* GC24-5115-2
File No. S370-34

Systems

OS/VS1 System Management Facilities (SMF)

Release 6

IBM

Third Edition (May 1978)

This edition applies to Release 6 of OS/VS1 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information contained herein; before using this publication in connection with the operation of IBM systems, consult the *IBM System/370 Bibliography*, GC20-0001 for the editions that are applicable and current.

This edition, GC24-5115-2, is a major revision of GC24-5115-1. It incorporates GC24-5129-0, the supplement issued in support of OS/VS1 Subsystem Attachment Support (SU 6). It also contains information on the OS/VS1 Programmed Cryptographic Facility, Program Number 5740-XY5 (SU 21). Changes to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of Amendments

For a list of changes, see page 3.

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

This publication has been produced by IBM Corporation, Programming Publications, Dept. G60, P.O. Box 6, Endicott, New York 13760.

Summary of Amendments for GC24-5115-2 VS1 Release 6

This manual now contains information on two VS1 Selectable Units:

- SU 6 – OS/VS1 Subsystem Attachment Support (SU ID 5741-606)
- SU 21 – OS/VS1 Programmed Cryptographic Facility, Program Number 5740-XY5 (SU ID 5741-621)

Subsystem Attachment Support documentation formerly published in Supplement GC24-5129-0 is now included in this manual. This support uses SMF record type 51 for subsystem data.

The Programmed Cryptographic Facility is an IBM Program Product. It generates SMF record type 82 for security data.

A number of minor clarifications and corrections have also been made throughout this manual.

Summary of Amendments for GC24-5115-1 VS1 Release 6

Programming Features

- IBM 3800 Printing Subsystem – New fields have been appended to SMF record type 6 to gather accounting information for the IBM 3800.
- RTAM – A new field has been added to SMF record type 44 to record modifications to the characteristics of remote workstations.
- ACF/VTAM – A new SMF record type (50) has been added to aid in gathering ACF/VTAM tuning statistics.

Publication Features

- Notes regarding the generation of VSAM records (SMF record types 62, 63, 64, 67, 68, and 69) by system tasks have been added.
- Notice of an exit routine restriction to the effect that VSAM data sets are not supported for SMF exits from the system reader, writer, RTAM, or VTAM has been included.
- The suffix "R" (signifying RTAM) has been dropped from references to record types 43-45 and 47-49.
- A number of minor clarifications and corrections have been made throughout this manual.

Summary of Amendments for GC24-5115 VS1 Release 5

This publication is a major revision of *OS/VS System Management Facilities (SMF)*, order number GC35-0004-6. It contains only VS1 information.

Programming Features

- RTAM – New fields have been added to SMF record types 43R-45R to reflect the modifications made to RES to support SNA devices. Where applicable, RTAM additions have also been made to the text and to the IFASMFR macro instruction.
- Record type 22 (Configuration) – The 40 bytes describing the 3850 Mass Storage System (MSS) units online at IPL have been defined.
- Record type 64 (VSAM Component Status) – Five new record fields describing data set characteristics have been added.
- SMF dump program return code – A X'10' return code in register 15 indicating that the operator attempted to dump the active SMF data set has been added.
- Modifying JCL statements – A restriction for adding or modifying JCL-statement operand fields has been added to the IEFUIV and IEFUJV exits.

Publication Features

- VS2 information – All VS2 information has been removed.
- Dumping the SMF data sets – Two sample procedures for dumping the SMF data sets to a standard-labeled tape by means of the operator START command have been added.
- User-written exit routines – The modules that interface with each SMF exit routine have been added.
- SMF records – The following information has been added to the SMF records:
  - The modules that write each SMF record.
  - The symbolic addresses of the SMF record fields as defined in the IFASMFR macro instruction.
  - The data area fields where SMF obtains the information for each field in each SMF record.
- Record descriptor word (RDW) – A note indicating that the address passed to the SMFWTM macro instruction must point to the RDW has been added.
- Storage protect keys – A statement indicating that all user-written exit routines receive control with the system enabled for interrupts and in storage protect key 0 has been added.
- Figure on SMF in the operating system – The system routines that pass control to the SMF exits have been added.
- SMF storage requirements – Figure 6 has been added to summarize the additional storage required for SMF.
- IEFUTL interlock warning – A sample ENQ macro instruction has been added to assist the IEFUTL user in minimizing the chances of a system interlock.

Preface

The objective of this book is to help installation managers and system programmers plan for, install, and use SMF in a VS1 system. This book explains:

- The five categories of information gathered by SMF data-collection routines.
- When each SMF record is written and the information contained in each record.
- The basic relationship of SMF to the operating system and to user-written exit routines.
- How to add SMF to a VS1 operating system by using the SCHEDULR macro instruction.
- How to add user-written exit routines to the appropriate SMF exits either before or after SYSGEN.
- The system areas that require additional storage for SMF and approximately how much additional storage is needed in each area.
- How to execute the SMF dump program.
- How to use an SMFxxxxx member of SYS1.PARMLIB to define the use of SMF.
- How to write records to installation-defined or SMF-defined data sets.
- How to plan and write exit routines that will monitor jobs or job steps at various points during their processing cycle.
- How to test user-written exit routines using the TEXTEXIT procedure.
- How to use sort/merge programs to sequence SMF data.
- How to design report programs that format and print the data from SMF records.

Related Publications

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

- *Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS,* GC35-0014, describes the VARY ONLINE,S and VARY OFFLINE,S commands, which cause SMF to produce record type 22.
- *Operator's Library: OS/VS1 Reference,* GC38-0110, describes the HALT EOD and SWITCH SMF commands, which dump the SMF data sets; it also describes several other commands that cause SMF to produce records.
- *OS/VS Data Management Services Guide,* GC26-3783, describes the record descriptor word (RDW), which is used to write records in the SMF data sets.
- *OS/VS Message Library: VS1 System Codes,* GC38-1003, defines system completion codes.
- *OS/VS Message Library: VS1 System Messages,* GC38-1001, explains SMF messages.
- *OS/VS1 Programmer's Reference Digest,* GC24-5091, lists the resource names used by system tasks.

- *OS/VS1 Supervisor Services and Macro Instructions,* GC24-5103, provides detailed information on return codes from the ENQ macro instruction.
- *OS/VS1 Utilities,* GC26-3901, describes the IEBUPDTE, IEBDG, IEBPTPCH, and IFASTATR utility programs, which are used to:
  1) enter an SMFxxxxx member into SYS1.PARAMLIB,
  2) generate samples of standard parameter lists for user-written exit routines,
  3) obtain listings of sample routines, and
  4) format and write information from SMF record type 21.
- *OS/VS-VM/370 Assembler Programmer's Guide,* GC33-4021, illustrates the ASMFCL cataloged procedure, which assembles and link-edits user-written exit routines.
- *OS/VS1 JCL Reference,* GC24-5099, summarizes the OUTLIM parameter, which is used with user-written IEFUSO exit routines.
- *OS/VS1 Planning and Use Guide,* GC24-5090, has information on how to handle accounting information when SMF=BASIC is specified, and how to use SMFxxxxx parameters when automated system initialization (ASI) is used.
- *OS/VS1 RES: System Programmer's Guide,* GC28-6878, has information on remote entry services and RTAM, for which SMF produces record types 43-45 and 47-49.
- *Advanced Communications Function for VTAM (ACF/VTAM) System Programmer's Guide,* SC38-0258, has information on ACF/VTAM, for which SMF produces record type 50.
- *OS/VS1 and OS/VS2 MVS Programmed Cryptographic Facility General Information Manual,* GC28-0942, describes the cryptographic facility that uses SMF record type 82.
- *OS/VS1 System Data Areas,* SY28-0605, describes the TIOT, JFCB, DCB, DEB and UCB data areas contained in SMF record types 14 and 15.
- *OS/VS1 Virtual Storage Access Method (VSAM) Logic,* SY26-3841, describes the VSAM catalog records contained in SMF record types 63 and 67.

(For information on the PL/1 and Sort/Merge program products, see *PL/1 Language,* SC33-0009, and *OS/VS Sort/Merge Programmer's Guide,* SC33-4035.)

Notational Conventions

The parameters and instructions shown in this publication use the following notational conventions:

- **Bold** type letters, words, and symbols: code them exactly as shown.
- *Italic* type letters, words, and symbols: substitute specific information.
- ␣ (blank characters): code a blank.
- { } (braces): code only one of the items or use the default value. Do not code the braces.
- [ ] (brackets): code any enclosed item or items – they are optional. Do not code the brackets but specify commas if they are included with the items enclosed in the brackets.
- | (OR signs): select only one of the items separated by the OR signs.
- Underscores: if you do not code a value, the underscored one is the default.

Contents

Figures

Introduction

SMF (System Management Facilities) is an optional feature of OS/VS1 that collects and records several types of information:

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

By creating analysis and report routines, installation managers or system programmers can use SMF-collected information in a variety of ways. For instance, they can use it for accounting system usage or for measuring system performance.

Note: SMF does not collect or record information for system tasks; for example, SMF does not supply information for JES.

SMF also has exits that can link to user-written routines for monitoring a job or job step at various points during its processing cycle. The user-written routines can perform functions such as: cancel jobs, write user-defined records to the SMF data set, access installation-defined data sets, or enforce standards such as user identification, resource allocation, and maximum execution time.

Because SMF data-collection routines and exit routines are independent of one another, one can use them in combination or separately. For example: by analyzing the information obtained by SMF data-collection routines, an installation manager can estimate a time limit for all jobs or job steps running on the system. Any job exceeding this time limit is automatically terminated. If, however, the installation manager would like to allow certain jobs to exceed the time limit, he or she can use the IEFUTL SMF exit to link to a user-written time-limit routine. Through the IEFUTL SMF exit, the execution time for selected jobs can be extended.

Information Collected by SMF

The information collected by SMF is formatted into many records that are categorized as follows:

- Accounting records
- Data set records
- Subsystem records
- System records
- Volume records

The following sections list the types of records in each of these categories, and describe some of the information they contain. For a detailed description of each SMF record and its format, see the chapter "SMF Records".

Accounting Records

The SMF accounting records are:

Type 4 – Step Termination
Type 5 – Job Termination
Type 6 – Output Writer
Type 20 – Job Initiation

These records describe the resources used by a job or job step. They include information such as:

- Job log and user identification
- Problem program name and start time
- Step name and number
- Job/step start and end times
- Amount of storage allocated and used
- Storage protect key
- Job/step priority
- User's logon identifier
- Job/step CPU time
- Accounting fields from JOB and EXEC statements
- I/O status indicators
- Job/step termination status
- Job/step completion code
- EXCP count, device class, unit type, and channel/unit addresses
- Page-ins and page-outs done to each partition
- SYSOUT class and SYSOUT start and end times

Data Set Records

The SMF data set records are:

Type 14 – INPUT or RDBACK Data Set Activity
Type 15 – OUTPUT, UPDAT, INOUT, or OUTIN Data Set Activity
Type 17 – Scratch Data Set Status
Type 18 – Rename Data Set Status
Type 62 – VSAM Component or Cluster Opened
Type 63 – VSAM Entry Defined
Type 64 – VSAM Component or Cluster Status
Type 67 – VSAM Entry Deleted
Type 68 – VSAM Entry Renamed

These records describe the characteristics, activity, and user of data sets. They include information such as:

- Job log and user identification
- Portions of the TIOT, JFCB, DCB, DEB and UCB data areas
- Data set and catalog names
- Number of volumes
- Volume serial numbers
- Extents on volume
- EXCP count and device type
- Number of records in data component
- Record formats and lengths
- Newly defined, altered, or deleted catalog entries

Subsystem Records

The SMF subsystem records are:

Type 43 – RTAM Start
Type 44 – RTAM Modify
Type 45 – RTAM Stop
Type 47 – RTAM Logon
Type 48 – RTAM Logoff
Type 49 – RTAM Integrity
Type 50 – ACF/VTAM Tuning Statistics
Type 51 – Subsystem Data
Type 82 – Security

Record types 43 through 49 describe the activities and events of the remote terminal access method (RTAM). They include information such as:

- Maximum number of readers and writers supported
- Number of line and logical port DCTs
- Line numbers and channel/unit addresses
- QID entries
- Types of MODIFY or STOP operator commands
- LOGON records

Record type 50 maps data useful in adjusting certain ACF/VTAM and NCP sysgen and START command variables to improve ACF/VTAM performance. Record type 51 is a standard record header for use by subsystems, which specify the length and content of the record. Record type 82 describes the events and operations of the Programmed Cryptographic Facility (Program Number 5740-XY5).

