of day when this step was selected by the initiator
43	2B	4	packed	Step initiation date
47	2F	4	binary	Number of card-image records in DD DATA and DD * data sets read by the reader for the job step
51	33	2	binary	Step completion code[2]
53	35	1	binary	Step priority (See *OS/VS Supervisor Services and Macro Instructions*, GC27-6979)
54	36	8	EBCDIC	Program name
62	3E	8	EBCDIC	Step name
70	46	2	binary	VS1—Partition size, in 1K units[12] VS2—Private area size, in 1K units[11]
72	48	2	binary	VS1—Reserved VS2—Storage used from top of private area, in 1K units[11]
74	4A	2	binary	VS1—Storage used, in 1K units[12] VS2—Storage used from bottom of private area, in 1K units[11]
76	4C	6	binary	Reserved
82	52	1	binary	Storage protect key (xxxx0000, where xxxx is the key which is described under TCBPKF in *OS/VS1 System Data Areas*, SY28-0605, and *OS/VS2 System Data Areas*, SY28-0606)
83	53	1	binary	Step termination indicators *Bit* — *Meaning When Set* 0 — Reserved 1 — VS1—Canceled by exit IEFUJV[3]; VS2—Reserved 2 — Canceled by exit IEFUJI[3] 3 — Canceled by exit IEFUSI[3] 4 — Canceled by exit IEFACTRT[3] 5 — VS1—Reserved; VS2—Step is to be restarted 6 — 0—Normal completion; 1—ABEND[4] 7 — Step not executed (that is, step was flushed)
84	54	2	binary	Reserved
86	56	4	binary	Device allocation start time of day, in hundredths of a second
90	5A	4	binary	Problem program load time of day, in hundredths of a second
94	5E	1	binary	Reserved
95	5F	3	binary	VS1—Reserved VS2—Step CPU time under SRB, in hundredths of a second[8]
98	62	2	binary	Record indicators *Bit* — *Meaning When Set* 6 — EXCP counts may be wrong (VS2)[7] 7 — 0—Storage is virtual; 1—Storage is real
100	64	2	binary	Offset from beginning of the record header to the relocate section
102	66	2	binary	Length of device entry portion, including this field, which is calculated: (8 x d) + 2, where d=number of devices[6]
For each device, there is an eight-byte entry with the following format:				
		1	binary	Device class from UCBTYP field of unit control block[5]
		1	binary	Unit type from UCBTYP field of unit control block[5]

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
		2	binary	Channel/unit address[5]
		4	binary	EXCP count[7]
After the device entries are the following fields:				
Accounting Section:				
		1	binary	Total length of next three fields: step CPU time, number of accounting fields, and the accounting fields
		3	binary	Step CPU time, in hundredths of a second[8] (under TCBs in VS2 only)
		1	binary	Number of accounting fields
				EXEC statement accounting fields[9]
Relocate Section:				
+0[10]		4	binary	Number of page-ins for this job step
+4		4	binary	Number of page-outs for this job step
End of VS1 Section:				
+8		4	binary	Number of swap outs
+12		4	binary	Number of pages swapped in
+16		4	binary	Number of pages swapped out
+20		4	binary	VIO page-ins
+24		4	binary	VIO page-outs

[1] The job name and the time and date that the reader recognized the JOB card for this job constitute the job log number.

[2] The contents of the completion code field vary according to the condition of termination, as follows:

- X'0ccc', which indicates system ABEND in the job step; ccc is the system ABEND code (see *OS/VS Message Library: VS1 System Codes,* GC38-1003 or *OS/VS Message Library: VS2 System Codes,* GC38-1008).
- X'8ccc', which indicates user ABEND in the job step; ccc is the user ABEND code.
- X'nnnn', which indicates normal completion; nnnn is the contents of the two low-order bytes in register 15 at termination.
- X'0000', which indicates either (1) that the job step was not executed—that is, it was flushed— because of an error, during allocation or in a preceding job step or (2) a return code of 0, indicating normal job completion. To distinguish between a job step flush code and a normal termination code and to distinguish between a system ABEND and a user ABEND, see the step termination indicators field.

[3] Job steps canceled by IEFUJV, IEFUJI, or IEFUSI will not be executed; therefore bit 7 will also be turned on. Job steps canceled by IEFACTRT will cause subsequent job steps to be canceled; in VS1 bit 7 will be turned on for subsequent steps. In VS2, record type 4 is not produced for subsequent job steps.

[4] If this bit indicates an ABEND, check the completion code field to determine the cause of the ABEND. A completion code of 0322 or 0522 indicates that the ABEND was caused by IEFUTL. A completion code of 0722 indicates that the ABEND was caused by IEFUSO.

[5] In VS2, entries for virtual I/O data sets have zero class and type and the channel/unit address is X'0FFF'. Entries for spooled data sets are all 0. For this data see offset 47 of this record and record type 6.

[6] There is an entry for each device assigned to each non-spooled data set in VS1 and for each device assigned to each data set in VS2. For DD *, DD DATA, and SYSOUT data sets, in VS2, the count is 0. For a DD DUMMY data set the entry is set to 0. (A DD DUMMY entry results when a forward reference to a DD name is encountered in the input stream but a DD statement having that DD name is not found or when DD DUMMY is specified.)

[7] In VS2, if a GETMAIN for the expanding TCTIOT control block (where the EXCP counts are maintained) fails, only the existing data sets are counted. If the functional recovery routine is entered, EXCP counting for the step is discontinued and no device entries are produced. The EXCP count does not include PCIs when address space equals REAL.

[8] CPU time is not expected to be constant between different runs of the same step. One or more of the following factors may cause small variations in CPU time: channel program retries (VS1 only), CPU architecture (such as core buffering), cycle stealing with integrated channels, and queue searching (such as enqueue).

[9] Each entry for an accounting field contains the length of the field (one byte, binary), followed by the field (EBCDIC). An omitted field is represented by a length indicator of 0.

[10] The displacement of this field is variable depending on the size of accounting fields and number of devices. The value contained in the field at displacement 100 is the displacement of this field.

[11] When a job step uses ADDRSPC=REAL in VS2, the private area size field contains the size of the continous real storage reserved for the program, the storage used from the bottom field contains how much of the contiguous area was used, and the storage used form the top field contains the storage which does not come out of the contiguous real storage reserved for the program.

[12] When a job step uses ADDRSPC=REAL in VS1, the partition size and storage used fields are the same and are equal to the amount of real storage requested.

Record Type 5—Job Termination

Record type 5 is written at the normal or abnormal termination of a job or when a job step is flushed during or after job initiation. The length is 117 bytes plus the length of the job accounting fields. The maximum length of this record type is 261 bytes.

The job is identified by job log number, programmer name, the installation-supplied user identification field, input class, requested priority, and the accounting fields from the JOB statement. Operating information includes the start and stop time for processing of the job by the reader and the device type and class of the reader device. (The device type and class of reader device is not provided for foreground-initiated background jobs.) The number of records in DD DATA and DD * data sets for the job and the number of steps in the job are included. Job CPU time equals the sum of the job step CPU times. In VS2, the CPU time is separated into two fields: execution under TCBs and SRBs. The job completion code is recorded, along with the storage protect key and a termination code indicating which of four SMF user written exit routines, if any, canceled the job. A flag marks each SYSOUT class used by the job (VS1 only).

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, of end of job
6	6	4	packed	Date of end of job, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that the reader recognized the JOB card for this job[1]
26	1A	4	packed	Date that the reader recognized the JOB card for this job[1]
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	1	binary	Number of steps in the job
39	27	4	binary	Job initiation time, in hundredths of a second, which is the time of day the job was selected by the initiator
43	2B	4	packed	Job initiation date
47	2F	4	binary	Number of card-image records in in DD DATA and DD * data sets for the job read by the reader
51	33	2	binary	Job completion code[2]
53	35	1	binary	Job priority[3]
54	36	4	binary	Time, in hundredths of a second, that the reader recognized the end of the job
58	3A	4	packed	Date that the reader recognized the end of the job

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
62	3E	1	binary	Job termination indicator *Bit* *Meaning When Set* 0 Reserved 1 VS1—Canceled by exit IEFUJV VS2—Reserved 2 Canceled by exit IEFUJI 3 Canceled by exit IEFUSI 4 Canceled by exit IEFACTRT (step exit only) 5 Reserved 6 VS1—0—Normal completion VS1—1—A step within the job abnormally ended VS2—Reserved 7 Reserved
63	3F	5	binary	VS1—SYSOUT class indicator[4] VS2—Reserved
68	44	1	binary	Reserved
69	45	1	binary	Reader device class from UCB
70	46	1	binary	Reader unit type from UCB
71	47	1	binary	Job input class
72	48	1	binary	Storage protect key (xxxx0000, where xxxx is the key which is described under TCBPKF in *OS/VS1 System Data Areas*, SY28-0605, and *OS/VS2 System Data Areas*, SY28-0606)
VS1 Section:				
73	49	3	binary	Reserved
76	4C	8	EBCDIC	VS1—User's logon identifier. Non-terminal oriented jobs have an identifier of 'CENTRAL '; terminal oriented jobs have as the identifier the QIDLGNID field from the Job Management Record for the terminal through which the job was submitted.
84	54	8	binary	VS1—Reserved
VS2 Section:				
73	49	3	binary	Job CPU time under SRB, in hundredths of a second
76	4C	4	binary	Total job service in service units
80	50	4	binary	Total transaction active time (unit is 1024 micro second)
84	54	4	binary	Reserved
88	58	2	binary	Performance group number (range 0-255)
90	5A	2	binary	Reserved
VS1 and VS2 Section:				
92	5C	1	binary	Length of rest of record not including this field
93	5D	20	EBCDIC	Programmer's name
113	71	3	binary	CPU time used by the job[5] in hundredths of a second (under TCBs in VS2 only)
116	74	1	binary	Number of accounting fields following
117	75			JOB statement accounting fields (variable length)[6]

[1] The job name and the time and date that the reader recognized the JOB card for this job constitute the job log number.

[2] The contents of the completion code field vary according to the condition of termination of steps processed by the scheduler, as follows:

X'0ccc', which indicates system ABEND in the last job step that abnormally terminated; ccc is the system ABEND code (see *OS/VS Message Library: VS1 System Codes*, GC38-1003 or *OS/VS Message Library: VS2 System Codes*, GC38-1008).

X'8ccc', which indicates user ABEND in the last job step that abnormally terminated; ccc is the user ABEND code.

X'nnnn', which indicates normal completion; nnnn is the contents of the two low-order bytes in register 15 at termination.
X'0000', which indicates a return code indicating normal job completion.

Job termination indicators are provided at byte 62 of this record. For more detailed information on job step termination, examine record type 4.

[3] The job priority is normally the user-assigned priority (0-13). If the job fails while being scheduled (for example, during device allocation), this field shows a priority of 14, reflecting ABEND processing.

[4] Each bit of the indicator represents the following classes:

Byte 0 *Bit–Class*	*Byte 1* *Bit–Class*	*Byte 2* *Bit–Class*	*Byte 3* *Bit–Class*	*Byte 4* *Bit–Class*
0–A	0–I	0–Q	0–Y	0–6
1–B	1–J	1–R	1–Z	1–7
2–C	2–K	2–S	2–0	2–8
3–D	3–L	3–T	3–1	3–9
4–E	4–M	4–U	4–2	
5–F	5–N	5–V	5–3	
6–G	6–O	6–W	6–4	
7–H	7–P	7–X	7–5	

Usually, this is the class specified by the SYSOUT parameter on the DD statement.

[5] The CPU time is the time used for the problem program by the CPU between job initiation and job termination. CPU time is not expected to be constant between different runs of the same job. One or more of the following factors may cause small variations in CPU time: channel program retries (VS1 only), CPU architecture (such as core buffering), cycle stealing with integrated channels, and queue searching (such as enqueue).

[6] Each entry for an accounting field contains the length of the field (one byte, binary), followed by the field (EBCDIC). An omitted field is represented by a length indicator of 0.

Record Type 6—Output Writer

In VS1, record type 6 is written when the writer has finished processing a SYSOUT class or form within a class for a job. At least one output writer record is written for each SYSOUT class used by the job. If two or more forms are used within a class, one output writer record is produced for each form. In VS2, record type 6 is written for each job output element, which represents a group of data sets differentiated by punch or printer set up and type of output. Record type 6 is also written for spun off data sets. The length is 65 bytes in a VS1 system and 90 bytes in a VS2 system using the JES writer. In all cases, the first 57 bytes are the same.

The output writer is identified by class and form number. The job is identified by job log number and the installation user identification. Output writer activity is recorded by a count of the number of logical records processed, the number of data sets within the record, writer start and end times, and a code indicating any input/output errors.