System Records

The SMF system records are:

Type 0 – IPL
Type 1 – System Statistics
Type 2 – Dump Header
Type 3 – Dump Trailer
Type 7 – Data Lost
Type 8 – I/O Configuration
Type 9 – VARY ONLINE
Type 10 – Allocation Recovery
Type 11 – VARY OFFLINE
Type 12 – End-of-Day
Type 13 – Dynamic Storage Configuration
Type 22 – Configuration

These records describe the system configuration and SMF options in effect, give system statistics, and record the occurrence of specific events. They include information such as:

- Job log and user identification
- SMF options
- Real and virtual storage sizes
- System wait time
- System page-ins, page-outs, and page reclaims
- Count of SMF records generated but not written during period when SMF data sets were not available for writing
- Start and end times of period without SMF recording
- Device class, unit type, and channel/unit addresses
- Storage assigned to each partition
- Partition number and job class
- MSS units online at IPL

Volume Records

The SMF volume records are:

Type 19 – Direct Access Volume
Type 21 – Error Statistics by Volume
Type 69 – VSAM Data Space Defined, Extended, or Deleted

These records describe the space available on direct access volumes, give error statistics for tape volumes, and describe data spaces in a VSAM catalog. By using the IFHSTATR utility program or user-written routines that examine the tape information, one can address problems of volume deterioration. (See "IFHSTATR" in *OS/VS Utilities*.) The volume records include information such as:

- Number of unused alternate tracks
- Number of unallocated cylinders and tracks
- Number of cylinders and tracks in the largest free extent
- Owner identification
- Volume serial number
- Channel/unit addresses
- Number of read and write errors
- Volume on which data space is allocated
- Catalog in which data space is defined
- Number of free data space extents

SMF in the Operating System

Figure 1 summarizes the functions of SMF in the operating system. This section, which briefly describes Figure 1, assumes that user-written exit routines are supplied for all SMF exits; all SMF-formatted records are written to the SMF data set; and user-written analysis and report routines are supplied. In any real application, of course, the user-written routines that are supplied and the records that are specified to be written to the SMF data set depend upon the installation's requirements.

Initializing SMF

During system initialization, the SMF initialization routine receives control from the master scheduler and:

1. Checks for the existence and validity of an SMFxxxxx member of SYS1.PARMLIB. This member contains the SMF parameters that define the use of SMF in the operating system. If SYS1.PARMLIB does not contain an SMFxxxxx member, (1) the operator may enter the SMF parameters from the console, or (2) the initialization routine uses the IBM-supplied default member SMFPRM00. (For more information, see "SMFxxxxx Parameters" in the chapter "Defining the Use of SMF".)

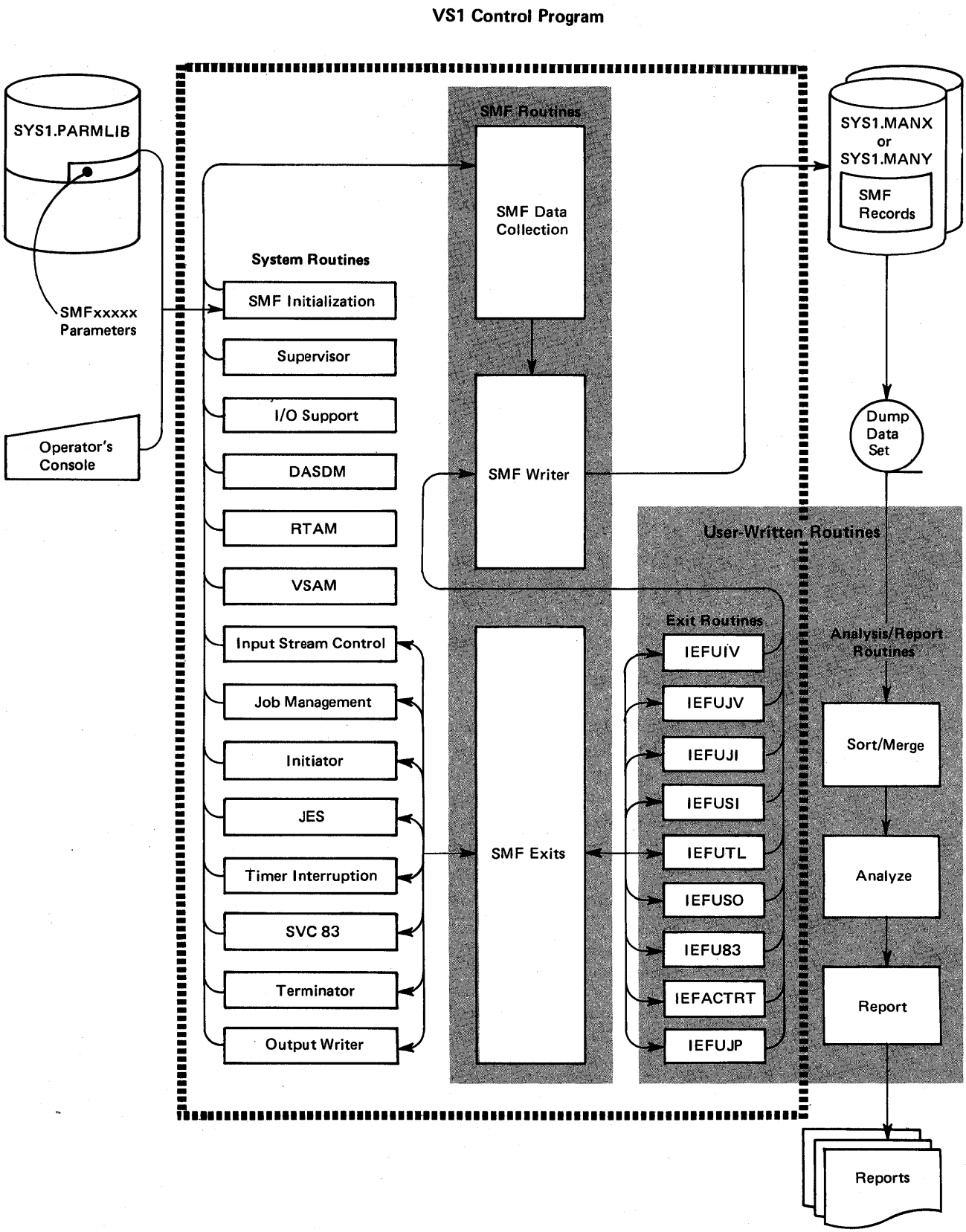


Figure 1. SMF in the Operating System

2. Checks for the existence and availability of the SMF data sets, SYS1.MANX and SYS1.MANY, if SMF recording was requested. If an error occurs, a message is issued indicating that recording on the SMF data sets is not allowed until the condition is corrected and the IPL procedure is repeated. If no errors occur, the initialization routine determines which SMF data set should receive records, as follows:

   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 filled data set receives the records. If both data sets contain data, the data set with the lesser amount of data receives the records.

3. Creates the IPL record (type 0), the I/O configuration record (type 8), the system statistics record (type 1), and the configuration record (type 22). These records are then written to the SMF data set.

After the SMF initialization completes without error, the initialization routine returns control to the master scheduler so that processing of input streams may begin.

Using User-Written Exit Routines

As the input stream control routine reads each job, it passes control to the SMF input stream validation exit, IEFUIV. A user-written IEFUIV exit routine can verify whether the job is to be accepted or rejected for processing.

Before each job control statement (or cataloged procedure) in the input stream is interpreted, job management passes control to the SMF job validation exit, IEFUJV. A user-written IEFUJV exit routine can verify any fields in the JCL statement, modify the JCL, or reject jobs that do not meet installation standards. After all the JCL is interpreted, the same user-written IEFUJV exit routine receives control for further validity checking.

Before the initiator selects a job, it passes control to the SMF job initiation exit, IEFUJI. In a user-written IEFUJI exit 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, SMF creates the job initiation record (type 20) and writes it to the SMF data set. Similar processing occurs for each job step: before the initiator selects a job step, it passes control to the SMF step initiation exit, IEFUSI.

When a job step terminates, either normally or abnormally, SMF produces a job step termination record (type 4) and, before the record is written, the terminator passes control to the SMF termination exit, IEFACTRT. A user-written IEFACTRT exit routine can modify the SMF type 4 record and create user records. It can also write these records to installation-defined or SMF-defined data sets. A user-written IEFACTRT exit routine also indicates whether the job is to continue and whether the type 4 record is to be written to the SMF data set. (Upon return, the type 4 record is written unless the user-written IEFACTRT exit routine suppresses it.)

At job termination, SMF produces a job termination record (type 5) and, before the record is written, the terminator again passes control to the SMF termination exit, IEFACTRT, for similar record modification. (Upon return, the type 5 record is written unless the user-written IEFACTRT exit routine suppresses it.)

After a job has terminated and all the SYSOUT output that pertains to the job (including the SMF output writer record (type 6)) is written, the SYSOUT writer passes control to the SMF job purge exit, IEFUJP. A user-written IEFUJP exit routine can write additional data statistics to the SMF data set.

Recording Events

For specific events, SMF collects and formats the following types of information:

- Configuration information is recorded when a DEFINE or VARY operator command is issued and after allocation recovery.
- Data set information is recorded when a data set opened by a user program is scratched, renamed, closed, or processed by end-of-volume (EOV).
- Direct access volume information is recorded for online, direct access devices at IPL; when a HALT EOD or SWITCH SMF operator command is issued; and when a volume is demounted.
- Subsystem information is recorded when START, MODIFY, and STOP subsystem operator commands are issued and when an invalid password is used.
- System information is recorded for ten-minute intervals of CPU-processing time, and when a HALT EOD or SWITCH SMF operator command is issued.

The following SMF exits also receive control whenever specific events occur:

- The time limit exit, IEFUTL, receives control from the timer interruption handler when the step CPU, job CPU, or continuous wait time limits are reached.
- The SYSOUT limit exit, IEFUSO, receives control from JES when the number of records written to an output data set exceeds the output limit for that data set.
- The SMF record exit, IEFU83, receives control from the SVC 83 routine when an SMF record is ready to be written to the SMF data set.

Dumping SMF Data Sets

SMF initially writes records to the SYS1.MANX data set until it becomes full. When SYS1.MANX is full, SMF opens and uses the SYS1.MANY data set to continue recording. Whenever SYS1.MANX or SYS1.MANY becomes full, SMF notifies the operator to use the SMF dump program to copy the full data set to a dump data set (usually on tape). For a detailed description and sample uses of the SMF dump program, see "Dumping the SMF Data Sets" in the chapter "System Requirements and Considerations".

The operator can use the HALT EOD and SWITCH SMF commands to cause SMF to switch recording from the active SMF data set to the inactive one. (These commands also cause SMF to record system statistics and to empty the SMF buffer into the active SMF data set before it is switched.) After the recording is switched, the operator can use the SMF dump program to copy the inactive data set to the dump data set. The dump data set can then serve as input to user-written analysis and report routines.

Using User-Written Analysis and Report Routines

In addition to user-written exit routines, an installation may supply various routines that can:

- Sort and merge the SMF records dumped by the operator from the SMF data set.
- Analyze the sorted SMF data to detect excessive system wait time, inefficient use of I/O devices, or any other statistics that might lead to improved system throughput.
- Report the data from selected SMF records and user-written analysis routines in an appropriate format.

System Requirements and Considerations

SMF is an optional part of the VS1 system that is selected at SYSGEN time. To record SMF data, an installation must define how it will use SMF (see the chapter "Defining the Use of SMF"), add user-written exit routines to the control program, allocate direct access space for the SMF data sets, and catalog the SMF data sets. This chapter describes these requirements as well as SMF storage requirements and performance considerations.

Including SMF in the Operating System

To add SMF to the VS1 operating system, specify the appropriate parameters in the SMF= operand of the SCHEDULR macro instruction at SYSGEN time. There are three SCHEDULR parameters for SMF:

- SMF=NOTSUPPLIED specifies that SMF processing is not to be included in the system. If you do not code either the SMF= parameter or the ESV=SMF parameter, NOTSUPPLIED is the default.
- SMF=BASIC specifies that the SMF records are not to be produced and exits IEFUSO, IEFACTRT, and IEFUJP are to be included in the system. (For further information on handling accounting information when SMF=BASIC is specified, see *OS/VS1 Planning and Use Guide.*)
- SMF=FULL specifies that the SMF records and exits IEFUIV, IEFUJV, IEFUJI, IEFUSI, IEFUTL, IEFUSO, IEFU83, IEFACTRT, and IEFUJP are to be included in the system. If you do not code the SMF= parameter and code the ESV=SMF parameter, FULL is the default.

Figure 2 summarizes the exits that are available when SMF=BASIC or SMF=FULL is specified. It also shows the times during normal job processing when the SMF exits receive control.

Exit Time	Exit Name	BASIC	FULL
JES Reader Time	IEFUIV	No	Yes
Interpretation	IEFUJV	No	Yes
Job Initiation	IEFUJI	No	Yes
Step Initiation	IEFUSI	No	Yes
Timer Expiration	IEFUTL	No	Yes
Output Limit Expiration	IEFUSO[1]	Yes	Yes
SMF Buffer Time	IEFU83	No	Yes
Allocation	IEFACTRT	Yes	No
Step Termination	IEFACTRT	Yes	Yes
Job Termination	IEFACTRT	Yes	Yes
Job Purge	IEFUJP	Yes	Yes

[1] If SMF=BASIC or SMF=FULL is specified in the SCHEDULR macro and a user-written IEFUSO exit routine is supplied, the IEFUSO routine receives control when the output limit is reached. To specify the output limit, use the JOUTLIM parameter of the JES macro instruction at SYSGEN time, or the OUTLIM parameter on the DD statement defining the output data set. Unless an IEFUSO exit routine is supplied, jobs will be canceled when the output limit is reached.

Figure 2. Exits Available When SMF=BASIC or SMF=FULL

Including User-Written Exit Routines In The Operating System

User-written exit routines are optional – SMF automatically provides dummy routines for all unused SMF exits. To include user-written exit routines in the operating system, add them to SYS1.AOS00 before generating the system. Otherwise, link-edit them into the required system load modules as shown in Figure 3.

Exit Routine	Descriptive Name	Required Load Module Assignment for Adding User-Written Exit Routine after SYSGEN
IEFUIV	Input Stream Validation	IEFJES in SYS1.LINKLIB
IEFUJV	Job Validation	IEFUJV in SYS1.LINKLIB
IEFUJI	Job Initiation	IEFSD162 in SYS1.LINKLIB
IEFUSI	Step Initiation	IEFSD162 in SYS1.LINKLIB
IEFUTL	Time Limit	IEANUC01 in SYS1.NUCLEUS
IEFUSO	SYSOUT Limit	IEFJES in SYS1.LINKLIB
IEFU83	SMF Record	IEANUC01 in SYS1.NUCLEUS
IEFACTRT	Termination	IEFSD161 in SYS1.LINKLIB[1] IEFW21SD in SYS1.LINKLIB[2]
IEFUJP	Job Purge	IEFJES in SYS1.LINKLIB

[1]You must link-edit IEFACTRT into this load module if either SMF=BASIC or SMF=FULL was specified at SYSGEN time.

[2]You must also link-edit IEFACTRT into this load module if SMF=BASIC was specified.

Figure 3. Including User-Written Exit Routines in the Operating System

Note: If a user-written exit routine is link-edited into a load module of an active system, a link-edit failure might make that load module inoperative. This is particularly important in the case of termination exit routines that direct output to SYSOUT because the cause of failure would also be lost.

When adding exit routines after SYSGEN, refer to your SYSGEN listing for exact load module names, aliases, link-edit control statements, and link-edit parameters. Specify link-edit parameters according to the characteristics of the user-written exit routines.

Figure 4 shows sample JCL for adding user-written exit routines to SYS1.LINKLIB after SYSGEN. Note that when the IEFUIV, IEFUSO, and IEFUJP exit routines become part of IEFJES, they do not become active until the next IPL of the system. This is also true for any user-written exit routine that is added to a resident load list.

Note: Refer to your SYSGEN listing for exact load module names, aliases, link-edit control statements, and link-edit parameters. Specify link-edit parameters according to the characteristics of the user-written exit routines.

```
//EXITLNK    JOB   MSGLEVEL=1
//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
  ORDER      IEFSD515
  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    *
```

(IEFUIV, IEFUSO, and IEFUJP object decks)

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

[1]You must also link-edit IEFACTRT in load module IEFW21SD if SMF=BASIC was specified at SYSGEN time. The required statements for this additional link-edit are as follows:

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

Figure 4. Sample JCL for Adding User-Written Exit Routines to SYS1.LINKLIB after SYSGEN

Figure 5 shows sample JCL for adding user-written IEFUTL and IEFU83 exit routines to SYS1.NUCLEUS after SYSGEN. Be aware that when the IEFUTL and IEFU83 exit routines become part of SYS1.NUCLEUS, they do not become active until the next IPL of the system.

Note: Refer to your SYSGEN listing for exact load module names, aliases, link-edit control statements, and link-edit parameters. Specify link-edit parameters according to the characteristics of the user-written exit routines.

```
//NUCLINK    JOB   MSGLEVEL=1
//             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    *
Link-edit, INSERT, and ORDER statements required[1]
(IEFUTL and IEFU83 object decks)
  INCLUDE    SYSLMOD(IEANUC01)
  NAME       IEANUC01(R)
/*
```

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

Figure 5. Sample JCL for Adding User-Written IEFUTL and IEFU83 Exit Routines to SYS1.NUCLEUS after SYSGEN

Storage Requirements

SMF requires additional real storage in the nucleus, the system queue area (SQA), and the partition queue area (PQA). SMF also requires additional auxiliary storage for the SMF data sets and for user-written exit routines. Figure 6 summarizes the system areas that require additional storage for SMF and the amount of storage that is needed in each area.

Item	Additional Storage Required for SMF	Area in System
SMF Data-Collection Routines	2700 bytes plus 148 bytes for each partition	Pageable Nucleus
SMF Writer Routine	1560 bytes	Pageable Nucleus
Timing Control Table (TCT) (one is created for each active job step)	If OPT=1, 116 bytes for each TCT If OPT=2, size of each 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.	Fixed Partition Queue Area (PQA)
System Management Control Area (SMCA)	180 bytes	Fixed System Queue Area (SQA)
Common Exit Parameter Area	124 bytes	Fixed PQA
SMF Buffer	400 to 8,192 bytes (See the following section, "SMF Buffer.")	Pageable SQA
SMF Data Sets	(See "SMF Data Sets" later in this chapter.)	DASD
User-Written Exit Routines	Size of IEFUIV, IEFUJV, IEFUJI, IEFUSI, IEFUSO, IEFACTRT, and IEFUJP user-written exit routines	SYS1.LINKLIB
	Size of IEFUTL and IEFU83 user-written exit routines	SYS1.NUCLEUS
Work Area for User-Written Exit Routines	Size of user-requested area (See "Obtaining Additional Work Areas" in the chapter "User-Written Exit Routines")	Fixed PQA and SQA

Figure 6. SMF Storage Requirements

SMF Buffer

If records are to be written to the SMF data set, you must supply a buffer size in the SMFxxxxx BUF parameter before or during IPL. (For detailed information on the BUF parameter, see "SMFxxxxx Parameters" in the chapter "Defining the Use of SMF".) The minimum SMF buffer size is 400 bytes; the maximum is 8,192 bytes.

The SMF buffer has two parts: while one half is being filled with SMF records, the other half is being written to the SMF data set. If a record size should exceed half of the buffer size, the record is segmented before it is written. In order to prevent segmenting records (and thereby improve performance), specify a buffer size that is at least twice the size of the largest record to be written to the SMF data set.

Figure 7 summarizes the sizes of the SMF records. When calculating the largest record size, be sure to include the block descriptor word (four bytes) and the record descriptor word (four bytes). Do not make the buffer size larger than necessary; if the system should fail, the records in the SMF buffer will be lost.

Record Type	Record Size (in bytes)
0	31
1	34
2	14
3	14
4	117 + 8 per DD statement + length of step accounting fields
5	117 + length of job accounting fields
6	99
7	24
8	16 + 4 per device
9	16 + 4 per device
10	40 + 4 per device
11	16 + 4 per device
12	34
13	16 + 22 per partition
14	264 + 24 per UCB + 28 per ISAM data set
15	264 + 24 per UCB + 28 per ISAM data set
17	88 + 8 per data set scratched
18	132 + 8 per data set renamed
19	64
20	61 + length of job accounting fields
21	44
22	18 + 42 at IPL + 6 per device varied
43	42 + 6 per line created
44	30 + 6 per line modified or 9 per user modified
45	31
47	152
48	68
49	152
50	58
51	18 + length of user/subsystem data
62	138 + 10 per volume
63	132 + length of catalog records
64	250 + 26 per extent
67	130 + length of catalog records
68	170
69	102
82	41 + length of variable relocate sections (5740-XY5)

Figure 7. SMF Record Sizes

Note: If you plan to reduce the SMF buffer size during consecutive IPL's, dump the SMF data sets first so that all records will be written with the same block size.

SMF Data Sets

You must allocate space for the SMF data sets, SYS1.MANX and SYS1.MANY, before IPL. Both data sets must be permanently resident on a direct access device. (Several factors, such as specific system configuration, amount of SMF data to be written, and report program requirements, will determine which device type is most efficient for a particular installation.) You must also catalog SYS1.MANX and SYS1.MANY. You should not date-protect these data sets because it is possible to lock up the system if an SMF record is being processed when a data set switch occurs.

Figure 8 illustrates sample DD statements for allocating the SMF data sets on direct access devices and cataloging them in the system catalog. Note that 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 8. Sample DD Statements for Allocating and Cataloging the SMF Data Sets

The amount of space to be allocated to SYS1.MANX and SYS1.MANY depends upon the amount of data generated by each job and how often the data sets are dumped. Figure 9 is an example of how to establish space requirements for an SMF data set, given certain assumptions. The values in Figure 9 are calculated for a four hour interval. Subsystem records are not shown in this example.

SYSTEM-RELATED RECORDS					
Event or Status	**Record Type**	**Assumption for This Example**	**Example No. of Bytes per Record**	**Example No. of Records**	**Example Total No. of Bytes**
IPL	0	Once per day	31	1	31
Devices Online at IPL	8	20 devices, including 6 DASD	96	1	96
	19		64	6	384
	22		60	1	60
Partition Defined	13	Once per day (4 partitions)	104	1	104
End of Day	12	Once per day	34	1	34
	19	6 DASD	64	6	384
Accumulated Wait Time	1	Once every 10 minutes	34	24	816
Devices Varied Online	9	Twice per hour	20	8	160
	22	1 MSS device per day	24	1	24
Device Recovered at Allocation	10	Once per hour	44	4	176
Devices Varied Offline	11	Twice per hour	20	8	160
	22	1 MSS device per day	24	1	24
TOTAL FOR SYSTEM-RELATED RECORDS				63	2453

Figure 9. Sample Data Set Space Requirements (Part 1 of 2)

JOB-RELATED RECORDS					
Event or Status	**Record Type**	**Assumption for This Example**	**Example No. of Bytes per Record**	**Example No. of Records**	**Example Total No. of Bytes**
Job Processing	5	Accounting data[1]	129	1	129
	20		73	1	73
	19	Demount 2 DASD volumes	64	2	128
	21	1 EOV or CLOSE	44	1	44
Step Processing	4	4 DD statements per step, accounting data[1], 3 steps per job	161	3	483
	14	1 EOV and 2 CLOSE per step, 3 steps per job	288	9	2592
	15	1 EOV and 2 CLOSE per step, 3 steps per job	288	9	2592
	17	Scratch non-VSAM data set in 1 step per 12 jobs[2]	96	1	96
	18	Rename non-VSAM data set in 1 step per 12 jobs[2]	140	1	140
	62	Open 2 components per step	168	2	336
	63	Define 1 non-VSAM entry per step	300	1	300
		Define VSAM index cluster in 1 step per 12 jobs[2]	280 cluster 550 data set 500 index	3	1330
		Alter each component of VSAM index cluster per step	1150 data set 930 index	2	2080
	64	1 EOV and 1 CLOSE per VSAM component	328	4	1312
	67	Delete non-VSAM entry per step	300	1	300
		Delete VSAM index cluster in 1 step per 24 jobs[2]	280 cluster 630 data set 530 index	3	1440
	68	ALTER rename a VSAM component in 1 step per 48 jobs[2]	170	1	170
	69	Define, extend or delete space on 1 volume in 1 step per 48 jobs[2]	102	1	102
SYSOUT Processing	6	2 output writers per step, 3 steps per job[3]	99	6	594
TOTAL FOR JOB-RELATED RECORDS PER JOB				52	14,241
TOTAL FOR 12 JOBS PER HOUR FOR 4 HOURS				2026	529,502
TOTAL FOR SYSTEM-RELATED AND JOB-RELATED RECORDS				2089	532,955
Record Descriptor Word	ALL	N/A	4	2089	8356
Block Descriptor Word	N/A	6 records per block	4	348 blocks	1392
TOTAL NUMBER OF BYTES FOR THIS EXAMPLE					531,911

[1] Accounting data consists of two 5-byte items.