If an external writer or user-supplied writer is used, SMF produces an incomplete record type 6. An incomplete record type 6 is written for each data set if the writing of records is allowed. In the VS1 case, the number of logical records, and I/O status indicator fields are zero.

For VS2, the incomplete record type 6 differs from the normal JES type 6 record as follows:

- The length is 84 bytes.
- The subsystem generating identification is 0.
- Five data fields are zero.
  - Number of logical records
  - I/O status indicators
  - Form number
  - Data set control indicators
  - JES logical output device name

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that the reader recognized the JOB card for this job[1]
26	1A	4	packed	Date that the reader recognized the JOB card for this job[1]
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	1	EBCDIC	SYSOUT class (blank if not SYSOUT in VS2)
39	27	4	binary	Time of SYSOUT start[2]
43	2B	4	packed	Date of SYSOUT start[2]
47	2F	4	binary	Number of records written by the writer, by form number and by class[3]

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
51	33	1	binary	I/O status indicator in VS1 *Bit* *Meaning When Set* 0–3 Reserved 4 I/O discontinued (remote output only) 5 Input error 6 Output error 7 Input error on SYS1.SYSJOBQE I/O status indicator in VS2 *Bit* *Meaning When Set* 0–4 Reserved 5 Data input error 6 Reserved 7 Control block input error
52	34	1	binary	Total number of data sets processed by writer and included in this record. If multiple copies are produced, each copy is counted.
53	35	4	EBCDIC	Form number
The following field applies when VS1 is used:				
57	39	8	EBCDIC	User's logon identifier. Non-terminal oriented jobs have an identifier of 'CENTRAL '; terminal oriented RES jobs have as the identifier the QIDLGNID field from the Job Management Record for the terminal to which the output was routed; terminal oriented CRJE jobs have the CRJE terminal users ID for the terminal to which the output was routed.
The following fields apply to VS2 when JES2 prodesses the output:				
57	39	1	binary	Reserved
58	3A	2	binary	Subsystem generating ID, JES2=2
60	3C	2	binary	Length of this section, including this field
62	3E	2	binary	Data set control indicators *Bit* *Meaning When Set* 0 Reserved 1 Record represents spun off data sets 2 Operator terminated this data group 3 Operator interrupted this data group 4 Operator restarted this data group 5 Record represents continuation of interrupted data group 6 Operator overrode programmed carriage control (printer only) 7 Punch output was interpreted 8-15 Reserved
64	40	4	EBCDIC	JES assigned job number
68	44	8	EBCDIC	JES logical output device name
76	4C	4	EBCDIC	FCB identifier[4]
80	50	4	EBCDIC	UCS identifier[4]
84	54	4	binary	Approximate page count[4] (skip to carriage control channel one counted as a page)
88	58	2	binary	Output route code

[1] The job name and the time and date that the reader recognized the JOB card for this job constitute the job log number.

[2] In VS2 these fields are the start time and date of the print/punch processor working on the data recorded in the record.

[3] In VS1, the number of records includes job related JOBLOG records if JESPARMS parameter in JOBLOG=YES. Similarly in VS2, if JOBLOG information is produced the counts are included here and in the number of data sets processed.

[4] The contents of these fields apply only to printed output.

Record Type 20—Job Commencement

Record type 20 is written at job initiation. The length is 61 bytes plus the length of the job accounting fields.

Note: For a job canceled at IEFUIV in VS1 no records for the job are written.

This record contains the record type, time stamp (time and date), CPU identification, job log number (job name, entry time, and entry date), programmer's name, user identification, number of accounting fields on the JOB statement, and accounting fields.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that the reader recognized the JOB card for this job[1]
26	1A	4	packed	Date that the reader recognized the JOB card for this job[1]
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	2	binary	Reserved
40	28	20	EBCDIC	Programmer's name
60	3C	1	binary	Number of accounting fields
61	3D			Accounting fields[2]

[1] The job name and the time and date that the reader recognized the JOB card for this job constitute the job log number.

[2] Each entry for an accounting field contains the length of the field (one binary byte), followed by the field (EBCDIC). An omitted field is represented by a length indicator of 0.

Record Type 26—Job Purge

Record type 26 (VS2 only) is written at job purge after all SYSOUT for the job has been processed. The length is 46 bytes, plus the length of the descriptor, events, and actuals sections. The minimum length is 232 bytes.

The job is identified by job log number, programmer name, and JES job number. Operating information includes the start and stop time for processing of the job by the reader, execution and SYSOUT output phases, JES logical input device name, and output statistics.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name
22	16	4	binary	Time, in hundredths of a second, that the reader recognized the JOB card for this job
26	1A	4	packed	Date that the reader recognized the job card for this job
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	4	binary	Reserved
42	2A	2	binary	Subsystem identification—X'0002' signifies JES2
44	2C	2	binary	Section indicator *Bit* *Meaning When Set* 0 Descriptor section present 1 Events section present 2 Actuals section present 3-15 Reserved
Descriptor Section:				
+0		2	binary	Length of descriptor section, including this field
+2		2	binary	Reserved
+4		1	binary	Job information *Bit* *Meaning When Set* 0 Background batch 1 Foreground time sharing 2 System task 3 No journal option 4 No output option 5 TYPE RUN=SCAN 6-7 Reserved
+5		1	binary	Job information *Bit* *Meaning When Set* 0 /*PRIORITY card present[1] 1 /*SETUP card present 2 TYPERUN=HOLD 3 No job log option 4 Execution batching 5 Job entered via an internal reader 6 Job rerun by JES 7 Job canceled by the operator

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
Descriptor Section: (cont'd)				
+6		4	EBCDIC	JES assigned job number
+10		8	EBCDIC	Job identification[2]
+18		20	EBCDIC	Programmer's name
+38		1	EBCDIC	Message class from job card
+39		1	EBCDIC	Job class from job card
+40		1	binary	Initial execution selection priority
+41		1	binary	Selection priority at time of job selection
+42		1	binary	Initial output selection priority
+43		1	binary	Output selection priority at selection
+44		2	binary	Input route code
+46		8	EBCDIC	Logical input device name
+54		4	EBCDIC	Programmer's accounting number[3]
+58		4	EBCDIC	Programmer's room number[3]
+62		4	binary	Estimated execution time[3]
+66		4	binary	Estimated output lines[3]
+70		4	binary	Estimated output punched[3]
+74		4	EBCDIC	Output form number[3]
+78		2	binary	Print copy count (if for all of job)[3]
+80		2	binary	Lines per page[3]
+82		2	binary	Print route code
+84		2	binary	Punch route code
+86		8	EBCDIC	DDNAME of PROCLIB used for JCL conversion
Events Section:				
+0		2	binary	Length of events section, including this field
+2		2	binary	Reserved
+4		4	binary	Reader stop time
+8		4	packed	Reader stop date
+12		4	binary	Converter start time
+16		4	packed	Converter start date
+20		4	binary	Converter stop time
+24		4	packed	Converter stop date
+28		4	binary	Execution start time
+32		4	packed	Execution start date
+36		4	binary	Execution stop time
+40		4	packed	Execution stop date
+44		4	binary	Output processor start time
+48		4	packed	Output processor start date
+52		4	binary	Output processor stop time
+56		4	packed	Output processor stop date

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
Actuals Section:				
+0		2	binary	Length of actuals section, including this field
+2		2	binary	Reserved
+4		4	binary	Number of input cards for job (JCL and SYSIN cards)
+8		4	binary	Output lines generated to spool
+12		4	binary	Output punched cards generated to spool
+16		4	binary	Reserved
+20		4	binary	Lines printed by subsystem
+24		4	binary	Approximate pages printed by subsystem (skip to carriage control channel one is counted as a page)
+28		4	binary	Cards punched by subsystem

[1] Unless /*PRIORITY is *.

[2] The first 4 characters identify the type of job and are followed by the JES assigned job number: JOB for normal job, TSU for TSO jobs, and STC for started task control jobs.

[3] These fields are JES2 defined subfields from the accounting information field in the JOB card or default values assigned for this job.

Record Type 34—TS-Step Termination

Record type 34 (VS2 only) is written each time the TSO logoff function processes a job step termination. The length is 137 bytes plus (1) 8 bytes for each device entry and (2) the length of the step accounting fields.

This record contains the record type, time stamp (time and date), CPU identification, LOGON time, time step initiated, count of TGETs satisfied and TPUTs issued, the time device allocation started, the time the problem program was loaded, termination status, program name, job step name, size of region, and main storage used and the storage protect key. The job step CPU time is recorded in two fields: execution under TCBs and SRBs. Input/output activity is recorded for each data set used by this job step; each entry lists the device class, type, and address, and the EXCP count for the data set. The data-set entry is zeros when the DD entry is TERM, DUMMY, or unallocated DYNAM.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, that step terminated
6	6	4	packed	Date that step terminated, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	User identification field
22	16	4	binary	LOGON time of day in hundredths of a second
26	1A	4	packed	LOGON date, in the form 00YYDDDF, where F is the sign.
30	1E	8	EBCDIC	Reserved for user (blanks)
38	26	1	binary	Step sequence number
39	27	4	binary	Time of day step initiated
43	2B	4	binary	Line-out count, number of TPUTs issued
47	2F	4	binary	Line-in count, number of TGETs satisfied
51	33	1	binary	Step completion code[1]
53	35	1	binary	Step dispatching priority
54	36	8	EBCDIC	Program name
62	3E	8	EBCDIC	Step name (Procedure)
70	46	2	binary	Private area size, in 1K units
72	48	2	binary	Storage used from top of private area, in 1K units
74	4A	2	binary	Storage used from bottom of private area, in 1K units
76	4C	6	binary	Reserved
82	52	1	binary	Storage protect key (See TCBPKF, the protection key field in the TCB, *OS/VS2 System Data Areas*, SY28-0606.)

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
83	53	1	binary	Step termination indicators *Bit* *Meaning When Set* 0 Reserved 1 Reserved 2 Canceled by exit IEFUJI[2] 3 Canceled by exit IEFUSI[2] 4 Reserved 5 Step is to be restarted 6 0—Normal completion 1—ABEND 7 Step not executed (that is, step was flushed)
84	54	2	binary	Reserved
86	56	4	binary	Device allocation start time of day, in hundredths of a second
90	5A	4	binary	Problem program load time of day, in hundredths of a second
94	5E	1	binary	Reserved
95	5F	3	binary	Step CPU time under SRB, in hundredths of a second[5]
98	62	2	binary	Record indicators *Bit* *Meaning When Set* 6 EXCP counts maybe wrong[3] 7 0—storage is virtual 1—storage is real
100	64	2	binary	Offset from beginning of the record header to the relocate section
102	66	2	binary	Length of device entry portion, including this field, which is calculated: (8 x d) + 2, where d=number of devices
For each device, there is an eight-byte entry with the following format:				
		1	binary	Device class from UCBTYP field of unit control block[4]
		1	binary	Unit type from UCBTYP field of unit control block[4]
		2	binary	Channel/unit address[4]
		4	binary	EXCP count[3]
After the device entries are the following fields:				
Accounting Section:				
		1	binary	Total length of next three fields: step CPU time, number of accounting fields, and the accounting fields
		3	binary	Step CPU time under TCB, in hundredths of a second[5]
		1	binary	Number of accounting fields
				Accounting fields[6]
Relocate Section:				
+0[7]		4	binary	Number of page-ins
+4		4	binary	Number of page-outs
+8		4	binary	Number of swap outs
+12		4	binary	Number of TSO swap page-ins
+16		4	binary	Number of TSO swap page-outs
+20		4	binary	Number of VIO page-ins
+24		4	binary	Number of VIO page-outs

[1] The contents of the completion code field varies according to the condition of termination, as follows:

X'0ccc', which indicates system ABEND; ccc is the ABEND code (see *OS/VS Message Library: VS2 System Codes*, GC38-1008).
X'8ccc', which indicates user ABEND; ccc is the user ABEND code.
X'nnnn', which indicates normal completion; nnnn is the contents of the two low-order bytes in register 15 at termination.
X'0000', which indicates either (1) that the job step was not executed (it was flushed) because of an error during allocation or (2) a return code of 0 indicating normal job completion. To distinguish between a job step flush and to distinguish between a system ABEND and a user ABEND, see the step termination indicator field.

Abnormal or normal termination can be determined from the job-termination indicator starting at byte 62 of record type 35.

[2] Job steps canceled by IEFUJI, or IEFUSI will not be executed; therefore, bit 7 will also be turned on.