[2] The number of bytes for these events is calculated only once for the four-hour interval.

[3] A type 6 record is issued for each data set destined for an IBM 3800.

Figure 9. Sample Data Set Space Requirements (Part 2 of 2)

Dumping the SMF Data Sets

When either SYS1.MANX or SYS1.MANY becomes full, the SMF writer routine issues a message to the console requesting a dump of the SMF data set.

The operator should use the SMF dump program, IFASMFDP, to transfer a full SMF data set to another data set (usually on tape), and to reset the status of the dumped data set to empty so that it can be used again for recording data.

Figure 10 shows sample JCL for executing the SMF dump program. The output is a non-temporary data set on a standard-labeled tape. The operator should record the volume serial number of the output data set so that other jobs can reference it.

```
//DUMPX     JOB  MSGLEVEL=1
//STEP1     EXEC PGM=IFASMFDP
//DUMPIN    DD   DSN=SYS1.MAN{X},DISP=OLD
                              {Y}
//DUMPOUT   DD   DSN=SMFDATA,UNIT=tapeaddr,
//                DISP=(NEW,KEEP),LABEL=(,SL),VOL=SER=serial
//SYSPRINT  DD   SYSOUT=A
```

Note: Do not specify the DCB= keyword when executing the SMF dump program. Although RECFM=VBS for both the SMF data set and the output data set, the SMF dump program alternately changes from RECFM=VBS to RECFM=VB in order to dump segmented records by individual segments. (See Figure 31 for sample JCL describing the output data set SMFDATA.)

Figure 10. Sample JCL for Executing the SMF Dump Program

The SMF dump program uses the basic sequential access method (BSAM) to physically copy the input data set, DUMPIN, to the output data set, DUMPOUT. During the copy process, the dump program creates two SMF records and writes them to the output data set: a dump header (record type 2) at the beginning of the data set and a dump trailer (record type 3) at the end of the data set.

The SMF dump program also writes messages, as required, to SYSOUT:

- The operator must not dump a data set that is being filled. If the operator attempts to dump the active SMF data set, IFASMFDP returns a code of X'10' in register 15 and writes a message to the console notifying the operator of the error. In response, the operator must cancel the dump program and then invoke it again to dump the correct data set.
- If IFASMFDP is unable to open either the input or output data set, it writes an error message indicating which data set was not successfully opened.
- IFASMFDP determines whether the blocksize of the output data set is smaller than the blocksize of the input data set. If it is, IFASMFDP writes an error message to the console.
- If both SMF data sets and the SMF buffer become full, SMF will be in a data lost condition (unable to record) until dumping takes place. When this condition occurs, the lost records are tracked in SMF record type 7 and the operator receives a message stating that he or she should dump either data set.

One method of executing the SMF dump program is to enter jobs that specify the program into the system, and hold them on the job queue until a dump is required. Another method is to start a reader to an input stream containing the JCL for the SMF dump program. Figure 11 illustrates two sample procedures (DUMPX and DUMPY) for dumping the SMF data sets to a standard-labeled tape (VOL=SER=SMFTAP) by means of the operator START command. In both procedures, the default tape specified on the PROC statement is 192. Figure 11 also illustrates sample JCL for adding these procedures to SYS1.PROCLIB.

```
//UPDATE    JOB   MSGLEVEL=1
//UPDATE    EXEC  PGM=IEBUPDTE,PARM=NEW
//SYSUT1    DD    DSN=SYS1.PROCLIB,DISP=SHR
//SYSUT2    DD    DSN=SYS1.PROCLIB,DISP=SHR
//SYSPRINT  DD    SYSOUT=A
//SYSIN     DD    DATA
./          ADD   NAME=DUMPX,LIST=ALL
//DUMPX     PROC  TAPE=192
//SMFDMP    EXEC  PGM=IFASMFDP
//DUMPIN    DD    DSNAME=SYS1.MANX,DISP=OLD
//DUMPOUT   DD    DSNAME=SMFDATA,UNIT=&TAPE,DISP=(MOD,KEEP),
//               LABEL=(,SL),VOL=SER=SMFTAP
//SYSPRINT  DD    SYSOUT=A
./          ADD   NAME=DUMPY,LIST=ALL
//DUMPY     PROC  TAPE=192
//SMFDMP    EXEC  PGM=IFASMFDP
//DUMPIN    DD    DSNAME=SYS1.MANY,DISP=OLD
//DUMPOUT   DD    DSNAME=SMFDATA,UNIT=&TAPE,DISP=(MOD,KEEP),
//               LABEL=(,SL),VOL=SER=SMFTAP
//SYSPRINT  DD    SYSOUT=A
./ENDUP
/*
```

Figure 11. Sample Procedures for Dumping the SMF Data Sets

Switching the SMF Data Sets

When an SMF data set becomes full, SMF writes a message to the console and automatically switches recording from the active SMF data set to the inactive SMF data set. To dump an SMF data set *before* it becomes full, however, the operator can use either the HALT EOD or SWITCH SMF command. (These operator commands are fully described in *Operator's Library: OS/VS1 Reference.*)

When the operator issues either the HALT EOD or the SWITCH SMF command, the following actions occur:

- A type 19 record is created for each online direct access device if DSV=1 or DSV=3 was specified.
- A type 12 record is created. This record contains the system wait time and paging statistics accumulated from the expiration of the ten-minute interval reflected in the last type 1 record to the time of the HALT or SWITCH command. (The next type 1 record will then contain the system wait time and paging statistics accumulated from the HALT or SWITCH command to the expiration of the next ten-minute interval.)
- The SMF data sets are switched so that the operator can dump the previously active data set.

  Note: When switching the SMF data sets, the inactive data set cannot become active unless it is empty. Therefore, the operator must dump the inactive data set (or initialize the system) before issuing the HALT or SWITCH command.

Performance Considerations

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

- SMF options selected through the SMFxxxxx parameters. (For a description of these parameters, see the chapter "Defining the Use of SMF".)
- SMF buffer size. (If the buffer size is too small, SMF segments records before writing them.)
- SMF data set size, device type, and dumping requirements.
- Execution time of user-written exit routines.
- System configuration, especially the type and degree of multiprogramming.
- Processing characteristics, such as the number of jobs (the number of records generated by SMF is dependent on the number of jobs), the contention for SMF resources, and user data set requirements.

Defining the Use of SMF

When SMF=FULL is specified at SYSGEN time, an installation must define how it will be using SMF through a SYS1.PARMLIB member SMFxxxxx:

- Required SMFxxxxx parameters specify the job wait time limit and the system on which SMF is active.
- Optional SMFxxxxx parameters allow an installation to select record types, to specify the physical length of SMF records, to permit operator modification of SMFxxxxx parameters, and to specify whether SMF exits are to be taken.

SMFxxxxx parameters are specified (1) before the first IPL of a newly generated system by adding SMFxxxxx as a member in SYS1.PARMLIB, (2) at each initialization of SMF by entering SMFxxxxx parameters at the console, or (3) by the setup and subsequent use of automated system initialization (ASI). (See *OS/VS1 Planning and Use Guide* for specifying SMFxxxxx parameters when using ASI.)

SMFxxxxx Parameters

The SMFxxxxx parameters are described in Figure 12. Note that the JWT and SID parameters are required for SMF. The BUF parameter is also required unless MAN=NONE is specified.

Note: If a job is recovered in a warm start, the values for the DSV, EXT, OPT, and REC SMFxxxxx parameters will be the values in effect when the job was read. Modifications for these parameters are ignored during a warm start. The value for the SID SMFxxxxx parameter will be the value in effect when the job was read for record types 4 and 5 only.

Parameter	Meaning and Use	Value Range	Default Value
BUF={*nnn* / *nnnn*}	This parameter is required unless MAN=NONE is specified. It indicates the SMF buffer size. (SMF buffer size requirements are discussed in the chapter "System Requirements and Considerations.") *Notes:* 1. Specify a BUF value that is a multiple of 8 bytes (double word); otherwise, SMF rounds the value to the next *lower* multiple of 8 bytes. 2. Before reducing the buffer size specified in the previous IPL, dump the SMF data set; otherwise the data sets cannot be dumped.	400 to 8,192	None (Operator is prompted if BUF value is needed.)
DSV={**0** / 1 / 2 / 3}	This optional parameter specifies whether data set information and/or direct access volume information is to be collected by SMF. 0 suppresses both data set information and direct access volume information. 1 generates direct access volume information and suppresses data set information. 2 generates data set information and suppresses direct access volume information. 3 generates both data set information and direct access volume information. *Note:* If OPT=1 and either DSV=2 or DSV=3 is specified, SMF converts OPT=1 into OPT=2 and issues a warning message.	0, 1, 2 or 3	0
EXT={**YES** / NO}	This optional parameter specifies whether SMF exits (except IEFUSO, which is always taken when the output limit is reached) are to be taken. YES specifies exits are to be taken. NO specifies exits are not to be taken. *Note:* If EXT=YES is specified, the exits taken will depend upon the data-collection parameter, OPT. If OPT=2, all exits defined for the system are taken. If OPT=1, the step initiation exit, IEFUSI, and the step termination exit, IEFACTRT, are not taken.	Not applicable	YES
JWT=*nnn*	This is a required parameter that initially specifies the number of minutes a job is allowed to wait continuously. When the specified time limit has expired, the time limit exit, IEFUTL, is entered (if exits are to be taken). The limit value can be changed by IEFUTL.	1-999	10
MAN={NONE / USER / **ALL**}	This optional parameter specifies whether user and/or SMF records are to be written to the SMF data sets. You must specify MAN=ALL or MAN=USER if records are to be written to the SMF data sets. Unless MAN=NONE is specified, the BUF parameter is also required. NONE does not write user or SMF records to the SMF data sets, regardless of values specified for DSV, OPT, and REC parameters. USER writes only user records (types 128 through 255) to the SMF data sets. ALL writes both SMF and user records to the SMF data sets. All SMF records are created unless suppressed by the DSV, OPT, or REC parameters. All SMF records that are created are written unless suppressed by user-written exit routines. *Note:* Even if MAN=USER is specified, the SMF record types 2, 3, and 7 will still be written.	Not applicable	ALL

Figure 12. SMFxxxxx Parameters (Part 1 of 2)

Parameter	Meaning and Use	Value Range	Default Value
OPI= {YES / NO}	This optional parameter specifies whether the SMFxxxxx parameters are to be presented on the console during initialization for the operator's inspection and/or modification. (This parameter is ignored if it is entered from the console.) YES allows the operator to modify the SMFxxxxx parameters. NO does not allow the operator to modify the SMFxxxxx parameters.	Not applicable	NO
OPT= {1 / 2}	This optional parameter specifies whether system and job information, as opposed to system, job, and job step information, is to be recorded. 1 generates only system and job-related information (that is, record type 4 is suppressed and the step initiation exit, IEFUSI, and step termination exit, IEFACTRT, are not taken). 2 generates system, job, and job step information. *Note:* If OPT=1 and DSV=2 or DSV=3 is specified, SMF converts OPT=1 into OPT=2 and issues a warning message.	1 or 2	2
REC= {0 / 2}	This optional parameter specifies whether record type 17 will be written for temporary data sets. This parameter is not effective unless either DSV=2 or DSV=3 is specified. 0 writes record type 17 for non-temporary data sets only. 2 writes record type 17 for both temporary and non-temporary data sets.	0 or 2	0
SID=*xxxx*	This is a required parameter that specifies the system and model on which SMF is active, provided the installation modifies the default value.	Four alphameric and/or special characters	155A

Figure 12. SMFxxxxx Parameters (Part 2 of 2)

Selecting SMF Records Using SMFxxxxx Parameters

Four SMFxxxxx parameters control the type of records to be written to the SMF data set: DSV, MAN, OPT, and REC. Figure 13 summarizes the use of these four parameters in selecting SMF records.

Parameter	Value	Meaning	Effect on SMF Records
DSV	0	No information for data sets or direct access volumes.	Suppresses record types 14, 15, 17, 18, 19, 62, 63, 64, 67, 68, and 69.
	1	Direct access volume information.[1]	Generates record types 19 and 69; suppresses record types 14, 15, 17, 18, 62, 63, 64, 67, and 68.
	2	Data set information.[2]	Generates record types 14, 15, 17, 18, 62, 63, 64, 67, and 68; suppresses record types 19 and 69.
	3	Data set and direct access information.[1,2]	Generates record types 14, 15, 17, 18, 19, 62, 63, 64, 67, 68, and 69.
MAN	NONE	No records.	The SMF data set is not used.
	USER	User records.	User record types 128 through 255 (and SMF record types 2, 3, and 7) can be written to the SMF data set.
	ALL	Both user and SMF records.	Record types 0 through 255 can be written to the SMF data set.
OPT	1	System and job information.	Generates record types 0-3, 5-13, 20, 22, 43-45, 47-49, and 50; suppresses record type 4.
	2	System, job, and job step information.	Generates record type 4 and all the above record types.
REC	0	No information for temporary data sets.	Generates record type 17 for non-temporary data sets only.
	2	Temporary and non-temporary data set information.	Generates record type 17 for both temporary and non-temporary data sets.

[1] Record type 21 is written to the SMF data set only when ESV=SMF is specified in the SCHEDULR macro instruction at SYSGEN time.

[2] If OPT=1 and either DSV=2 or DSV=3 is specified, SMF converts OPT=1 to OPT=2 and issues a warning message.

Figure 13. Summary of the Use of SMFxxxxx Parameters to Select SMF Records

User-written exit routines IEFU83 (SMF writer) and IEFACTRT (termination) can also control which records are to be written to the SMF data set. After inspecting an SMF record, these routines return a code to the system indicating whether the record is to be written to the SMF data set.

Entering SMFxxxxx in SYS1.PARMLIB

When you have determined which SMF parameters to use, place them in an SMFxxxxx member of SYS1.PARMLIB. The five alphameric characters represented by xxxxx are appended to SMF to identify your SMFxxxxx member. If you do not specify an SMFxxxxx member name, the default member SMFPRM00 is used. The parameters in SMFPRM00 are:

```
OPT=2,EXT=YES,SID=155A,BUF=2000,JWT=10,OPI=YES,MAN=ALL
```

For entering SMFxxxxx in SYS1.PARMLIB when using automated system initialization (ASI), see the AUTO= keyword in *OS/VS1 Planning and Use Guide.*

The SMFxxxxx parameters can be in any order; however, note the following coding restrictions:

- Code each series of parameters in logical records no more than 80 bytes long.
- Use columns 1-71; columns 72-80 are ignored.
- Enter each parameter in the format: **keyword=*value.***
- Do not use embedded blanks.
- Separate consecutive parameters by commas.
- Do not divide a parameter between consecutive records.
- Indicate continuation by placing a comma after the last entry on a record, followed by a blank before column 72.

To add the SMFxxxxx parameters as a member of SYS1.PARMLIB, use the IEBUPDTE utility program. Figure 14 illustrates sample JCL for using IEBUPDTE to enter SMFPRM01 into SYS1.PARMLIB. To change the default member, SMFPRM00, or the installation-defined SMFxxxxx member, replace them with a new version by again executing IEBUPDTE. For information on the IEBUPDTE program, see *OS/VS1 Utilities.*

```
//ENTER    JOB   MSGLEVEL=1
//         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 member)
./         ENDUP
/*
```

[1]To access SMFPRM00 on the distribution package before SYSGEN, use the SYS1.APARMLIB data set.

Figure 14. Sample JCL for Entering SMFPRM01 into SYS1.PARMLIB Using IEBUPDTE

If OPI=YES was specified, the operator can modify the SMFxxxxx parameter values from the console during system initialization. If parameter errors occur, the operator will be prompted for correct parameters regardless of the value specified for OPI.

SMF has exits in the control program that can link to user-written routines for monitoring jobs or job steps at various points in their processing cycles. An installation can use any or all of the SMF exits by including user-written exit routines in SYS1.AOS00 before SYSGEN or in SYS1.LINKLIB and SYS1.NUCLEUS after SYSGEN. SMF automatically provides dummy routines for all unused exits. (For detailed information on adding user-written exit routines, see "Including Exit Routines in the Operating System" in the chapter "System Requirements and Considerations".)

Planning Exit Routines

This section introduces the SMF exits. It briefly describes when each exit is called, the parameters passed to it, and the return codes required from it.

Exit Routines and Their Characteristics

SMF supplies nine exits, which can link to user-written exit routines, as follows:

- The input stream validation exit (IEFUIV) receives control from the input stream control routine for each job in the input stream. A return code from this exit indicates whether a job is to be accepted for processing.
- The job validation exit (IEFUJV) receives control from job management before each job control statement (or cataloged procedure) in the input stream is interpreted. This exit also receives control again after all the JCL is interpreted. It is not taken for JCL comment statements. A return code from this exit indicates whether processing of the job is to be continued.
- The job initiation exit (IEFUJI) receives control from the initiator before a job on the input queue is selected for initiation. A return code from this exit indicates whether the job is to be started or canceled.
- The step initiation exit (IEFUSI) receives control from the initiator before each job step is started (prior to allocation). A return code from this exit indicates whether the job step is to be started or the job is to be canceled.
- The time limit exit (IEFUTL) receives control from the timer interruption handler 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 the default from the reader procedure); or the continuous wait time limit for the job (from the SMFxxxxx JWT parameter). A return code from this exit indicates whether the job step is to be terminated or processing is to be continued with a new time limit.
- The SYSOUT limit exit (IEFUSO) receives control from JES when the number of records written to a SYSOUT data set exceeds the output limit for that data set. A return code from this exit indicates whether the job is to be terminated or processing is to be continued using a new limit.

- The SMF record exit (IEFU83) receives control from the SVC 83 routine before each record is written to the SMF data set. A return code from this exit indicates whether the current SMF record is to be suppressed.
- The termination exit (IEFACTRT) receives control from the terminator on the normal or abnormal termination of each job step and job. When SMF=BASIC is specified, this exit also receives control at the beginning of allocation. A return code from this exit indicates whether the job is to be continued (for job steps only) or terminated, and whether the SMF termination records are to be written to the SMF data set (for SMF=FULL only).
- The job purge exit (IEFUJP) receives control from the SYSOUT writer routine when a job is ready to be purged from the system (that is, after the job has terminated and all SYSOUT output that pertains to the job has been written). This exit does not return a code to the control program.

Figures 15 and 16 summarize when each SMF exit is called, the modules that call each user-written exit routine, the information passed to each exit (in addition to the common exit parameters, described in Figure 18), and the return from each exit to the control program. Figure 15 summarizes this information for the exits available when SMF=BASIC is specified; Figure 16 summarizes it for the exits available when SMF=FULL is specified.

Module:	At:	Interfaces With:	For User Exit:	Parameters Passed:	Type of Return:
IEFSMPUT	Output Limit Expiration	IEFSMFSO	IEFUSO	DCB address.	Continue with new limit or cancel.
IEFSD21Q	Allocation	IEFACTLK	IEFACTRT	Job step name, programmer name, job CPU time, job accounting fields, step CPU time, step accounting fields, completion code.	Continue or cancel.
IEFSD31Q	Job Termination	IEFACTLK	IEFACTRT		
IEFZGST2	Step Termination	IEFACTLK	IEFACTRT		
IEFOSC05	Job Purge	directly	IEFUJP	Programmer name, job CPU time, job accounting fields.	None.

Figure 15. Exit Routine Characteristics When SMF=BASIC

Module:	At:	Interfaces With:	For User Exit:	Parameters Passed:	Type of Return:
IEFVMB	JES Reader	directly	IEFUIV	JOB statement image.	Continue or cancel.
IEFVHH	Interpretation	directly	IEFUJV	JCL statement image, type of JCL statement.	Continue or cancel.
IEFVHEB	Interpretation	directly	IEFUJV		
IEFSD162	Job Initiation	IEFSMFIE	IEFUJI	Programmer name, job priority, job accounting fields.	Continue or cancel.
IEFSD162	Step Initiation	IEFSMFIE	IEFUSI	Job step name, program name, step accounting fields.	Continue or cancel.
IEATLEXT	Timer Expiration	directly	IEFUTL	None.	Continue with new time limit or cancel.
IEFSMPUT	Output Limit Expiration	IEFSMFSO	IEFUSO	DCB address.	Continue with new limit or cancel.
IEESMF8C	SMF Buffer Time	directly	IEFU83	SMF record to be written.	Write or do not write record to SMF data set.
IEFSD31Q	Job Termination	IEFSMFLK	IEFACTRT	Job step name, programmer name, job CPU time, job accounting fields, step CPU time, step accounting fields, completion code, SMF termination record.	Continue or cancel; write or do not write record to SMF data set.
IEFZGST2	Step Termination	IEFSMFLK	IEFACTRT		
IFOSC05	Job Purge	directly	IEFUJP	Programmer name, job CPU time, job accounting fields.	None.

Figure 16. Exit Routine Characteristics When SMF=FULL

Sample Exit Routines in SYS1.ASAMPLIB

Sample assembler language exit routines for some SMF exits are provided in the member SMFEXITS of SYS1.ASAMPLIB. Figure 17 shows sample JCL for obtaining a listing of these sample routines.

```
//PRINT    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      SYSOUT=A
//SYSIN    DD      *
         PRINT   TYPORG=PO,MAXNAME=1,MAXFLDS=1
         MEMBER  NAME=SMFEXITS
         RECORD  FIELD=(80)
/*
```

[1]The volume and unit parameters depend upon your installation's request.

Figure 17. Sample JCL for Obtaining a Listing of Sample Exit Routines

A summary of the sample exit routines in SYS1.ASAMPLIB follows:

- IEFUIV – No sample provided.
- IEFUJV – Sample routine checks the validity of a continued JOB statement and of values supplied for REGION, PRTY, TIME, and accounting parameters in the JOB statement. The routine uses characters from the account number to index a table that contains allowable values for these parameters. If any value is invalid, the sample IEFUJV routine terminates the job.

- IEFUJI – Sample routine determines how long a job has been on the input job queue before it is initiated. It then writes this value and the job priority to the SMF data set as a user record.
- IEFUSI – No sample provided.
- IEFUTL – Sample routine terminates a job if either the job CPU time limit or the job step CPU time limit has been exceeded. If the continuous wait time limit for the job has been exceeded, the routine extends the limit twice; on the third entry for exceeding the continuous wait time limit, it cancels the job. Each time the routine is entered for exceeding the continuous wait time limit, it writes a record describing the action taken to the SMF data set.
- IEFUSO – No sample provided.
- IEFU83 – Sample routine determines whether the record to be written is an IPL record. If it is, the routine 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 the TESTEXIT procedure. Remove this macro definition 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.
- IEFACTRT – Sample routine changes the SMF job termination and job step termination records (types 4 and 5) to user records, and attempts to write them to the SMF data set. If the data set is full, the routine writes a message to the console indicating that SMF records are being lost. At job termination, the routine writes a record containing the job name, programmer's name, and account number to the SYSOUT device.
- IEFUJP – No sample provided.

Writing Exit Routines

This section describes:

- Exit routine restrictions and facilities, including information on communication among user-written exit routines; the common parameters passed to all routines; the SMFWTM macro instruction, which is used to write records to the SMF data set; and the IFASMFR macro instruction, which is used to symbolically address fields in the SMF records.
- Exit routines, including a full description of the parameters passed to each routine and the required return codes.