[3] If a GETMAIN for expanding the TCTIOT control block (where the EXCP counts are maintained) fails, only the existing data sets are counted. If the functional recovery routine is entered, EXCP counting for the step is discontinued, and no device entries are produced. The EXCP count does not include PCIs when address space equals REAL.

[4] Entries for virtual I/O data sets have zero class and type, and X'0FFF' for channel/unit address.

[5] CPU time is not expected to be constant between different runs of the same step. One or more of the following factors may cause small variations in CPU time: CPU architecture (such as core buffering), cycle stealing with integrated channels, and queue searching (such as enqueue).

[6] Each entry in an accounting field contains the length of the field (one byte, binary) followed by accounting information (EBCDIC). An omitted field is represented by a length indicator of 0.

[7] The displacement of this field is variable depending on the number of accounting fields. The value contained in the field at displacement 100 is the displacement of this field.

Record Type 35—LOGOFF

Record type 35 (VS2 only) is written each time a LOGOFF process has been completed. The length is 117 bytes plus the length of each job accounting field.

This record contains the record type, time stamp (time and date), CPU identification, number of TGETs satisfied and TPUTs, storage protect key, session termination status, LOGON priority, logon sequence time, termination indicator, and session CPU time. The CPU time is separated into two fields: execution under TCBs and SRBs.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time of LOGOFF
6	6	4	packed	Date of LOGOFF, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	User identification field
22	16	4	binary	LOGON time of day in hundredths of a second
26	1A	4	packed	LOGON date, in the form 00YYDDDF, where F is the sign
30	1E	8	EBCDIC	Reserved
38	26	1	binary	Number of steps in session
39	27	4	binary	Reserved
43	2B	4	binary	Line-out count, number of TPUTs issued
47	2F	4	binary	Line-in count, number of TGETs satisfied
51	33	2	binary	Job completion code[1]
53	35	1	binary	LOGON priority
54	36	4	binary	LOGON enqueue time of day in hundredths of a second
58	3A	4	packed	LOGON date, in the form 00YYDDDF, where F is the sign
62	3E	1	binary	Termination indicators *Bit Meaning When Set* 0 Reserved 1 Reserved 2 Canceled at exit IEFUJI 3 Canceled at exit IEFUSI 4 Reserved 5 Reserved 6 Reserved 7 Reserved

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
63	3F	9	binary	Reserved
72	48	1	binary	Storage protect key (xxxx0000, where xxxx is the key which is described under TCBPKF in *OS/VS2 System Data Areas,* SY28-0606)
73	49	3	binary	Job CPU time under SRB, in hundredths of a second
76	4C	4	binary	Total session service, in service units
80	50	4	binary	Total transaction active time for session (unit is 1024 micro seconds)
84	54	4	binary	Total number of transactions
88	58	2	binary	Performance group number (range 0-255)
90	5A	2	binary	Reserved
92	5C	1	binary	Length of rest of record not including this field
93	5D	20	EBCDIC	Reserved
113	71	3	binary	Session CPU time under TCBs, in hundredths of a second[2]
116	74	1	binary	Number of accounting fields
117	75			Accounting fields[3]

[1] The contents of the completion code field varies according to the condition of the condition of termination, as follows:

X'0ccc', which indicates system ABEND; ccc is the system ABEND code (see *OS/VS Message Library: VS2 System Codes*, GC38-1008).
X'8ccc', which indicates user ABEND; ccc is the user ABEND code.
X'nnnn', which indicates normal completion; nnnn is the contents of the two low-order types in register 15 at termination.
X'0000', which indicates a return code that indicates normal job completion.

Job termination indicators are provided at byte 62 of this record. For more detailed information on job step termination examine record type 34.

[2] CPU time is not expected to be constant between different runs of the same job. One or more of the following factors may cause small variations in CPU time: CPU architecture (such as core buffering), cycle stealing with integrated channels, and queue searching (such as enqueue).

[3] Each entry in an accounting field contains the length of the field (one byte, binary) followed by accounting information (EBCDIC). An omitted field is represented by a length indicator of 0.

Record Type 40—Dynamic DD

Record type 40 (VS2 only) is written when the dynamic allocation function processes a de-allocation, concatenation, or de-concatenation request. The length is 62 bytes plus 8 bytes for each device entry.

When a de-allocation is processed, a data set entry is produced only for the data set de-allocated. When a concatenation or de-concatenation is processed, a data set entry is produced for all DD entries in the TCTIOT.

This record contains the record type, time stamp (time and date), CPU identification, LOGON time, dynamic allocation function indicators, and a data set entry. Each data set entry consists of the device class, unit type, channel/unit address, and EXCP count. The data set entry is zeros when the DD entry is TERM, DUMMY, or unallocated DYNAM.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, that record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	User identification field
22	16	4	binary	LOGON time of day in hundredths of a second[1]
26	1A	4	packed	Logon date, in the form 00YYDDDF, where F is the sign[1]
30	1E	8	EBCDIC	Reserved for user
38	26	1	binary	Step sequence number
39	27	1	binary	Functional indicators 02—De-allocate 03—Concatenate 04—De-concatenate
40	28	2	binary	Record indicators *Bit Meaning When Set* 6 EXCP count may be wrong[2]
42	2A	18	binary	Reserved
60	3C	2	binary	Length of rest of record including this field
For each device, there is an eight-byte entry with the following format:				
		1	binary	Device class from UCBTYP field of unit control block[3]
		1	binary	Unit type from UCBTYP field of unit control block[3]
		2	binary	Channel/unit address[3]
		4	binary	EXCP count[2]

[1] When dynamic allocation is used by background jobs, these fields are the time and date the reader recognized the JOB card for the job.

[2] If a GETMAIN for expanding the TCTIOT control block (where the EXCP counts are maintained) fails, only the existing data sets are counted. If the functional recovery routine is entered, EXCP counting for the step is discontinued and no device entries are produced. The EXCP count does not include PCIs when address space equals REAL.

[3] Entries for virtual I/O data sets have zero class and type and X'0FFF' for channel/unit address.

Data Set Activity Records

Data set activity records describe the characteristics, activity, and user of data sets. The following record types, which constitute the group of data set activity records, are described in this chapter:

- Record type 14—INPUT or RDBACK Data Set Activity.
- Record type 15—OUTPUT, UPDAT, INOUT, or OUTIN Data Set Activity.
- Record type 17—Scratch Data Set Status.
- Record type 18—Rename Data Set Status.
- Record type 62—VSAM Cluster or Component Opened.
- Record type 64—VSAM Component Status.
- Record type 68—VSAM Entry Renamed.

Fields in these records marked "Reserved" are reserved for use by SMF and are not available for your use.

VSAM produces record types 63 and 67 for use in catalog backup and conversion. These records are documented in the *OS/VS Access Methods Services,* GC26-3836.

Record Type 14—INPUT or RDBACK Data Set Activity

Record type 14 is written whenever a non-VSAM data set that is defined by a DD statement and opened for INPUT or RDBACK processing by a user program is closed or processed by EOV. Record type 14 is not written for a data set defined by a DD * or DD DATA statement. For accounting purposes, the card image count for these data sets is provided in record type 4. The length varies from 288 to 6,412 bytes, depending upon the number of volumes for the data set.

This record contains the device type, EXCP count, data set indicator, data set organization, record format, record length, number of volumes, volume serial numbers, and additional information that depends on whether the data set is on a tape unit or a direct access device and the access method used.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that the reader recognized the JOB card for this job[1]
26	1A	4	packed	Date that the reader recognized the JOB card for this job[1]
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	2	binary	Record indicators *Bit* *Meaning When Set* 0 Reserved 1 Record written by EOV 2 DASD device 3 Temporary data set 4 DCBDSORG=DA 5 DCBDSORG=IS 6 JFCDSORG=IS 7 VIO data set(VS2 only) 8-15 Reserved
40	28	4	binary	Segment sizes *Byte* *Contents* 0 Size of DCB/DEB segment 1 Number of UCB segments[2] 2 Size of each UCB segment 3 Size of extension segment
44	2C	4	binary	Reserved
48	30	16		TIOT[3] segment—a portion of the TIOT, including: *Byte* *Contents* 0 TIOELNGH 1 TIOESTTA 2 TIOEWTCT 3 TIOELINK 4 TIOEDDNM 12 TIOEJFCB 15 TIOESTTC

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
64	40	176		JFCB[3] segment—the JFCB, excluding JFCB extensions
240	F0	24		DCB/DEB[3] segment *Byte* *Contents* 0 DCBDSORG 2 DCBRECFM 3 DCBMACRF 5 DCBOFLGS 6 DCBOPTCD 7 Reserved 8 DEBOFLGS 9 DEBOPATB 10 DEBVOLSQ (Tape extension) 12 DCBBLKCT 16 Data set serial number 22 Reserved (DASD extension) 12 Reserved (may be non-zero) 16 Number of tracks released by the DADSM routine 20 Number of extents released by the DADSM routine 21 Reserved
264	108	24		UCB[3] segment (24 bytes for each UCB in the data set) *Bytes* *Contents* 0 UCBCHA 1 UCBUA 2 UCBVOLI 8 UCBTYP 12 UCBSTAB 13 Number of extents 14 Reserved 16 EXCP count by problem program[4] (Tape extension) 20 UCBFSCT 22 UCBFSEQ (DASD extension) 20 Total number of tracks allocated on the device
		28		ISAM Extension for DCBDSORG=IS *Byte* *Contents* 0 Reserved 2 DCBMAC 3 DCBNLEV 4 DCBRORG3 8 DCBNREC 12 DCBRORG2 14 DCBNOREC 16 DCBRORG1 18 Reserved 19 DEBNIEE 20 DEBNPEE 21 DEBNOEE 22 Number of cylinders in Independent Index Area 24 Number of cylinders in Prime Area 26 Number of cylinders in Independent OVFL Area

[1] The job name and the time and date that the reader recognized the JOB card for this job constitute the job log number.

[2] For ISAM data sets, the number of UCB segments in the order stated is one for the index extent, one per volume for primary extents, and one for the overflow extent.

For BPAM concatenated data sets used as input, there is one UCB segment for each data set in the concatenated data set.

[3] For further information about the contents of the TIOT, JFCB, DCB, DEB, and UCB, refer to *OS/VS1 System Data Areas,* SY28-0605, and *OS/VS2 System Data Areas,* SY28-0606. Note that the channel/unit address for VIO data sets is X'0FFF'.

[4] The EXCP count accumulates over the entire step. Therefore, if a data set is opened and closed twice during a single step, the count in the second record is the sum of all EXCPs for both uses of the data set. The EXCP count in the last type 14 record for the step is equal to the corresponding entry for the data set in the type 4 record.

Record Type 15—OUTPUT, UPDAT, INOUT, or OUTIN Data Set Activity

Record type 15 is written whenever a non-VSAM data set that has been defined by a DD statement and opened for OUTPUT, UPDAT, INOUT, or OUTIN processing by a user program is closed or processed by EOV. The length varies from 288 to 6,412 bytes, depending upon the number of volumes for the data set. Record type 15 is not written for data sets defined as SYSOUT data sets on DD statements. For accounting purposes, the SYSOUT logical record count is included in record type 6.

This record contains the device type, EXCP count, data set indicator, data set organization, record format, record length, number of volumes, volume serial numbers, and additional information that depends on whether the data set is on a tape unit or a direct access device and the access method used.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that the reader recognized the JOB card for this job[1]
26	1A	4	packed	Date that the reader recognized the JOB card for this job[1]
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	2	binary	Record indicators *Bit* *Meaning When Set* 0 Reserved 1 Record written by EOV 2 DASD device 3 Temporary data set 4 DCBDSORG=DA 5 DCBDSORG=IS 6 JFCDSORG=IS 7 VIO data set(VS2 only) 8-15 Reserved
40	28	4	binary	Segment sizes *Byte* *Contents* 0 Size of DCB/DEB segment 1 Number of UCB segments[2] 2 Size of each UCB segment 3 Size of extension segment
44	2C	4	binary	Reserved

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
48	30	16		TIOT[3] segment—a portion of the TIOT, including: *Byte* *Contents* 0 TIOELNGH 1 TIOESTTA 2 TIOEWTCT 3 TIOELINK 4 TIOEDDNM 12 TIOEJFCB 15 TIOESTTC
64	40	176		JFCB[3] segment—the JFCB, excluding JFCB extensions
240	F0	24		DCB/DEB[3] segment *Byte* *Contents* 0 DCBDSORG 2 DCBRECFM 3 DCBMACRF 5 DCBOFLGS 6 DCBOPTCD 7 Reserved 8 DEBOFLGS 9 DEBOPATB 10 DEBVOLSQ (Tape extension) 12 DCBBLKCT 16 Data set serial number 22 Reserved (DASD extension) 12 Relative track (TTR) of the last record processed for a physical sequential data set (left order three bytes followed by a byte of zeros). The value of this field is valid only for writing physical sequential data sets.[5] This field contains zeros for all data set organizations other than physical sequential. 16 Number of tracks released by the DADSM routine 20 Number of extents released by the DADSM routine 21 Reserved
264	108	24		UCB[3] segment (24 bytes for each UCB in the data set) *Byte* *Contents* 0 UCBCHA 1 UCBUA 2 UCBVOLI 8 UCBTYP 12 UCBSTAB 13 Number of extents 14 Reserved 16 EXCP count by problem program[4] (Tape extension) 20 UCBFCST 22 UCBFSEQ (DASD extension) 20 Total number of tracks allocated on the device

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
		28		ISAM extension for DCBDSORG=IS

Byte	*Contents*
0	Reserved
2	DCBMAC
3	DCBNLEV
4	DCBRORG3
8	DCBNREC
12	DCBRORG2
14	DCBNOREC
16	DCBRORG1
18	Reserved
19	DEBNIEE
20	DEBNPEE
21	DEBNOEE
22	Number of cylinders in Independent Index Area
24	Number of cylinders in Prime Area
26	Number of cylinders in Independent OVFL Area

[1] The job name and the time and date that the reader recognized the JOB card for this job constitute the job log number.