Exit Routine Restrictions

This section describes only the restrictions and conditions common to most user-written exit routines. For those restrictions and conditions that involve only one or two exit routines, see the specific routine(s) later in this chapter.

Before writing an exit routine, note the following user-written exit routine restrictions and conditions:

- All user-written exit routines should be reenterable because the link-edit attributes of the load modules are subject to change from release to release. The exit routines IEFUTL and IEFU83, and any other routines that become resident, must be reenterable.
- All user-written exit routines receive control via a BALR instruction. The routines must save registers when they receive control and restore registers when they return control 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.
- All user-written exit routines receive control in storage protection key 0.
- All user-written exit routines receive control with the system enabled for interrupts.
- No user-written exit routines except IEFACTRT can write to the system output message data set.
- Do not allocate installation-defined data sets to SYSOUT. To write to installation-defined data sets, define the data sets as follows:
  - A data set used by the IEFUIV exit routine requires a DD statement in the reader cataloged procedure.
  - A data set used by the exit routines IEFUJV, IEFUJI, IEFUSI, and IEFACTRT requires a DD statement in the initiator cataloged procedure. However, a job scheduled by the START command does not run in the initiator environment; it runs as if an initiator had not been started in the partition. Such a job does not have access to the initiator TIOT and thus cannot access user-defined data sets for that job's accounting.
  - A data set used by the IEFUJP exit routine requires a DD statement in the writer cataloged procedure.

  User-written exit routines IEFUTL, IEFUSO, and IEFU83 cannot write to installation-defined data sets.

  VSAM data sets are not supported for SMF exits from the system reader, writer, RTAM, or VTAM.
- If you plan to use the TESTEXIT procedure for testing user-written exit routines, see "TESTEXIT Exit Routine Requirements" in the chapter "Testing Exit Routines."
- If you plan to use the IBM OS/VS Sort/Merge Program Product, do not create user records smaller than the minimum length required by the program. (When using tape work devices, the minimum length this program can sort is 18 bytes. Otherwise, the minimum is one byte.)

Exit Routine Facilities

This section describes the facilities common to most user-written exit routines. For more information on the facilities that are particular to a specific exit routine, see that routine later in this chapter.

Common Exit Parameter Area

When a user-written exit routine receives control, register 1 points to a list of four-byte addresses. The first entry in this list is common to all exit routines except IEFU83. The first entry points to a parameter area that is 124 bytes long.

Figure 18 describes the format of the common exit parameter area. Note that the fields in this parameter area are filled chronologically; therefore, not all fields are meaningful for all user-written exit routines.

Displacement from Pointer	Field Size	Data Format	Description
0	8	EBCDIC	Job name
8	4	binary	Time, in hundredths of a second, reader recognized the JOB card for this job
12	4	packed	Date reader recognized the JOB card for the job, in the form 00YYDDDF where F is the sign
16	4	EBCDIC	System identification (taken from SID parameter[1])
20	8	EBCDIC	User identification. Refer to description of record type 6 for special usage. This field is initialized to EBCDIC blanks when each job is read[2]
28	1	binary	Number of the step being processed
29	1	binary	Indicator of the SMF options selected by the user.
			Bit *Option Selected When Set*
			0 System and job accounting (OPT=1)
			1 System, job, and step accounting (OPT=2)
			2 User exits will be taken (EXT=YES)
			3 Data set accounting (DSV=2 or 3)
			4 Volume accounting (DSV=1 or 3)
			5 Reserved
			6 Type 17 records will be written for temporary data sets (REC=2)
			7 Reserved
30	2	binary	Reserved
32	4	binary	User-communication field. This field is intended for communication among user exit routines. It is initialized to zeros when a reader is started [2]
36	4	binary	Real time in reader, in hundredths of a second [5]
40	4	binary	Number of cards read [5]
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[3,4]
56	4	binary	Real time to punch, in hundredths of a second[3]
60	4	binary	Number of SYSOUT lines punched[3]
64	4	binary	Real time for tape, in hundredths of a second[3]
68	4	binary	Number of SYSOUT lines written to tape[3]
72	44	binary	Reserved[3]
116	8	EBCDIC	RTAM user identification

[1]Modifications for the SID parameter are ignored during a warm start IPL for SMF record types 4 and 5.

[2]These fields are provided for user modification.

[3]These fields are filled in at writer job purge time, and are valid only to the IEFUJP exit routine.

[4]This field includes job-related lines if the JOBLOG=YES parameter is specified in the JES macro instruction at SYSGEN.

[5]This does not include time or records read from associated data sets.

Figure 18. Common Exit Parameter Area

Communicating between Exit Routines

User-written exit routines can communicate with each other in two ways: by using the user-communication field or by using the user-identification field. Both of these fields are contained in the common exit parameter area, which is passed to all user-written exit routines except IEFU83.

All exit routines (except IEFU83) that are executing within the same job can communicate via the user-communication field (displacement 32 in Figure 18) and the user-identification field (displacement 20 in Figure 18). The IEFUIV exit routine is the only routine that can communicate between different jobs; it must use the user-communication field to do so.

Note: The user-communication field is initialized to zeros each time a reader is started; the user-identification field is initialized to EBCDIC blanks each time a job is read. Neither of these fields is maintained if the system is restarted.

Obtaining Additional Work Areas

Any user-written exit routine can obtain an additional work area by issuing a GETMAIN macro instruction that specifies an appropriate subpool. Figure 19 shows the subpools (and their characteristics) that are required to obtain additional work areas. Be sure to consider the storage required by an additional work area when estimating the sizes of the system queue area and/or the partition queue area.

Subpool Number	Area in Storage	Storage Attributes
241	System Queue Area	Explicitly freed, by issuing a FREEMAIN macro instruction.
253	Partition Queue Area	Automatically freed at end of task.
254	Partition Queue Area	Automatically freed at end of step.
255	Partition Queue Area	Explicitly freed, by issuing a FREEMAIN macro instruction. The area must be freed while the job is running in the same partition from which the area was gotten.

Figure 19. Required Subpools for Obtaining Additional Work Areas

If desired, you can place the address of the work area in the user-communication field. However, be aware that the address will be destroyed if the field is re-initialized to zero because a new reader was started.

Using the SMFWTM Macro to Write Records

Use the SMFWTM macro instruction in any user-written exit routine except IEFU83 to write records to the SMF data set. The SMFWTM macro is written in assembler language and is supplied on SYS1.MACLIB. Its format 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 to be written. The value for (r) can be either the absolute register number or a symbol for the register. In either case, you must code the parentheses, for example, (2) or (REG2).

Record types 128 through 255 are available for user-written records. When using the SMFWTM macro instruction to write user records, you must include the standard SMF record header and a record descriptor word (RDW) for each record. (For a description of the standard SMF record header, see the chapter "SMF Records" in this book. For a description of the RDW, see *OS/VS Data Management Services Guide.*)

Record types 0 through 127 are SMF-formatted records. For all SMF-formatted records except types 4 and 5, you must supply only the record type field in the standard SMF record header (offset 1). The SMFWTM macro supplies the remaining header information.

The SMFWTM macro instruction returns a code in register 15 that indicates the record's status, as follows:

- 0 indicates the record was written without error.
- 4 indicates the record was truncated because it would not completely fit in an empty SMF data set.
- 8 indicates the record was not written because the length specified in the RDW was less than five bytes.
- 12 indicates the record was not written because the user-written exit routine was not authorized to write to the SMF data set.
- 16 indicates the record was not written because (1) the MAN=NONE parameter suppressed the writing of records to the SMF data set, or (2) the writing of records was allowed but the SMF data set was full. (If the SMF data set is full, the operator must dump it before additional SMF records can be written. See "Dumping the SMF Data Sets" in the chapter "System Requirements and Considerations" for the procedure for executing the SMF dump program.)
- 20 indicates the record was not written because the user-written IEFU83 exit routine suppressed the record.

The exit routine which issues the SMFWTM macro instruction must be operating in key 0 when it issues the macro instruction. (Refer to the description of the MODESET macro instruction in the *OS/VS1 Planning and Use Guide.*)

Using the IFASMFR Macro to Address SMF Record Fields

Use the IFASMFR macro instruction in user-written exit routines (or in any problem program application) to symbolically address SMF record fields. The macro is written in assembler language and is supplied on SYS1.AMODGEN. (Depending on your installation's requirements, you may want to copy the IFASMFR macro from SYS1.AMODGEN into your own macro library or SYS1.MACLIB.)

The format of the IFASMFR macro is:

```
[label] IFASMFR n
```

where:

n
: is the record type to be defined. You must specify at least one record type with the macro; if more than one record type is specified, you must enclose the record types in parentheses and separate them by commas. The values for n can be: 0-15, 17-22, 43-45, 47-51, 62-64, 67, 68 and 69.

Notes:

1. The "Name" column in all the SMF record formats (see the chapter "SMF Records") contains the symbolic addresses defined by the IFASMFR macro instruction.
2. Do not specify both record type 14 and record type 15 in the same program. Because these records are identical, whenever record type 15 is specified in the IFASMFR macro, record type 14 is defined.
3. If you do not want the IFASMFR macro to use part of the problem program's storage, then supply a CSECT or DSECT statement ahead of the macro instruction.

IEFUIV – Input Stream Validation Exit

IEFUIV receives control from the input stream validation control routine every time a new job is encountered in the input stream. A user-written IEFUIV exit routine can verify that the JOB statement is correct and acceptable for processing. It can also modify any of the operand fields in the JOB statement.

Notes:

1. To record the jobs that were not accepted by IEFUIV on the SMF data set, use the IEFUIV exit routine. To use installation-defined data sets with this exit routine, you must define them with a DD statement in the reader cataloged procedure.
2. A user-written IEFUIV exit routine can use the user-communication field to communicate between different jobs. Data placed in this field by IEFUIV will become part of each job, will be accessible by all exit routines entered during each job (except IEFU83), and will be the same the next time IEFUIV is entered unless a new reader is started. (When a new reader is started, the user-communication field is initialized to zeros.)
3. When modifying a JOB statement, do not include any additional continuation cards. Also, when adding or modifying operand fields, begin the first operand field in the same place that it was before any additions or modifications were made.

Parameters

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

1. The address of the common exit parameter area. (See Figure 18.)
2. The address of an 80-character JOB statement image.

Return Codes

Before the IEFUIV exit routine returns to the control program, it must place a return code in register 15, as follows:

0 indicates the job is acceptable for processing.

4 indicates the job is not acceptable for processing.

IEFUJV – Job Validation Exit

IEFUJV receives control from job management before each job control statement (or cataloged procedure) in the input stream is interpreted. This exit routine also receives control again after all of the JCL is interpreted. It is not taken for comment statements. A user-written IEFUJV exit routine can 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 job priority.
- Validate or assign the REGION request.
- Validate or assign job time and job step time parameters.

- Control output stream data by using the OUTLIM or SPACE parameters.
- Check for authorization to use data sets.
- Create user-written records.
- Assign the user identification to be included in the SMF termination records (types 4 and 5) and the SMF SYSOUT record (type 6).

Notes:

1. If a cataloged procedure is used, it is expanded *before* the IEFUJV 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 statements being overriden. Note, however, that symbolic parameters are resolved *after* the IEFUJV exit routine is taken.
2. When modifying a JCL statement, do not include additional JCL statements or continuation cards. Also, when adding or modifying operand fields, begin the first operand field in the same place that it was before any additions or modifications were made.
3. Depending upon the processing to be performed, it may be more efficient to check JOB and EXEC statement accounting fields in the IEFUJI exit routine and the first IEFUSI exit routine, respectively. The accounting fields are passed as parameters to IEFUJI and IEFUSI, making a statement scan routine unnecessary. Either of these exit routines can assign user identification, and the IEFACTRT exit routine can write messages to the system output message data set.
4. At job step or job termination, use the termination indicators in record types 4 and 5, respectively, to determine whether IEFUJV canceled the job.
5. To use installation-defined data sets with this exit routine, you must define them with a DD statement in the initiator cataloged procedure.
6. IEFUJV receives control when a job is reinterpreted for restart. IEFUIV does not receive control because the input stream is not reread.
7. In SMF SYSOUT records (type 6) the user identification appears only in the last type 6 record for that particular output class for that job.