[2] For ISAM data sets, the number of UCB segments in the order stated is one for the index extent, one per volume for primary extents, and one for the overflow extent.

[3] For further information about the contents of the TIOT, JFCB, DCB, DEB, and UCB, refer to *OS/VS1 System Data Areas,* SY28-0605, and *OS/VS2 System Data Areas,* SY28-0606. Note that the channel/unit address for VIO data sets is X'0FFF'.

[4] The EXCP count accumulates over the entire step. Therefore, if a data set is opened and closed twice during a single step, the count in the second record is the sum of all EXCPs for both uses of the data set. The EXCP count in the last type 15 record for the step is equal to the corresponding entry for the data set in the type 4 record.

[5] These conditions can be determined by interrogation bytes 0 and 3 of the DCB/DEB segment.

Record Type 17—Scratch Data Set Status

Record type 17 is written whenever a non-VSAM user data set is scratched. (A user's data set is one defined by a user's DD statement either explicitly or implicitly. When a user's DD statement defines a volume, all data sets on that volume are implicitly defined.) The REC parameter determines whether record type 17 is created for only non-temporary data sets or for both temporary and non-temporary data sets. The length is 88 bytes plus 8 bytes for each volume. The length varies from 96 to 2,136 bytes.

This record contains the data set name, number of volumes, and volume serial numbers.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that the reader recognized the JOB card for this job[1]
26	1A	4	packed	Date that the reader recognized the JOB card for this job[1]
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	2	binary	Reserved
40	28	44	EBCDIC	Data set name
84	54	3	binary	Reserved
87	57	1	binary	Number of volumes
88	58			Volume serial number (eight bytes for each volume) *Byte* *Contents* 0-1 Reserved 2-7 Volume serial number (EBCDIC)

[1] The job name and the time and date that the reader recognized the JOB card for this job constitute the job log number.

Record Type 18—Rename Data Set Status

Record type 18 is written whenever any non-VSAM user data set is renamed. The length is 132 bytes plus 8 bytes for each volume. The length varies from 140 to 2,180 bytes, depending upon the number of volumes for the data set.

This record contains the old data set name, new data set name, number of volumes, and volume serial numbers.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that the reader recognized the JOB card for this job[1]
26	1A	4	packed	Date that the reader recognized the JOB card for this job[1]
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	2	binary	Reserved
40	28	44	EBCDIC	Old data set name
84	54	44	EBCDIC	New data set name
128	80	3	binary	Reserved
131	83	1	binary	Number of volumes
132	84			Volume serial number (eight bytes for each volume) *Byte Contents* 0-1 Reserved 2-7 Volume serial number (EBCDIC)

[1] The job name and the time and date that the reader recognized the JOB card for this job constitute the job log number.

Record Type 62—VSAM Cluster or Component Opened

Record type 62 is written at the successful or unsuccessful opening of a VSAM component or cluster. The length is 138 bytes plus 10 bytes for each volume listed.

Record type 62 identifies the VSAM component or cluster and indicates whether it was successfully opened. It names the VSAM catalog in which the component or cluster is defined and gives the volume serial number of the volume on which this catalog is stored. It gives the volume serial number and device type of the volume on which the component or cluster is stored. The job that issued the OPEN macro is identified by job log number and user identification.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that reader recognized JOB statement for this job[1]
26	1A	4	packed	Date reader recognized JOB statement for this job, in form 00YYDDDF, where F is the sign[1]
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	4	binary	Open status indicator *Bit* *Meaning When Set* 0 Successful 1 Security violation, that is, invalid password
42	2A	44	EBCDIC	Name of catalog in which the component or cluster is defined
86	56	6	EBCDIC	Volume serial number of the volume containing the catalog
92	5C	44	EBCDIC	Name of the component or cluster
136	88	2	binary	Number of online volumes containing the component or cluster[2]
For each volume, there is a 10-byte entry with the following format:				
		6	EBCDIC	Volume serial number
		4	binary	Device type[3]

[1] The job name and the time and date that the reader recognized the JOB statement for this job constitute the job log number. If a system task issued the OPEN macro, the job-name field may contain blanks, and the time and date fields contain zeros.

[2] The number of volumes is also the number of pairs of fields in the list of volumes. Each pair is 10 bytes long.

[3] This is the UCBTYP field from the unit control block.

Record Type 64—VSAM Component Status

Record type 64 is written when a VSAM component or cluster is closed, when it becomes necessary to switch to another volume to continue to read or write, and when there is no more space available to continue to process. If a cluster is closed, one record is written for each component in the cluster. The length is 228 bytes plus the length of the list of extents.

Record type 64 indicates whether the component was closed, another volume was switched to, or no additional space was available. It describes the device and volume on which the component is stored. It gives statistics about various processing events that have occurred since the component was defined, such as the number of records in the data component, the number of records that have been inserted, and the number of control intervals that have been split. The job is identified by job log number and user identification.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that reader recognized JOB card for this job[1]
26	1A	4	packed	Date that reader recognized JOB card for this job, in form 00YYDDDF, where F is the sign[1]
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	1	binary	Situation indicator *Bit* *Meaning When Set* 0 Close 1 Volume switch 2 No space available
39	27	1	binary	Indicator of component being processed *Bit* *Meaning When Set* 0 Data component 1 Index component
40	28	44	EBCDIC	Name of the catalog in which the component is defined
84	54	44	EBCDIC	Name of the component
128	80	2	binary	Number of tracks that were requested but could not be allocated
130	82	4	binary	Current high RBA[2]
134	86	2	binary	Length of the extent information in the following fields

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
For each extent, there is a 26-byte entry with the following format:				
+0		4	binary	Beginning cylinder and track, in the form CCHH, where CC is the cylinder number and HH is the track number
+4		4	binary	Ending cylinder and track, in the form CCHH, where CC is the cylinder number and HH is the track number
+8		6	EBCDIC	Volume serial number
+14		2	binary	Channel and unit
+16		2	binary	Spindle identification
+18		4	binary	Unit type[3]
+22		4	binary	Reserved
Statistics Section:[4]				
Accumulative Statistics from Creation Until the Current OPEN:				
+0		4	binary	Length of statistics section, including this field
+4		4	binary	Number of levels in the index
+8		4	binary	Number of extents
+12		4	binary	Number of records in the component
+16		4	binary	Number of records that have been deleted in a component
+20		4	binary	Number of records that have been inserted in the component
+24		4	binary	Number of records that have been updated in the component
+28		4	binary	Number of records that have been retrieved from the component
+32		4	binary	Number of unused control intervals in the component
+36		4	binary	Number of control intervals that have been split in the component
+40		4	binary	Number of control areas that have been split in the component
+44		4	binary	Number of EXCPs
Change from OPEN *in Statistics at time of* EOV *and* CLOSE:				
+48		4	binary	Change in number of levels in the index
+52		4	binary	Change in number of extents
+56		4	binary	Change in number of records
+60		4	binary	Change in number of records that have been deleted in the component
+64		4	binary	Change in number of records that have been inserted in the component
+68		4	binary	Change in number of records that have been updated in the component
+72		4	binary	Change in number of records that have been retrieved from the component
+76		4	binary	Change in number of unused control intervals in the component. This field may be negative
+80		4	binary	Change in number of control intervals that have been split in the component
+84		4	binary	Change in number of control areas that have been split in the component
+88		4	binary	Change in number of EXCPs

[1] The job name and the time and date that the reader recognized the JOB statement for this job constitute the job log number. If a system task caused this record to be written, the job-name field may contain blanks, and the time and date fields contain zeros.

[2] This field is applicable only when the record is written during loading—not for subsequent processing.

[3] This is the UCBTYP field from the unit control block.

[4] All of the fields are present. Inapplicable ones contain zeros. The numbers are cumulative, from the time the object was defined.

Record Type 68—VSAM Entry Renamed

Record type 68 is written when a VSAM catalog entry (a cluster, component, non-VSAM data set, or catalog) is renamed. The length is 170 bytes.

Record type 68 identifies the entry defined and gives the old name and the new name. The job is identified by job log number and user identification.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that reader recognized the JOB card for this job
26	1A	4	packed	Date reader recognized the JOB card for this job, in form 00YYDDDF, where F is the sign
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	44	EBCDIC	Name of the catalog in which the entry is defined
82	52	44	EBCDIC	Old name of the entry
126	7E	44	EBCDIC	New name of the entry

[1] The job name and the time and date that the reader recognized the JOB statement for this job constitute the job log number.

Volume Records

Volume records describe the space available on direct access volumes and certain error statistics for tape volumes (ESV). The following record types, which constitute the group of volume records, are described in this chapter:

- Record type 19—Direct Access Volume.
- Record type 21—ESV.
- Record type 69—VSAM Data Space Defined or Deleted

Fields in these records marked "Reserved" are reserved for use by SMF and are not available for your use.

Record Type 19—Direct Access Volume

Record type 19 is written for each direct access device on line at IPL and when a HALT EOD command or SWITCH SMF command is processed. Record type 19 is also written for a user volume whenever it is demounted. The length is 64 bytes.

This record contains the volume serial number, VTOC address, owner identification number, device type, number of unused alternate tracks, number of unallocated cylinders and tracks, number of cylinders and tracks in the largest free extent, number of unallocated extents, channel and unit address, and module identification for devices having movable address plugs.

Note: Record type 19 is not created for DOS volumes used under the operating system.

Synchronization of clocks is essential in a shared file environment in order to determine the latest status of the shared file.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8		Volume serial number *Byte* *Contents* 0,1 Reserved 2-7 Volume serial number (EBCDIC)
22	16	10	EBCDIC	Owner identification of direct access volume
32	20	4	binary	Device type
36	24	5	binary	VTOC address
41	29	1	binary	DS4VTOCI
42	2A	2	binary	Number of DSCBs
44	2C	2	binary	Number of format 0 DSCBs
46	2E	2	binary	Number of unused alternate tracks
48	30	2	binary	Number of unallocated cylinders
50	32	2	binary	Number of unallocated tracks
52	34	2	binary	Number of cylinders in the largest free extent
54	36	2	binary	Number of tracks in the largest free extent
56	38	2	binary	Number of unallocated extents
58	3A	2	binary	Reserved
60	3C	2	binary	Channel and unit address in the form X'0cuu' where c is the channel address and uu is the unit address
62	3E	2	binary	Module identification or drive number indicating physical identity of devices having movable address plugs. This field is taken from bits 2-7 of sense byte 4 for these devices. (Refer to the component descriptions of these devices for the meaning of sense byte 4.)

Record Type 21—ESV

Record type 21 is written by the Error Statistics by Volume (ESV) option when a user data set on magnetic tape is closed or processed by End-of-Volume. This record is written to the SMF data set only if ESV=SMF is specified in the SCHEDULR macro instruction at system generation. The length is 44 bytes.

This record contains error statistics information about the tape volume which can be analyzed, formatted, and printed by IFHSTATR or your own user routine. (See "IFHSTATR" in *OS/VS Utilities,* GC35-0005.)