Parameters

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

1. The address of the common exit parameter area. (See Figure 18.)
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 to this exit routine.)
3. The address of a one-byte area that indicates the type of JCL statement being passed to this exit routine. The indicator will be a binary value, as follows:
   0 indicates a null statement.
   1 indicates a JOB statement.
   2 indicates an EXEC statement.
   4 indicates a DD statement.
   8 indicates a PROC statement (for symbolic parameter definition).
   16 indicates all JCL images have been passed to IEFUJV.

Return Codes

Before the IEFUJV exit routine returns to the control program, it must place a return code in register 15, as follows:

0 indicates job processing should be continued.

4 indicates job processing should be canceled.

IEFUJI – Job Initiation Exit

IEFUJI receives control from the initiator when a job on the input queue is selected for initiation. A user-written IEFUJI exit routine can validate job accounting information or write to a user data set. It can also determine how long a job was on the input job queue before it was selected for initiation.

Notes:

1. If an installation uses major and minor account numbers with several fields, this exit routine is easier to use than IEFUJV for account number processing because the accounting fields are placed in a formatted list. Figure 20 shows the format of the JOB statement accounting information that is available to IEFUJI.

Offsets Dec.	Offsets Hex.	Length	Format	Description
0	0	1	binary	Number of accounting fields.
1	1	variable	EBCDIC	Accounting fields. Each entry for an accounting field contains the length of the field (one byte, binary) followed by the field (EBCDIC). A zero indicates an omitted field.

Figure 20. Format of Accounting Information Passed to IEFUJI, IEFUSI, IEFACTRT, and IEFUJP

2. At job step or job termination, use the termination indicators in record types 4 and 5, respectively, to determine whether IEFUJI canceled the job.
3. To use installation-defined data sets with this exit routine, you must define them with a DD statement in the initiator cataloged procedure.

Parameters

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

1. The address of the common exit parameter area. (See Figure 18.)
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 padded with blanks if necessary.
3. The address of a one-byte area indicating the requested job priority.
4. The address of an area containing the accounting information from the JOB statement. (See Figure 20.)

Return Codes

Before the IEFUJI exit routine returns control to the control program, it must place a return code in register 15, as follows:

0 indicates job processing should be continued.

4 indicates job processing should be canceled.

IEFUSI – Step Initiation Exit

IEFUSI receives control from the initiator before each job step is started (prior to allocation). A user-written IEFUSI exit routine can validate job step accounting information or write to a user data set.

Notes:

1. If OPT=1 was specified in the SMFxxxxx member or entered from the console at IPL time, this exit routine is not taken.
2. If an installation uses major and minor account numbers with several fields, this exit routine is easier to use than IEFUJV for account number processing because the accounting fields are placed in a formatted list. Figure 20 shows the format of the EXEC statement accounting information that is available to IEFUSI.
3. At job or job step termination, use the termination indicators in record types 4 and 5, respectively, to determine whether IEFUSI canceled the job.
4. To use installation-defined data sets with this exit routine, you must define them with a DD statement in the initiator cataloged procedure.

Parameters

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

1. The address of the common exit parameter area. (See Figure 18.)
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 padded with blanks if necessary.
3. The address of an eight-byte area containing the program name (in EBCDIC) from the EXEC statement. This area is aligned left and padded with blanks if necessary.
4. The address of an area containing the accounting information from the EXEC statement. (See Figure 20.)

Return Codes

Before the IEFUSI exit routine returns control to the control program, it must place a return code in register 15, as follows:

0 indicates job processing should be continued.

4 indicates job processing should be canceled.

IEFUTL – Time Limit Exit

IEFUTL receives control from the timer interruption handler when one of the following time limits expire:

- The job CPU time limit (from the JOB statement).
- The step CPU time limit (from the EXEC statement or the default from the reader procedure).
- The continuous wait time limit for the job (from the SMFxxxxx JWT parameter).

A user-written IEFUTL exit routine can control and record time expirations. For example, it can inform the operator that a job has exceeded its continuous wait time limit and request a reply to either cancel the job or extend the time limit.

Notes:

1. IEFUTL must be reenterable.
2. *A system interlock occurs anytime IEFUTL enqueues on a resource already enqueued on by the job step task or any of its subtasks.* (More specifically, the initiator abnormally terminates if IEFUTL enqueues on such a resource because the asynchronous exit interface routine sets a "step must complete" status before IEFUTL receives control.) The enqueue can come from within SVCs; for example, it can come from the SMFWTM macro instruction. To minimize the chances of an interlock, issue an ENQ macro of the following format before issuing an SMFWTM macro instruction:

   ```
   ENQ (SYSIEFSD,BUF,mode,3,SYSTEM),RET=TEST
   ```

   (Resource names used by system tasks are listed in the *OS/VS1 Programmer's Reference Digest*.) If a return code other than 0 is returned from this test ENQ, consider the resource unavailable. If IEFUTL does not run disabled, the status of the resource could change during the time between the test ENQ and the actual ENQ. (Refer to *OS/VS1 Supervisor Services and Macro Instructions* for detailed information on return codes from the ENQ macro instruction.)
3. The IEFUTL exit routine cannot access installation-defined data sets. It can access the SMF data set or write a message to the console; however, in doing so, a system interlock could occur. (See Note 2.)
4. If a job time limit is not specified on the JOB statement, the time limit for each job step is the value specified for the TIME= parameter on the EXEC statement, or the default value from the reader procedure.

   If a job time limit is specified on the JOB statement, the time limit for each job step is the remaining job time or the job step time limit (from the TIME= parameter or the reader procedure default), whichever is smaller.
5. If TIME=1440 is specified on the JOB statement, all timing for the job is eliminated. If TIME=1440 is specified on the EXEC statement and a job time limit is not specified on the JOB statement, all timing for the job step is eliminated.
6. You can extend execution time only within a step; each extension resets the limit for the wait currently in process. When the step with the time extension completes, 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.
7. In a VS1 system with extended timer support, the smallest time extension granted is 2^{20} microseconds or 1.048576 seconds.

Parameters

At entry to the IEFUTL exit routine, register 1 points to the four-byte address of the common exit parameter area. (See Figure 18.) Register 0 will have a binary value, as follows:

0 indicates the job CPU time limit expired.

4 indicates the step CPU time limit expired.

8 indicates the continuous wait time limit for the job expired.

Return Codes

Before the IEFUTL exit routine returns control to the control program, it must place a return code in register 15, as follows:

0 indicates job processing is to be canceled.

4 indicates job processing is to be continued with a time extension (in timer units).

You must place the time extension in register 1; you can determine the number of timer units by the algorithm:

1 second = 38400 timer units.

IEFUSO – SYSOUT Limit Exit

IEFUSO receives control from JES when the number of records written to an output data set exceeds the output limit for that data set. Unless an IEFUSO exit routine is supplied, jobs are canceled when the output limit is reached. To specify the output limit, use the JOUTLIM parameter of the JES macro instruction at SYSGEN time. Use the OUTLIM parameter on the DD statement defining the output data set to override the JOUTLIM value. Note that the OUTLIM parameter limits only output to spooled data sets; it does not apply to direct data sets. This parameter is described in *OS/VS1 JCL Reference.*

Notes:

1. The IEFUSO exit routine is not controlled by the SMFxxxxx EXT parameter; if SMF=BASIC or SMF=FULL was specified in the SCHEDULR macro instruction at SYSGEN time, and an IEFUSO exit routine was supplied, the IEFUSO routine will always receive control when the output limit is reached. Otherwise, the job is canceled.
2. If IEFUSO indicates that the output limit is to be increased (register 15=4) but you do not increase the limit (register 1=0), the exit routine will receive control again when the next record is written to the output data set.
3. The IEFUSO exit routine cannot access installation-defined data sets.

Parameters

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

1. The address of the common exit parameter area. (See Figure 18.)
2. The address of the DCB for the data set.

Return Codes

Before the IEFUSO exit routine returns control to the control program, it must place a return code in register 15, as follows:

0 indicates job step processing is to be canceled.

4 indicates job step processing is to be continued and the output limit is to be increased by the value placed in register 1.

IEFU83 – SMF Record Exit

IEFU83 receives control from the SVC 83 routine before each record is written to the SMF data set. This exit routine is not taken for records whose writing has been suppressed either because of a system failure, or because of options selected at IPL time (see "Selecting SMF Records Using SMFxxxxx Parameters" in the chapter "Defining the Use of SMF"). A user-written IEFU83 exit routine can select the records to be written to the SMF data set or check the circumstances that caused SMF to generate a given record. An example of the latter is asking the operator the reason for an IPL when SMF generates an IPL record. (IEFU83 can also direct output to the console.)

Notes:
1. IEFU83 must be reenterable.
2. The IEFU83 exit routine cannot access installation-defined data sets. Also, it cannot use the SMFWTM macro instruction to write to the SMF data set.
3. If the installation does not plan to use record types 63 (VSAM Entry Defined) or 67 (VSAM Entry Deleted), you can use IEFU83 to suppress writing them to the SMF data set or to truncate them.
4. The addresses of the user-communication and user-identification fields of the common exit parameter area are not passed to the IEFU83 exit routine.

Parameters

At entry to the IEFU83 exit routine, register 1 points to the four-byte address of the record descriptor word (RDW) of the SMF record to be written.

Return Codes

Before the IEFU83 exit routine returns control to the control program, it must place a return code in register 15, as follows:

0 indicates the record is to be written to the SMF data set.

4 indicates the record is not to be written to the SMF data set.

IEFACTRT – Termination Exit

IEFACTRT receives control from the terminator when each job or job step normally or abnormally terminates. If SMF=BASIC was specified in the SCHEDULR macro, IEFACTRT also receives control at the beginning of allocation. If OPT=1 was specified in the SMFxxxxx member of SYS1.PARMLIB or the operator entered OPT=1 from the console at system initialization time, this exit routine receives control only at job termination. A user-written IEFACTRT exit routine can perform various functions that are unique to an installation's requirements. For example, it can change the SMF record types 4 and 5 to user records and write them to an installation-defined data set for further analysis.

Notes:

1. IEFACTRT is the only exit routine than can write to the system output message data set, and only by passing a message to module IEFYS. If a user-written IEFACTRT exit routine writes messages for system output, the contents of register 12 must be the same as when the routine received control, and register 13 must contain the address of an 18-word work area. Figure 21 shows the procedure to use when writing system output messages from IEFACTRT. Note that the maximum number of characters printed on one line is 132.

```
         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[1]
VIEFYS   DC     V(IEFYS)
```

[1]The message will be truncated to 132 characters if necessary.

Figure 21. Writing System Output Messages from IEFACTRT

2. IEFACTRT can direct output to the console or to the system output device. It can also write to the SMF data set or to an installation-defined data set. To use installation-defined data sets with this exit routine, you must define them with a DD statement in the initiator cataloged procedure.
3. At job step or job termination, use the termination indicators in record types 4 and 5, respectively, to determine whether IEFACTRT canceled the job.

Parameters

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

1. The address of the common exit parameter area. If SMF=BASIC was specified at SYSGEN time, the address is that of the jobname. (See Figure 18.)
2. The address of an eight-byte area containing the job step name (in EBCDIC). This area is aligned left and padded with blanks if necessary. 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 padded with blanks if necessary.

4. The address of a four-byte area whose first three bytes contain the job CPU time in hundredths of a second (in binary), and whose last byte contains the number of accounting fields in the JOB statement (in binary).
5. The address of an area that contains accounting information from the JOB statement. This area has the format described earlier in Figure 20, excluding the first field shown (the number of accounting fields).
6. The address of a four-byte area whose first three bytes contain the step CPU time in hundredths of a second (in binary), and whose last byte contains the number of accounting fields in the EXEC statement (in binary). 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 20, excluding the first field shown (the number of accounting fields). 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, the 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. If SMF=BASIC was specified at SYSGEN time, 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 (type 4) or the job termination record (type 5) to be written to the SMF data set. If SMF=BASIC was specified at SYSGEN time, this field does not exist.

At entry to the IEFACTRT exit routine, register 0 contains a binary code indicating the reason for entry, as follows:

8 indicates job step allocation.

12 indicates job step termination.

16 indicates job termination.