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Length of rest of record including this field
16	10	6	EBCDIC	Volume serial number
22	16	2	binary	Channel/unit address
24	18	4	binary	UCB type
28	1C	1	binary	Number of temporary read errors
29	1D	1	binary	Number of temporary write errors
30	1E	2	binary	Number of start I/Os
32	20	1	binary	Number of permanent read errors
33	21	1	binary	Number of permanent write errors
34	22	1	binary	Number of noise blocks
35	23	2	binary	Number of erase gaps
37	25	2	binary	Number of cleaner actions
39	27	1	binary	Tape density (Format of this field is the same as that of DCBDEN, the tape density field in the DCB.)
40	28	2	binary	Block size or 0[1]
42	2A	2		Reserved

[1] This field is 0 if RECFM in the DCB specifies variable or unblocked records, or if you are doing your own EXCP processing.

Record Type 69—VSAM Data Space Defined or Deleted

Record type 69 is written when a VSAM data space is defined, extended, or deleted. Record type 69 is not written, for device activity, when a catalog is defined or when a uinque data set definition occurs. The length is 102 bytes.

Record type 69 gives the total number of free data space extents and the amount of unallocated space on the affected volume after the definition, extension, or deletion of the data space. The job is identified by job log number and user identification.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that reader recognized the JOB card for this job[1]
26	1A	4	packed	Date reader recognized the JOB card for this job, in the form 00YYDDDF, where F is the sign[1]
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	2	binary	Channel and unit
40	28	2	binary	Spindle identification
42	2A	2	binary	Number of free data space extents on the affected volume after the data space is defined, extended, or deleted
44	2C	2	binary	Number of unallocated cylinders in all of the data spaces on the volume
46	2E	2	binary	Number of unallocated tracks in all of the data spaces on the volume in addition to the number of free cylinders
48	30	2	binary	Number of cylinders in the largest continuous unallocated area in any data space on the volume
50	32	2	binary	Number of tracks (in addition to the number of free cylinders) in the largest continuous unallocated area in any data space on the volume
52	34	44	EBCDIC	Name of the catalog in which the data space is defined
96	60	6	EBCDIC	Volume serial number of the volume on which the data space is allocated

[1] The job name and the time and date that the reader recognized the JOB statement for this job constitute the job log number. If this information is not available when this record is to be written, the job-name field contains blanks, and the time and date fields contain zeros.

System Use Records

System use records describe the configuration and SMF options in effect, give system statistics, and record certain events. The following record types, which constitute the group of system use records, are described in this chapter:

- Record type 0—IPL.
- Record type 1—Wait Time (VS1 only).
- Record type 2—Dump Header.
- Record type 3—Dump Trailer.
- Record type 7—Data Lost.
- Record type 8—I/O Configuration.
- Record type 9—VARY ONLINE.
- Record type 10—Allocation Recovery.
- Record type 11—VARY OFFLINE.
- Record type 12—End-of-Day (VS1 only).
- Record type 13—Dynamic Storage Configuration (VS1 only).
- Record type 22—Configuration (VS2 only).
- Record type 31—TIOC Initialization (VS2 only).
- Record type 70—CPU Activity (VS2 only).
- Record type 71—Paging Activity (VS2 only).
- Record type 72—Workload Activity (VS2 only).
- Record type 73—Channel Activity (VS2 only).
- Record type 74—Device Activity (VS2 only).

Fields in these records marked "Reserved" are reserved for use by SMF and are not available for your use.

Record Type 0—IPL

Record type 0 is written after every IPL of the system. It includes the virtual and real storage size and the SMF options in effect. The length is 31 bytes.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	4	binary	Limit in minutes of continuous wait time for the job (from value specified on JWT parameter)
18	12	4	binary	Number of bytes in SMF buffer from value specified in BUF parameter
22	16	4	binary	Number of 1K bytes in virtual storage
26	1A	1	binary	SMF options *Bit* *Meaning When Set* 0 System and job data to be collected 1 System, job, and step data to be collected 2 Exits requested 3 Data set accounting 4 Volume accounting 5 Reserved 6 Temporary data set scratch records 7 Reserved
27	1B	4	binary	Number of 1K bytes in real storage

Record Type 1—System Statistics

Record type 1 (VS1 only) is written after every IPL of the system and at the first job or job step termination following the expiration of a ten-minute interval.[1] The length is 34 bytes.

Elapsed time of day (ordinary CPU-processing time) is divided into ten-minute intervals for the purpose of collecting system statistics. A system statistics record is written at SMF initialization and contains the CPU wait time and paging statistics accumulated during the IPL process. This record marks the beginning of the first ten-minute interval. Subsequent system statistics records are written at the first job or job step termination following the expiration of a ten-minute interval. Each system statistics record contains the wait time and paging counts accumulated during all the ten-minute intervals that expired since the last system statistics record was written and the time of day that the last ten-minute interval ended. At job and step termination, a check is made to see whether at least one ten-minute interval has expired since the last system statistics record was written. Processing continues, as follows:

- If a ten-minute interval has expired, a system statistics record is created, and the wait time and paging count accumulated during the expired ten-minute interval are moved into the record.
- If more than one ten-minute interval has expired, the wait time and paging counts accumulated during the expired ten-minute intervals since the last system statistics record was created is moved into the record.
- If a ten-minute interval has not expired, no system statistics record is created.

Note that the only connection between a job or step and system statistics records is that the termination of a job or step causes SMF to check whether at least one ten-minute interval has expired.

Figure 42 shows how wait time is collected. The process is similar for collecting system paging statistics. When job/step A terminates, three ten-minute intervals have expired. The total wait time collected in these three intervals (783 seconds) is moved to a system statistics record. When job/step B terminates, no ten-minute interval has expired since the last system statistics record was written; therefore, a system statistics record is not written. When job/step C terminates, three intervals have expired. The total wait time collected in these intervals (809 seconds) is moved to a system statistics record.

[1] If a HALT command or SWITCH command is issued before the completion of a ten-minute interval, the wait time collected for that interval is written in a record type 12. If the system continues processing, the next record type 1 contains the wait-time accumulated from the HALT command or SWITCH command to the expiration of the ten-minute interval.

Ten-Minute Intervals[1]							
Wait Time Interval	217	263	303	342	265	202	
Wait Time Collected[2] in a Record Type 1			783			809	
Job/Step Termination			A	B		C	

[1] Elapsed time is divided into ten-minute intervals by SMF and the wait time is collected in seconds for each successive interval.

[2] Collected by totaling the time found in each wait time interval completed before or at each job/step termination. A record type 1 is written when a job/step terminates if a ten-minute interval has expired.

Figure 42. Wait Time Collection

Note: If the stop button is pushed to suspend CPU processing on an IBM System/370, (1) timing of the ten-minute interval is suspended, but (2) the Time-of-Day clock continues to run. The ten-minute interval is based on CPU-processing time, not on the Time-of-Day clock. Therefore, the wait time interval reflected by time of day (time stamp) is equal to the normal ten-minute CPU-processing interval plus the time that CPU processing was stopped.

There is a relationship among wait time, elapsed time, job time, and system time, as shown in the following formula:

Elapsed time=Job time+Wait time+System time

Elapsed time is the length of the measurement interval. It can be obtained by calculating the difference between the time stamp on the first type 1 record and the time stamp on the type 12 record written when a HALT command or a SWITCH command was processed after all jobs processed during the measurement interval have terminated.

Job time is the total time required by all jobs processed in the interval reflected by elapsed time. This value can be obtained by summing the CPU time values from all the type 5 records produced during the elapsed time.

Wait time is the total CPU wait time collected during the interval. This value can be obtained by summing the wait time values from the type 12 record written at the end of the interval and all but the first type 1 records.

System time is the total time required to process system tasks. This value can be calculated when the three other values are known.

The time stamp in the header of a system statistics record reflects the time at which the record was written, not the expiration time of the last ten-minute interval. The expiration time field after the header gives the ending time of the interval.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	4	binary	System wait time, in hundredths of a second, for ten-minute intervals that have expired since the last record type 1 or 12
18	12	4	binary	Expiration time of the end of the interval whose statistics are reported in this record
22	16	4	binary	Total page-ins for the entire system during the interval
26	1A	4	binary	Total page-outs for the entire system during the interval
30	1E	4	binary	Total pages reclaimed for the entire system during the interval

Record Type 2—Dump Header

Record type 2 is written by the SMF dump program at the beginning of a dump data set. The length is 14 bytes.

This record consists of only the standard record header. It indicates the beginning of a dump of the SMF data set from a direct access device to tape. Record type 2 is written directly to the dump data set by the SMF dump program.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was written to the dump data set
6	6	4	packed	Date record was written to the dump data set, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification

Record Type 3—Dump Trailer

Record type 3 is written by the SMF dump program at the end of a dump data set. The length is 14 bytes.

This record consists of only the standard record header. It marks the end of a dump of the SMF data set from a direct access device to tape. Record type 3 is written directly to the dump data set by the SMF dump program.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was written to the dump data set
6	6	4	packed	Date record was written to the dump data set, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification

Record Type 7—Data Lost

Record type 7 is the first record built when an SMF data set becomes available after a period when no SMF data sets were available for recording. Data existing in the SMF buffers is written to the newly available SMF data set before record type 7 is built in a buffer. Consequently, record type 7 is not the first record in the data set. The length is 24 bytes.

This record contains a count of SMF records not written, and the start and end times of the period during which no records were written. (The end time is the time recorded in the record header.)

Note: The time stamp of record types 4, 5, 34, and 35 reflects the time of the end of the job or job step instead of the time that the record was moved to the buffer. Therefore, it is possible for a few records, after the record type 7, to have a time stamp earlier than that of the type 7.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was built in SMF buffer
6	6	4	packed	Date record was built in SMF buffer in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Number of SMF records lost
16	10	4	binary	Time, in hundredths of a second, of start of data loss
20	14	4	packed	Starting date at which no data set was available for recording SMF records, in the form 00YYDDDF, where F is the sign

Record Type 8—I/O Configuration

Record type 8 is written after completion of IPL, following the SET DATE command. The length is 16 bytes plus 4 bytes for each device online at IPL.

This record consists of the standard record header and an entry describing each device that is on line at IPL. Devices are identified by device class, unit type, and device address.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Length of rest of record including this field
For each online device, there is a four-byte entry with the following format:				
		1	binary	Device class from UCBTYP field of unit control block
		1	binary	Unit type from UCBTYP field of unit control block
		1	binary	Channel address
		1	binary	Unit address

Record Type 9—VARY ONLINE

Record type 9 is written when a VARY ONLINE command is processed. The length is 16 bytes plus 4 bytes for each device entry.

This record identifies the system resource being added to the configuration.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Length of rest of record including this field
For each device added, there is a four-byte entry with the following format:				
		1	binary	Device class from UCBTYP field of unit control block
		1	binary	Unit type from UCBTYP field of unit control block
		1	binary	Channel address
		1	binary	Unit address

Record Type 10—Allocation Recovery

Record type 10 is written after successful device allocation recovery. The length is 40 bytes plus 4 bytes for each device entry.

This record identifies the device brought on line, or otherwise made available, by device class, unit type, and device address. The job requiring the allocation is identified by job name, reader start time, and the user identification field. The record is not produced if the operator cancels the job instead of attempting recovery.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	8	EBCDIC	Job name[1]
22	16	4	binary	Time, in hundredths of a second, that the reader recognized the JOB card for this job[1]
26	1A	4	packed	Date that the reader recognized the JOB card for this job[1]
30	1E	8	EBCDIC	User identification field from common exit parameter area
38	26	2	binary	Length of rest of record including this field
For each device being made available, there is a four-byte entry with the following format:				
		1	binary	Device class from UCBTYP field of unit control block
		1	binary	Unit type from UCBTYP field of unit control block
		1	binary	Channel address
		1	binary	Unit address

[1] The job name and the time and date that the reader recognized the JOB card for this job constitute the job log number. If allocation recovery is for a system task, the job name field contains blanks and the reader start time and reader start date fields will contain binary zeros.

Record Type 11—VARY OFFLINE

Record type 11 is written when a VARY OFFLINE command is processed. The length is 16 bytes plus 4 bytes for each device entry.

This record identifies the system resource being removed from the configuration.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Length of rest of record including this field
For each device removed, there is a four-byte entry with the following format:				
		1	binary	Device class from UCBTYP field of unit control block
		1	binary	Unit type from UCBTYP field of unit control block
		1	binary	Channel address
		1	binary	Unit address

Record Type 12—End-of-Day

Record type 12 (VS1 only) is written when a HALT command or a SWITCH command is processed. The length is 34 bytes.

This record includes the system wait time and paging statistics accumulated between the expiration time recorded in the last systems statistics record (record type 1 or 12) and the time a HALT command or a SWITCH command was issued.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	4	binary	System wait time, in hundredths of a second, since the last ten minute interval expired whose data was in the last record type 1 written[1]
18	12	4	binary	Expiration time of the end of the collection period whose statistics are reported in this record
22	16	4	binary	Total page-ins for the entire system during the interval
26	1A	4	binary	Total page-outs for the entire system during the interval
30	1E	4	binary	Total pages reclaimed for the entire system during the interval

[1] If the system continues running after the HALT command or SWITCH command, the next record type 1 contains the wait time accumulated from the HALT command or SWITCH command to the expiration of the next ten-minute interval.

Record Type 13—Dynamic Storage Configuration

Record type 13 (VS1 only) is written at IPL and after each DEFINE command is processed. It shows the amount of storage assigned to each active problem program partition. The length is 16 bytes plus 22 bytes for each partition entry.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Number of bytes remaining, including this field
For each active problem program partition, there is a 22-byte entry with the following format:				
		1	binary	Partition number
		2	binary	Storage in 1K blocks
		2	binary	Reserved
		1	binary	Number of job classes
		16	EBCDIC	Job classes specified by EBCDIC letters A-O, assigned to this partition. The job classes are in their specified order, adjusted right, and padded to the left with blanks.

Record Type 22—Configuration

Record type 22 (VS2 only) is written at IPL and each time a VARY command changes the status of a CPU, storage range, or channel. It identifies the element by type and address, and indicates when the record was written. The lenght is 18 bytes plus 6 bytes for each element in the record.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Record creator indicator *Value Meaning* 1 IPL 2 Vary online 3 Vary offline
16	10	2	binary	Number of 6 byte elements following
CPU Element:				
+0		1	binary	Reserved
+1		1	binary	Processor identification—1
+2		2	binary	CPU model number
+4		1	binary	Reserved
+5		1	binary	CPU address
Channel Element:				
+0		1	binary	Reserved
+1		1	binary	Channel identification—2
+2		2	binary	Channel type and model number
+4		1	binary	Channel address
+5		1	binary	CPU address
Storage Element:				
+0		1	binary	Reserved
+1		1	binary	Storage identification—3
+2		2	binary	Page number of lowest page in storage range
+4		2	binary	Number of pages in storage range

Record Type 31—Initialization

Record type 31 (VS2 only) is written each time the TIOC initialization routine is entered as a result of a MODIFY TCAM command. The length is 54 bytes.

This record contains the record type, time stamp (time and date), CPU identification, total number of time sharing buffers, size of time sharing buffers, maximum number of output buffers allowed each terminal before OWAIT (program wait for output buffers), and maximum number of input buffers allowed each terminal before LWAIT (terminal lockup). It also contains the OWAIT threshold (the number of buffers that must be freed in order to be freed from OWAIT), RESTART threshold (the number of buffers that must be freed in order to be freed from LWAIT), number of buffers reserved on the free queue, and the size of one terminal status block.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Total number of buffers
16	10	2	binary	Buffer size
18	12	2	binary	Reserved
20	14	2	binary	Maximum number of output buffers allowed per terminal before OWAIT[1]
22	16	2	binary	Maximum number of input buffers allowed per terminal before LWAIT[2]
24	18	2	binary	OWAIT threshold. The number of buffers that must be freed in order to be freed from OWAIT.
26	1A	2	binary	RESTART threshold. The number of buffers that must be freed in order to be freed from LWAIT.
28	1C	2	binary	Number of buffers reserved on the free queue
30	1E	2	binary	Reserved
32	20	1	binary	Size of one terminal status block
33	21	21	binary	Reserved

[1] OWAIT is the suspension of the program during input/output to the terminal because no output buffers are available.

[2] LWAIT is the locking up of the terminal user's keyboard because he has filled all the input buffers available to him.

Record Type 70—CPU Activity

Record type 70 (VS2 only) is produced by the MF/1 function.

It contains data to identify the CPU, the amount of WAIT time, and the status of the CPU.

The length is variable.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
Common Control Data:				
+0		2	binary	Size of common control data section
+2		4	packed	Time of day measurement interval started in the form 0HHMMSSF, where F is the sign
+6		4	packed	Date measurement interval started in the form 00YYDDDF, where F is the sign
+10		4	packed	Duration of measurement interval in the form MMSSTTTF, where F is the sign
+14		2	packed	Zero in the form 000F
+16		6	binary	Reserved
+22		2	EBCDIC	MF/1 version number
+24		2	binary	Reserved
+26		4	EBCDIC	Operating system release number and level in the form NNLL
CPU Control Area:				
+0		2	binary	Size of CPU control area
+2		2	binary	Number of CPU data areas in this record
+4		2	binary	Size of a CPU data area
+6		2	binary	Reserved
CPU Data Area:				
+0		8	binary	CPU wait time in TOD clock format (bit 51 = 1 microsecond)
+8		2	binary	CPU identification, such as X'0001'
+10		1	binary	Reserved
+11		1	binary	Configuration activity *Bit Meaning When Set* 6 CPU varied ONLINE or OFFLINE during the measurement interval. Data for this CPU is invalid. 7 CPU currently ONLINE
+12		1	binary	Reserved
+13		3	binary	CPU serial number (6 hexadecimal digits)

Record Type 71—Paging Activity

Record type 71 (VS2 only) is produced by the MF/1 function.

It contains data concerning the page activity measurements of the system and the current size and availability of storage.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
Common Control Data:				
+0		2	binary	Size of common control data section
+2		4	packed	Time of day measurement interval started in the form 00YYDDDF, where F is the sign
+6		4	packed	Date measurement interval started in the form 00YYDDDF, where F is the sign
+10		4	packed	Duration of measurement interval in the form MMSSTTTF, where F is the sign
+14		2	packed	Zero in the form 000F
+16		6	binary	Reserved
+22		2	EBCDIC	MF/1 version number
+24		2	binary	Reserved
+26		4	EBCDIC	Operating system release number and level in the form NNLL
Paging Control Area:				
+0		2	binary	Size of paging control area
+2		2	binary	Size of paging data area
Paging Data Area:				
+0		4	binary	Number of page-ins excluding VIO and SWAP
+4		4	binary	Number page-outs excluding VIO and SWAP
+8		4	binary	Number of page reclaims excluding VIO
+12		4	binary	Number of memory swap sequences
+16		4	binary	Number of pages swapped in
+20		4	binary	Number of pages swapped out
+24		4	binary	Number of VIO page-ins
+28		4	binary	Number of VIO page-outs
+32		4	binary	Number of VIO page reclaims
+36		4	binary	Number of non-VIO page-ins performed in system pagable area (SPA)
+40		4	binary	Number of non-VIO page-outs performed in SPA
+44		4	binary	Number of non-VIO page reclaims performed in SPA
+48		4	binary	Number of VIO page-ins performed in SPA
+52		4	binary	Number of VIO page-outs performed in SPA
+56		4	binary	Number of VIO page reclaims performed in SPA

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
Paging Data Area:				
+60		4	binary	Number of page frames available in main storage
+64		4	binary	Number of page frames defined in main storage
+68		4	binary	Auxiliary total slot count (page frames)
+72		4	binary	Auxiliary data set slot count
+76		4	binary	Auxiliary address space slot count
+80		4	binary	Auxiliary unallocated slot count

Record Type 72—Workload Activity

Record type 72 (VS2 only) is a variable length record produced by the MF/1 function. A record is created for each performance group (PG) defined in the installation performance specification (IPS). The records are created in the order of low to high PG number. Each record contains data on the one-to-eight PG periods for the PG number.

This record contains the IPS name, number of PG periods for a PG number, and for each PG period, the number of terminated transactions, terminated transactions elapsed time, all transactions active time, all transactions service units, and all transactions workload level.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	binary	System identification
Common Control Data:				
+0		2	binary	Size of common control data section
+2		4	packed	Time of day measurement interval started in the form 00YYDDDF, where F is the sign
+6		4	packed	Date measurement interval started in the form 00YYDDDF, where F is the sign
+10		4	packed	Duration of measurement interval started in the form of MMSSTTTF, where F is the sign
+14		2	packed	Zero in the form 000F
+16		2	binary	Performance group number
+18		4	binary	Reserved
+22		2	EBCDIC	MF/1 version number
+24		2	binary	Reserved
+26		4	EBCDIC	Operating system release number and level in the form NNLL
Workload Control Area:				
+0		2	binary	Size of workload control area
+2		2	binary	Number of PG period data areas in this record
+4		2	binary	Size of each PG period data area
+6		2	binary	Highest PG number defined in installation performance specification (IPS)
+8		8	EBCDIC	Name of IPS
Performance Group Period Data Area:				
+0		4	binary	Number of transactions terminated
+4		4	binary	Active time of all transactions (unit 1024 micro second)
+8		4	binary	Service units used in all transactions
+12		4	binary	Elapsed time accumulated by all transactions which terminated in this PG period (unit 1024 micro second)
+16		4	binary	Workload level of all transactions

Record Type 73—Channel Activity

Record type 73 (VS2 only) is a variable length record produced by the MF/1 function.

This record contains data to give the channel identification, its status and activity measurement, such as, start I/Os, and overlap of channel busy with CPU wait.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
Common Control Data:				
+0		2	binary	Size of common control data section
+2		4	packed	Time of day measurement in the form of 00YYDDDF, where F is the sign
+6		4	packed	Date measurement interval started in the form of 00YYDDDF, where F is the sign
+10		4	packed	Duration of measurement interval in the form MMSSTTTF, where F is the sign
+14		2	packed	Sampling cycle length in the form TTTF, where F is the sign
+16		2	binary	Reserved
+18		4	binary	Number of samples
+22		2	EBCDIC	MF/1 version number
+24		2	binary	Reserved
+26		4	EBCDIC	Operating system release number and level in the form NNLL
Channel Control Area:				
+0		2	binary	Size of channel control area
+2		2	binary	Number of channel data areas in this record
+4		2	binary	Size of each channel data area
+6		2	binary	Reserved
Channel Data Area:				
+0		2	binary	CPU identifier
+2		1	binary	Channel identifier
+3		1	binary	Indicators xxxx ....Type of channel 0000 ....Selector channel 0001 ....Byte multiplexor 0010 ....Block multiplexor .... .1.. Invalid channel id .... ..1. Data invalid, varied on or off ....1 Channel is currently online
+4		4	binary	Number of successful SIOs issued to channel
+8		4	binary	Number of samples indicated channel in burst mode (0 for byte multiplexor)
+12		4	binary	Number of samples channel busy and CPU was in wait state (0 for byte multiplexor)

Record Type 74—Device Activity

Record type 74 (VS2 only) is a variable length record produced by the MF/1 function.

This record contains data that identifies the device, its status and measures of activity.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
Common Control Data:				
+0		2	binary	Size of common control data section
+2		4	packed	Time of day measurement interval started in the form 00YYDDDF, where F is the sign
+6		4	packed	Date measurement interval started in the form 00YYDDDF, where F is the sign
+10		4	packed	Duration of measurement interval in the form MMSSTTTF, where F is the sign
+14		2	packed	Sampling cycle length in the form TTTF, where F is the sign
+16		2	binary	UCB device type code in right byte
+18		4	binary	Number of samples
+22		2	EBCDIC	MF/1 version number
+24		2	binary	Reserved
+26		4	EBCDIC	Operating System release number and level in the form NNLL
Device Control Area:				
+0		2	binary	Size of device control area
+2		2	binary	Number of device data areas in this record
+4		2	binary	Size of each device data area
+6		2	binary	Reserved
Device Data Area:				
+0		2	packed	Device address in the form HHHF, where F is the sign
+2		1	binary	Reserved
+3		1	binary	Configuration activity .... ..1. Data invalid, varied on or off ....1 Device is currently online
+4		4	binary	UCB type word
+8		6	EBCDIC	Volume serial number
+14		2	binary	Reserved
+16		4	binary	Number of requests serviced by this device
+20		4	binary	Number of samples when device was busy
+24		4	binary	Sum of number of ENQ requests for this device observed at samples

Subsystem records describe the activities and events of the subsystems. The record number is followed by a letter, such as 43R for RTAM, that identifies the subsystem with which it is associated.

The following record types, which constitute the group of subsystem records, are described in this chapter:

- Record type 43H—JES2 Start (VS2 only)
- Record type 45H—JES2 Stop (VS2 only)
- Record type 47H—SIGNON/Start Line (VS2 only)
- Record type 48H—SIGNOFF/Stop Line (VS2 only).
- Record type 49H—JES2 Integrity (VS2 only).
- Record type 43R—RTAM Start (VS1 only).
- Record type 44R—RTAM Modify (VS1 only).
- Record type 45R—RTAM Stop (VS1 only).
- Record type 47R—LOGON (VS1 only).
- Record type 48R—LOGOFF (VS1 only).
- Record type 49R—RTAM Integrity (VS1 only).

Fields in these records marked "Reserved" are reserved for use by SMF and are not available for your use.

Record Type 43H—JES2 Start

Record type 43H (VS2 only) is written when a START JES2 command is processed. The length is 24 bytes.

This record contains the record type, time stamp (time and date), CPU identification, and the JES2 options.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Subsystem identification—X'0002' signifies JES2
16	10	2	binary	Reserved
18	12	2	binary	Length of rest of record, not including this field
20	14	3	binary	Reserved
23	17	1	binary	JES2 options: *Bit* *Meaning When Set* 0 Format the spool 1 Cold start 2 Request automatic initiator 3 List replacement card option 4–7 Reserved

Record Type 45H—JES2 Stop

Record type 45H (VS2 only) is written when the system operator enters a STOP JES2 command. It is not written at normal or abnormal system termination and at abnormal termination of JES2. The length is 24 bytes.

This record contains the record type, time stamp (time and date), CPU identification and JES2 completion code.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit* *Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Subsystem identification—X'0002' signifies JES2
16	10	2	binary	Reserved
18	12	2	binary	Length of rest of record, not including this field
20	14	2	binary	Indicators *Bit* *Meaning When Set* 0 Abnormal termination of JES2 1-15 Reserved
22	16	2	binary	JES2 completion code

Record Type 47H—SIGNON/Start Line

Record type 47H (VS2 only) is written by HASPRTAM under two conditions: (1) when the system operator enters a Start Line command and (2) when a remote user signs on. The length is 22 bytes, plus the length of the identification and message sections.

This record contains the record type, time stamp (time and date), CPU identification, JES2 subsystem event, remote name, line name, password, and message text.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Subsystem identification—X'0002' signifies JES2
16	10	2	binary	Reserved
18	12	2	binary	Length of rest of record, not including this field
20	14	2	binary	JES2 SIGNON/Start Line subsystem event—X'0001' signifies SIGNON and X'0002' signifies Start Line
Identification Section:				
+0		2	binary	Length of identification section, including this field
+2		8	EBCDIC	Remote name
+10		8	EBCDIC	Line name
+18		8	EBCDIC	Password
Message Section:[1]				
+0		2	binary	Length of SIGNON message section, including this field
+2		36	EBCDIC	Message text[2]

[1] If this is a Start Line record, this section will not appear.

[2] If this is a SIGNON record, information from columns 35 through 70 of the SIGNON card image is placed in this field.

Record Type 48H—SIGNOFF/Stop Line

Record type 48H (VS2 only) is written by HASPRTAM under two conditions: (1) when the system operator enters a Stop Line command and (2) when a remote user signs off. The length is 71 bytes.

This record contains the record type, time stamp (time and date), CPU identification, JES2 subsystem event, remote name, line name, password, and summary of data and error activity.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Subsystem identification—X'0002' signifies JES2
16	10	2	binary	Reserved
18	12	2	binary	Length of rest of record, not including this field
20	14	2	binary	JES2 SIGNOFF/Stop Line subsystem event—X'0001' signifies SIGNOFF and X'0002' signifies Stop Line
22	16	2	binary	Reserved
24	18	8	EBCDIC	Remote name
32	20	8	EBCDIC	Line name
40	28	8	EBCDIC	Password
48	30	4	binary	Number of EXCPs
52	34	4	binary	Number of negative acknowledgements to write text
56	38	4	binary	Number of data checks to read text
60	3C	4	binary	Number of time outs to read text
64	40	4	binary	Sum of all other line errors
68	44	3	EBCDIC	Line adapter address of UCB

Record Type 49H—JES2 Integrity

Record type 49H (VS2 only) is a duplicate of record type 47H except the password is not valid.

This record contains the record type, time stamp (time and date), CPU identification, JES2 subsystem event, remote name, line name, password (invalid), and message text.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 6 VS2
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Subsystem identification—'0002' signifies JES2
16	10	2	binary	Reserved
18	12	2	binary	Length of rest of record, not including this field
20	14	2	binary	JES2 SIGNON subsystem event—X'0001' signifies SIGNON
Identification Section:				
+0		2	binary	Length of identification section, including this field
+2		8	EBCDIC	Remote name
+10		8	EBCDIC	Line name
+18		8	EBCDIC	Password (invalid)
Message Section:				
+0		2	binary	Length of SIGNON message section, including this field
+2		36	EBCDIC	Message text[1]

[1] Information from columns 35 through 70 of the SIGNON card image is placed in this field.

Record Type 43R—RTAM Start

Record type 43R (VS1 only) is written by RTAM during RTAM initialization. The length is 42 bytes, plus a six-byte entry for each line created.

This record contains the record type, time stamp (time and date), CPU identification, RTAM start procedure name, maximum number of readers and writers supported, number of times to activate, and number of lines to start at this time. For each line created, a six-byte entry containing line name and unit address is made.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Subsystem identification—X'0001' signifies RTAM
16	10	2	binary	Reserved
18	12	2	binary	Length of rest of record, not including this field
20	14	8	EBCDIC	RTAM start procedure name
28	1C	8	binary	Reserved
36	24	1	binary	Maximum number of readers supported
37	25	1	binary	Maximum number of writers supported
38	26	1	binary	Number of entries in LINE table
39	27	1	binary	Number of line DCTs
40	28	1	binary	Number of lines to activate
41	29	1	binary	Number of lines to start at this time
For each line DCT created, there is a six-byte entry, as follows:				
		3	EBCDIC	Line name
		3	EBCDIC	Unit address, in the form cuu where c is the channel and uu is the unit

Record Type 44R—RTAM Modify

Record type 44R (VS1 only) is written by RTAM whenever a MODIFY RTAM command is issued. The length is 30 bytes, plus a six-byte entry for each line modified.

This record contains the record type, time stamp (time and date), CPU identification, RTAM start procedure name, type of modification (start, stop, or restart), and number of lines modified at this time. For each line modified, a six-byte entry containing line number and unit address is made.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Subsystem identification—X'0001' signifies RTAM
16	10	2	binary	Reserved
18	12	2	binary	Length of rest of record, not including this field
20	14	8	EBCDIC	RTAM start procedure name
28	1C	1	binary	MODIFY type *Value Meaning* 1 Start 2 Stop 3 Restart
29	1D	1	binary	Number of lines modified at this time
For each modified line, there is a six-byte entry, as follows:				
		3	EBCDIC	Line number
		3	EBCDIC	Unit address, in the form cuu where c is the channel and uu is the unit

Record Type 45R—RTAM Stop

Record type 45R (VS1 only) is written by RTAM when a STOP RTAM command is issued. The length is 30 bytes.

This record contains the record type, time stamp (time and date), CPU identification, RTAM start procedure name, STOP status, and number of lines started when the STOP was received.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in the form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Subsystem identification—X'0001' signifies RTAM
16	10	2	binary	Reserved
18	12	2	binary	Length of rest of record, not including this field
20	14	8	EBCDIC	RTAM start procedure name
28	1C	1	binary	STOP begun or ended: *Value Meaning* 0 Begun 1 Ended
29	1D	1	binary	Number of lines started when STOP was received

Record Type 47R—LOGON

Record type 47R (VS1 only) is written by RTAM whenever a valid LOGON record is received by RTAM. The length is 152 bytes.

This record contains the record type, time stamp (time and date), CPU identification, QID entry, passback area, and LOGON record.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Subsystem identification—X'0001' signifies RTAM
16	10	2	binary	Reserved
18	12	2	binary	Length of rest of record, not including this field
20	14	48	binary	QID entry
68	44	4	binary	Passback area—X'8000 0000'
72	48	80	EBCDIC	LOGON record

Record Type 48R—LOGOFF

Record type 48R (VS1 only) is written by RTAM whenever a LOGOFF record is received by RTAM. The length is 68 bytes.

This record contains the record type, time stamp (time and date), CPU identification, and QID entry.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Subsystem identification—X'0001' signifies RTAM
16	10	2	binary	Reserved
18	12	2	binary	Length of rest of record, not including this field
20	14	48	binary	QID entry

Record Type 49R—RTAM Integrity

Record type 49R (VS1 only) is written by RTAM whenever an invalid LOGON record is received by RTAM. The length is 152 bytes.

This record contains the record type, time stamp (time and date), CPU identification, QID entry, passback area, and LOGON area.

The format is:

Decimal Displacement	Hexadecimal Displacement	Field Size	Data Format	Contents
0	0	1	binary	System indicator *Bit Meaning When Set* 7 VS1
1	1	1	binary	Record type
2	2	4	binary	Time, in hundredths of a second, record was moved to SMF buffer
6	6	4	packed	Date record was moved to SMF buffer, in form 00YYDDDF, where F is the sign
10	A	4	EBCDIC	System identification
14	E	2	binary	Subsystem identification—X'0001' signifies RTAM
16	10	2	binary	Reserved
18	12	2	binary	Length of rest of record, not including this field
20	14	48	binary	QID entry
68	44	4	binary	Passback area—X'FF00 0000'
72	48	80	EBCDIC	LOGON area

Appendix A: Field-to-Record Cross-Reference

This appendix lists all of the fields in the SMF records in alphabetical order and gives the record type containing each field and the displacement of the field within the record.

Record types 22, 26, 31, 34, 35, 40, 43H, 45H, 47H–49H, and 70–74 are available in a VS2 system. Record types 1, 12, 13, 43R–45R and 47R–49R are available only when you are using SMF in a VS1 system.

Note: Under the DECIMAL DISPLACEMENT column the following abbreviations will appear:

ACT	Actuals section
ACCT	Accounting section
CHAN	Channel section
COM	Common control data section
CPUD	CPU data section
CPU	CPU section
DESC	Descriptor section
DEV	Device data section
EVET	Events section
IDEN	Identification section
MSG	Message section
PAGE	Page data section
PERF	Performance group period and data section
REL	Relocate section
STOR	Storage section

These abbreviations indicate the section of the record where the field is found.

Field Name	Record Type	Displacement	
		Decimal	Hex.
Accounting fields	4	ACCT	
	5	117	75
	20	61	3D
	34	ACCT	
	35	117	75
Active time of all transactions	72	PERF	
Alternate tracks available	19	46	2E
Auxiliary slot counts	71	PAGE	
Block size	21	40	28
Buffer size	31	16	10
Channel and unit address	19	60	3C
	21	22	16
	69	38	26
Channel address	22	CHAN	
Channel identifier	73	CHAN	
Channel indicators	73	CHAN	
Channel type and model number	22	CHAN	
Cleaner actions	21	37	25
Converter start time and date	26	EVET	

Field Name	Record Type	Displacement	
		Decimal	Hex.
Converter stop time and date	26	EVET	
CPU address	22	CPU	
CPU configuration activity	70	CPUD	
CPU identification	70	CPUD	
CPU identifier	73	CHAN	
CPU model number	22	CPU	
CPU serial number	70	CPUD	
CPU wait time	70	CPUD	
CPU time used by job	5	113	71
Current high RBA	64	130	82
Cylinders available (unallocated)	19	48	30
Data set name	17	40	28
Data set control indicators	6	62	3E
DCB/DEB segment	14	240	F0
	15	240	F0
Device added entry	9	14	E
	10	38	26
Device address	74	DEV	
Device allocation time	4	86	56
	34	86	56
Device configuration activity	74	DEV	
Device entry	4	104	68
	34	104	68
	40	62	3E
Device removed entry	11	14	E
Device type	19	32	20
DDNAME of PROCLIB used for JCL conversion	26	DESC	
DSCBs (total number)	19	42	2A
DS4VTOCI	19	41	29
Elapsed time accumulated by all transactions transmitted in this performance group period	72	PERF	
Erase gaps	21	35	23
Estimated execution time	26	DESC	
Estimated output lines	26	DESC	
Estimated punched output	26	DESC	
Execution priority	26	DESC	
Execution start time and date	26	EVET	
Execution stop time and date	26	EVET	
Extents available (unallocated)	19	56	38
FCB identifier	6	76	4C
Form number	6	53	35

Field Name	Record Type	Displacement	
		Decimal	Hex.
Format 0 DSCBs	19	44	2C
Functional indicators	40	39	27
Generated output records	26	ACT	
I/O status	6	51	33
Input route code	26	DESC	
Indicator of component or cluster being processed	64	39	27
JES2 assigned job number	26	DESC	
	26	74	4A
JES2 completion code	45H	22	16
JES2 execution selection priority	6	64	40
JES2 logical output device name	6	68	44
JES2 indicators	45H	20	14
JES2 options	43H	23	17
	26	DESC	
JES2 output selection priority	26	DESC	
JES2 subsystem event	47H	20	14
	48H	20	14
	49H	20	14
JFCB segment	14	64	40
	15	64	40
Job class	26	DESC	
Job completion code	5	51	33
	35	51	33
Job and CPU time under SRB	5	73	49
	35	73	49
Job identification	26	DESC	
Job information	26	DESC	
Job initiation time and date	5	39	27
Job input class	5	71	47
Job service	35	76	4C

Field Name	Record Type	Displacement	
		Decimal	Hex.
Job name	4	14	E
	5	14	E
	6	14	E
	10	14	E
	14	14	E
	15	14	E
	17	14	E
	18	14	E
	20	14	E
	26	14	E
	62	14	E
	64	14	E
	68	14	E
	69	14	E
Job options	26	DESC	
Job priority	5	53	35
Job termination indicator	5	62	3E
Length of actuals section	26	ACT	
Length of descriptor section	26	DESC	
Length of events section	26	EVET	
Length of identification section	47H	IDEN	
Length of rest of record	43H	18	12
Limit of continuous wait time for job	0	14	E
Line name	47H	IDEN	
	48H	32	20
	49H	IDEN	
Line number	44R	30	1E
Lines per page	26	DESC	
Line-in count, number of TGETs satisfied	34	47	2F
	35	47	2F
Line-out count, number of TPUTs issued	34	43	2B
	35	43	2B
Logical input device name	26	DESC	
LOGON area	49R	72	48
Logon enqueue time	35	54	36
Logon priority	35	53	35
LOGON record	47R	72	48
Logon time	34	22	16
	35	22	16
	40	22	16
Main storage occupancy time	34	39	27
Main storage used	34	74	4A

Field Name	Record Type	Displacement	
		Decimal	Hex.
Maximum number of input buffers allowed per terminal before LWAIT	31	22	16
Maximum number of output buffers allowed per terminal before OWAIT	31	20	14
Maximum number of readers	43R	36	24
Measurement interval start time and date	70	COM	
	71	COM	
	72	COM	
	73	COM	
	74	COM	
Measurement interval duration	70	COM	
	71	COM	
	72	COM	
	73	COM	
	74	COM	
Message class from job card	26	DESC	
Message text	47H	MSG	
	49H	MSG	
MF1 version number	70	COM	
	71	COM	
	72	COM	
	73	COM	
	74	COM	
MODIFY type	44R	28	1C
Module identification or drive number of devices having movable address plugs	19	62	3E
Name of catalog in which component or cluster is defined	64	40	28
	68	38	26
Name of catalog in which component is defined	69	52	34
Name of component or cluster	62	92	5C
	64	84	54
New data set name	18	84	54
New name of component or cluster	68	128	7E
Noise blocks	21	34	22
Non-VIO paging in system pageable area	71	PAGE	
Number of buffers	31	14	E
Number of buffers reserved on free queue	31	28	1C
Number of bytes in real storage	0	27	1B
Number of bytes in SMF buffer	0	18	12
Number of bytes in virtual storage	0	22	16
Number of cylinders in largest free extent	19	52	36
Number of cylinders in the largest continous unallocated area in any data space on the volume	69	48	30
Number of data sets processed by writer for this job	6	52	34
Number of data spaces on the affected volume after the data space is defined, extended, or deleted	69	42	2A

Field Name	Record Type	Displacement	
		Decimal	Hex.
Number of input cards	26	ACT	
Number of lines modified	44R	29	1D
Number of lines started when STOP received	45R	29	1D
Number of online volumes containing the component or cluster	62	136	88
Number of pages in storage range	22	STOR	
Number of records in DD DATA and DD * read for the job	5	47	2F
Number of records written by writer	6	47	2F
Number of regions	41	80	50
Number of samples	73	COM	
	74	COM	
Number of samples indicated channel busy and CPU wait state	73	CHAN	
Number of samples indicated channel in burst mode	73	CHAN	
Number of samples indicating device busy	74	DEV	
Number of successful SIDs issued to channel	73	CHAN	
Number of SMF records lost	7	14	E
Number of request serviced by this device	74	DEV	
Number of steps in job	5	38	26
Number of tracks in addition to the number of free cylinders in the largest continous unallocated area in any data space on the volume	69	50	32
Number of tracks in largest free extent	19	54	36
Number of tracks requested but not allocated	64	128	80
Number of transactions	35	84	54
Number of unallocated tracks in all data spaces on the volume in addition to the number of free cylinders	69	46	2E
Number of volumes	17	87	57
	18	131	83
non-VIO paging in system pageable area	71	PAGE	
Old data set name	18	40	28
Old name of component or cluster	68	82	52
Online I/O device entry	8	14	E
Open status indicator	62	38	26
Operating system release number and level	70	COM	
	71	COM	
	72	COM	
	73	COM	
	74	COM	
Output form number	26	DESC	
Output lines generated to spool	26	ACT	
Output punched cards generated to spool	26	ACT	
Output priority	26	DESC	
Output processor start time and date	26	EVET	
Output processor stop time and date	26	EVET	

Field Name	Record Type	Displacement	
		Decimal	Hex.
Output route code	6	88	58
OWAIT threshold	31	24	18
Owner identification of direct access volume	19	22	16
Page count	6	84	54
Page frames available in main storage	71	PAGE	
Page frames defined in main storage	71	PAGE	
Page-ins for job step	4	REL	
Page-ins for system during interval	1	22	16
	12	22	16
Page-ins for TSO session	34	REL	
Page number of lowest page in storage range	22	STOR	
Page-outs for job step	4	REL	
Page-outs for system during interval	1	26	1A
	12	26	1A
Page-outs for TSO session	34	REL	
Pages reclaimed for system during interval	1	30	1E
	12	30	1E
Pages swapped in	4	REL	
	34	REL	
Pages swapped out	4	REL	
	34	REL	
Page swapping	71	PAGE	
Paging excluding VIO and SWAP	71	PAGE	
Partition characteristic entry	13	16	10
Partition or region size	4	70	46
Passback area	47R	68	44
	49R	68	44
Password	47H	IDEN	
	48H	40	28
	49H	IDEN	
Performance group number	5	88	58
	35	88	58
	72	COM	
Print copy count	26	DESC	
Print route code	26	DESC	
Printed lines	26	ACT	
Printed pages	26	ACT	
Private area size	4	70	46
	34	70	46
Problem program load time	4	90	5A
	34	90	5A

Field Name	Record Type	Displacement	
		Decimal	Hex.
Program name	4	54	36
Programmer's accounting number	26	DESC	
Programmer's name	5	93	5D
	20	40	28
	26	DESC	
Programmer's room number	26	DESC	
Punch form number	26	DESC	
Punch route code	26	DESC	
Punched cards	26	ACT	
QID entry	47R	20	14
	48R	20	14
	49R	20	14
Read errors (permanent)	21	32	20
Read errors (temporary)	21	28	1C
Reader device class and type	5	69	45
Reader stop time and date	26	EVET	
Record creator indicator	22	14	E
Record indicators	4	98	62
	14	38	26
	15	38	26
	40	40	28
Record type	All	1	1
Records in DD DATA and DD * read for the job step	4	47	2F
Remote name	47H	IDEN	
	48H	24	18
	49H	IDEN	
Reserved for your use	34	30	1E
	35	30	1E
	40	30	1E
Restart threshold	31	26	1A
RTAM start procedure	43R	20	14
	44R	20	14
	45R	20	14
Sampling cycle length	73	COM	
	74	COM	
Section indicator	26	44	2C
Segment sizes	14	40	28
	15	40	28
Service units used in all transactions	72	PERF	
Session CPU time	35	113	71
Situation indicator	64	38	26

Field Name	Record Type	Displacement	
		Decimal	Hex.
Size of one terminal status block	31	32	20
Size of region	34	70	46
SMF options	0	26	1A
Start I/Os	21	30	1E
Starting date at which no data set was available for recording SMF records	7	20	14
Step completion code	4	51	33
	34	51	33
Step CPU time	4	ACCT	
	34	ACCT	
Step CPU time under SRB	4	95	5F
	34	95	5F
Step dispatching priority	34	53	35
Step initiation time and date	4	39	27
Step name	4	62	3E
	34	62	3E
Step number	4	38	26
Step priority	4	53	35
Step sequence number	34	38	26
	40	38	26
Step termination indicators	4	83	53
	34	83	53
Steps in session	35	38	26
Stop begun or ended	45R	28	1C
Storage protect key	4	82	52
	5	72	48
	34	82	52
	35	72	48
Storage used	4	74	4A
Storage used from bottom of private area	4	74	4A
	34	74	4A
Storage used from top of storage area	4	72	48
	34	72	48

Field Name	Record Type	Displacement	
		Decimal	Hex.
Subsystem identification	6	58	3A
	26	42	2A
	43H	14	E
	45H	14	E
	47H	14	E
	48H	14	E
	49H	14	E
	43R	14	E
	44R	14	E
	45R	14	E
	47R	14	E
	48R	14	E
	49R	14	E
Sum of number of ENQ requests for this observed at sample	74	DEV	
Swaps that occurred for session	34	REL	
Swaps that occur for step	4	REL	
	34	REL	
SYSOUT class	6	38	26
SYSOUT class indicator	5	63	3F
SYSOUT classes for session	35	63	3F
System identification	All	10	A
System indicator	All	0	0
System model identifier	All	12	C
System wait time	1	14	E
System wait time since last record type 1	12	14	E
Tape density	21	39	27
Terminal monitor program name	34	54	36
Termination indicators	35	62	3E
Time and date of SYSOUT start	6	39	27
Time and date reader recognized end of job	5	54	36

Field Name	Record Type	Displacement	
		Decimal	Hex.
Time and date reader recognized JOB card	4	22	16
	5	22	16
	6	22	16
	10	22	16
	14	22	16
	15	22	16
	17	22	16
	18	22	16
	26	22	16
	62	22	16
	64	22	16
	68	22	16
	69	26	1A
Time and date record moved to SMF buffer	All	2	2
Time of end of collection period	12	18	12
Time of end of interval	1	18	12
Time of start of data loss	7	16	10
Time reader recognized job card	20	22	16
Time record was written to the dump data set	2	2	2
	3	2	2
TIOT segment	14	48	30
	15	48	30
Tracks available (unallocated)	19	50	32
Transaction active time	5	80	50
	35	80	50
Transaction terminated	72	PERF	
TSO swap page-ins	34	REL	
TSO swap page-outs	34	REL	
UCB device type code	74	COM	
UCB segment	14	264	108
	15	264	108
UCB type	21	24	18
UCS identifier	6	80	50
UCB type word	74	DEV	
Unit address	44R	33	21

Field Name	Record Type	Displacement	
		Decimal	Hex.
User-identification field from common exit parameter area	4	30	1E
	5	30	1E
	6	30	1E
	10	30	1E
	14	30	1E
	15	30	1E
	17	30	1E
	18	30	1E
	20	30	1E
	26	30	1E
	34	14	E
	35	14	E
	40	14	E
	62	30	1E
	64	30	1E
	68	30	1E
	69	30	1E
User-identification field from QIDLGNID field	5	76	4C
	6	57	39
VIO page-ins	4	REL	
	34	REL	
VIO page-outs	4	REL	
	34	REL	
VIO paging	71	PAGE	
VIO paging in system pageable area	71	PAGE	
Volume serial number	17	88	58
	18	132	84
	19	14	E
	74	DEV	
Volume serial number of volume containing the catalog	62	86	56
Volume serial number of volume on which data space is allocated	69	96	60
VTOC address	19	36	24
Workload of all transactions	72	PERF	
Write errors (permanent)	21	33	21
Write errors (temporary)	21	29	1D

Indexes to systems reference library manuals are consolidated in *OS/VS Master Index*, GC28-0602. For additional information about any subject listed below, refer to other publications listed for the same subject in the *Master Index*.

Note: If more than one page number is given, the primary discussion is listed first. The entries in the index appear the same way they appear in the body of the book, which means that entries with bold type or italic type in the body of the book appear in bold type or italic type in the index.

J

L

M

N

O

P

Q

R

S

T

U

V

W

Your Comments, Please . . .

OS/VS System Management Facilities (SMF)

GC35-0004-5

Your comments and constructive criticism regarding this book will help us to improve it so that you may be better served. Each comment will be reviewed carefully by those responsible for the publication. Comments and suggestions become the property of IBM.

Requests for copies of publications or for assistance in using IBM systems and programs should be directed to your local IBM representative.

Thank you for your cooperation.

Comments:

No postage necessary if mailed in the U.S.A.

First Class
Permit No. 568
Boulder, Colorado

Business Reply Mail
No Postage Necessary If Mailed In The U. S. A.

Postage will be paid by:

IBM Corporation
Post Office Box 1900
Boulder, Colorado 80302

Attention:
Programming Publications

Printed in U.S.A. GC35-0004-5

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)

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
(USA Only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)