Return Codes

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

- In register 1 (for SMF=FULL only):
  - 4 indicates the termination record is not to be written to the SMF data set.
  - A value other than 4 indicates the termination record is to be written to the SMF data set.
- In register 15:
  - 4 indicates the remaining job steps are to be canceled.
  - A value other than 4 indicates job processing is to be continued.

IEFUJP – Job Purge Exit

IEFUJP receives control from the SYSOUT writer routine when a job is ready to be purged from the system, that is, after a job has terminated and all the SYSOUT output that pertains to the job has been written. A user-written IEFUJP exit routine (and the SMFWTM macro instruction) can write additional data statistics found in the common exit parameter area to the SMF data set. (See Figure 18.)

Note: To use installation-defined data sets with this exit routine, you must define them with a DD statement in the writer cataloged procedure.

Parameters

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

1. The address of the common exit parameter area. (See Figure 18.)
2. Reserved.
3. The address of a 20-byte area containing the programmer's name (in EBCDIC). This area is aligned left and padded with blanks if necessary.
4. The address of a four-byte area whose first three bytes contain the job CPU time in hundredths of a second (in binary) and whose last byte contains the number of accounting fields in the JOB statement (in binary).
5. The address of an area that contains accounting information from the JOB statement. This area has the format described earlier in Figure 20, excluding the first field shown (the number of accounting fields).

Return Codes

IEFUJP does not return a code to the control program.

Testing Exit Routines

One method of testing user-written exit routines is by using the TESTEXIT procedure in SYS1.ASAMPLIB. This procedure contains an assembler language source program (also named TESTEXIT) which attaches the data generator utility program (IEBDG) to create sample parameter lists. The source program then calls each user-written exit routine being tested, and passes the appropriate parameter list to it. Figure 22 illustrates the input/output and control flow of the TESTEXIT source program.

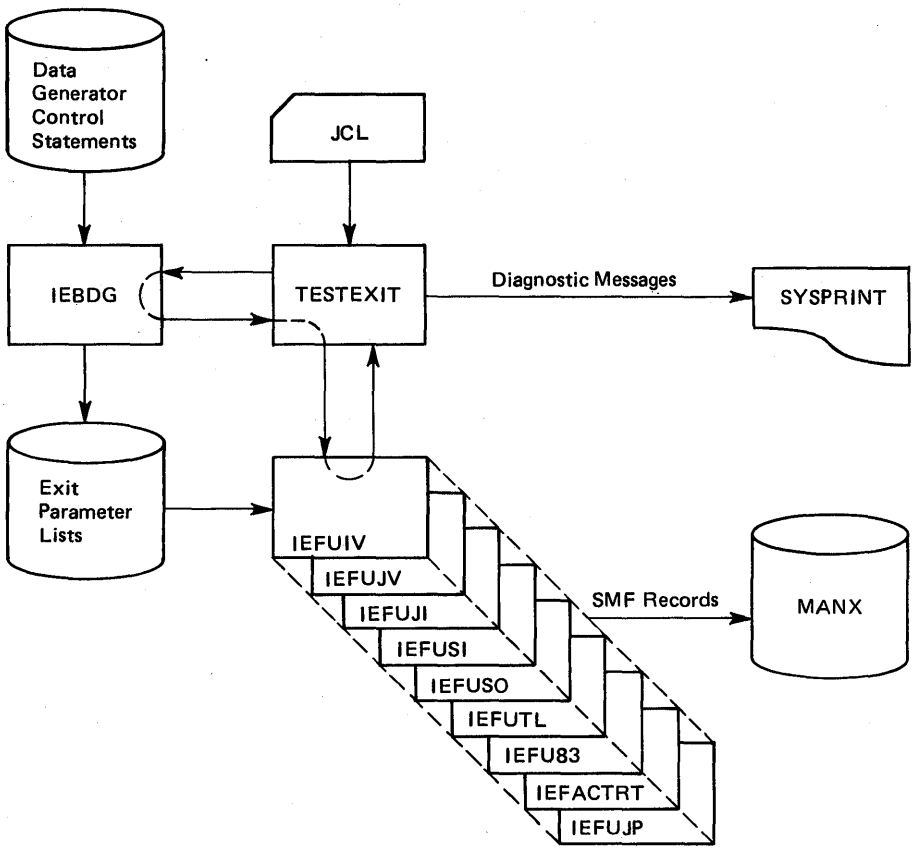


Figure 22. TESTEXIT Input/Output and Control Flow

Before using the TESTEXIT procedure:

1. Fulfill the following user-written exit routine testing requirements:
   - Specify a user subpool (0-127) in all GETMAIN macro instructions included in the routines.
   - Provide a special SMFWTM macro instruction in all routines that use the macro.
   - Place the routines in a partitioned data set.
2. Obtain the TESTEXIT procedure from SYS1.ASAMPLIB.
3. Modify the procedure to meet the installation's testing requirements.

TESTEXIT Exit Routine Requirements

Before using the TESTEXIT procedure, fulfill the following exit routine requirements:

- Specify a user subpool (0-127) in all GETMAIN macro instructions included in your routines. When testing is completed, specify one of the subpools shown in Figure 19 for the area used to communicate between exit routines.

- Provide a special SMFWTM macro instruction in all exit routines that use the macro. The special macro definition writes to the TESTEXIT data set defined by the DD statement named MANX. (With the normal SMFWTM macro instruction, the data is written to SYS1.MANX or SYS1.MANY.) Using this macro definition, then, data is processed without accessing the system data on SYS1.MANX or SYS1.MANY. When testing is completed, remove the macro definition. Figure 23 shows the SMFWTM macro instruction that is required for using the TESTEXIT procedure.

```
         MACRO
&NAME    SMFWTM   &MSGAD
         AIF      (T'&MSGAD EQ 'O').E1
         AIF      ('&MSGAD'(1,1) EQ '(').REGA
&NAME    LA       1,&MSGAD
.LIST    CNOP     0,4
         BAL      15,*+8
         DC       V(TSMFWTM)
         L        15,0(15)
         BALR     14,15
         MEXIT
.REGA    ANOP
&NAME    DS       0H
         AIF      ('&MSGAD' EQ '(1)').LIST
         LR       1,&MSGAD(1)
         AGO      .LIST
.E1      MNOTE    4,'*** NO OPERAND SPECIFIED ***'
         MEND
```

Figure 23. SMFWTM Macro Definition Required for Using TESTEXIT

- Place the exit routines in a partitioned data set named EXITLIB. Figure 24 shows sample JCL for entering the routines into EXITLIB.

```
//UPDTE     JOB   MSGLEVEL=1
//          EXEC  PGM=IEBUPDTE,PARM=NEW
//SYSUT2    DD    DSNAME=EXITLIB,VOLUME=SER=231400,
//          UNIT=2314,SPACE=(TRK,(10,3,1)),DISP=(,KEEP),
//          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 24. Sample JCL for Entering User-Written Exit Routines into EXITLIB

Obtaining TESTEXIT from SYS1.ASAMPLIB

Figure 25 shows sample JCL for obtaining a punched deck of TESTEXIT from SYS1.ASAMPLIB.

```
//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.

Figure 25. Sample JCL for Obtaining a Punched Deck of TESTEXIT

Modifying the TESTEXIT Procedure

Figure 26 shows sample JCL for executing the TESTEXIT procedure in an unmodified system.

```
//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     DATA
./ 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)
./ ADD            NAME=UJP
(IEBDG Control Statements for IEFUJP)
```

Figure 26. Sample JCL for Executing TESTEXIT (Part 1 of 2)

```
./ 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 26. Sample JCL for Executing TESTEXIT (Part 2 of 2)

A summary of the operations performed by the procedure shown in Figure 26 is as follows:

- The TESTEXIT job assembles the TESTEXIT source program (not illustrated in the figure) and link-edits it with the exit routines being tested. (Note that the exit routines must reside in EXITLIB, a partitioned data set.)
- 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 source program.
- The TESTING job includes the execution of the TESTEXIT source program.

Use the TESTEXIT procedure provided in SYS1.ASAMPLIB, without modification, to link-edit the sample exit routines in SYS1.ASAMPLIB, generate sample parameter lists, and test the sample exit routines. To adapt the TESTEXIT procedure to your installation's testing requirements, however, note the following modifications:

- The TESTEXIT job shown in Figure 26 link-edits the TESTEXIT source program with the exit routines. You must substitute an INCLUDE control statement specifying the names of the exit routines being tested.
- The DATAGEN job shown in Figure 26 creates a partitioned data set containing control statements that generate samples of standard parameter lists. You should omit control statements and their associated ADD statements for any exit routines not being tested. When testing for special conditions or required additional test parameters, you must make appropriate modifications and additions to the control statements.
  Note that you must supply control statements in such an order that the records subsequently generated by the IEBDG utility program will be grouped as complete parameter lists that conform in length and format to the exit parameters previously defined in this chapter. (Be sure to include the entry code passed to exits IEFUTL and IEFACTRT in register 0 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, see *OS/VS1 Utilities.*
- The TESTING job shown in Figure 26 includes the execution of the TESTEXIT source 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:

xxx
is an exit routine identifier.

nnn
is the number of times an exit routine is to be tested (the maximum value is 255).

The DD statements to be included depend upon the exit routines being tested. Figure 27 shows the exit-routine identifiers, specified on the EXEC statement, and the DD statements that you must include for each exit routine being tested.

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 27. Parameters and DD Statements for Executing TESTEXIT

Note that you must include DD statements for any other data sets used by the exit routines in the JCL for the TESTEXIT procedure.

Producing a report usually requires at least two operations: sorting the SMF records and writing them in an appropriate format.

Sorting SMF Records

Any sort/merge program can be used to sort SMF records; this section describes two sample sort/merge exit routines that you may use with the IBM OS/VS Sort/Merge Program Product (Program No. 5740-SM1). (For a detailed description of this particular sort/merge program, see *OS/VS Sort/Merge Programmer's Guide.*)

Sample Sort/Merge Exit Routines

The IBM OS/VS Sort/Merge Program Product can, during various phases of execution, pass control to routines designed and written to perform specific functions. SYS1.ASAMPLIB has two sample routines that receive control from exits E15 and E35 of this sort/merge program. The sample E15 exit routine, called SMFE15, extracts all SMF records without a job log identification (job name and time and date that the reader recognized the JOB card) from the SMF dump data set. SMFE15 retains the dump header and dump trailer records (types 2 and 3) in the temporary data set HDRDATA. It retains all other system-oriented records (records without a job log identification) in the temporary data set SORDATA.

The sample E35 exit routine, called SMFE35, places all the records extracted by the SMFE15 routine in the sort output data set. These records are inserted in the data set as follows: dump header records, dump trailer records, all other system-oriented records, and the sorted job-oriented records.

Note: If tape work devices are used, the minimum record length the IBM Sort/Merge Program Product can sort is 18 bytes. Otherwise, the minimum is one byte. The sample routines SMFE15 and SMFE35 use SMF record types 0 through 13 for input; the minimum length of these SMF records is 18 bytes.

Figure 28 shows sample JCL for obtaining a listing of the SMFE15 and SMFE35 exit routines from SYS1.ASAMPLIB. Figure 28 also shows sample JCL for obtaining a listing of the SYS1.ASAMPLIB member named SMFSORT. SMFSORT contains sample JCL for executing the IBM OS/VS Sort/Merge Program Product.

```
//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.

Figure 28. Sample JCL for Obtaining a Listing of Sample Sort Exit Routines

To include the sample exit routines in your sort application, you must assemble and link-edit them before executing the sort/merge program. Figure 29 shows sample JCL for this procedure, including one possible sort application. In this example, SMF records are to be sorted first on the job log identification (major control field), and then on the date and time portions of the time stamp (minor control fields). Displacements of these fields (as shown in the next chapter) are 14, 6, and 2. Note, however, that you must add an additional four bytes for the RDW and one byte for the sort procedure's initial count of 1 to these displacements. Hence, displacements 19, 11, and 7 are shown in the SORT FIELDS= statement in Figure 29.

```
//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)),
//             VOL=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, VOL=SER=SYSMAN,DISP=OLD,[6]
//           LABEL=(,SL),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=(,SL),[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). (For a description of the ASMFCL procedure, see *OS/VS—VM/370 Assembler Programmer's Guide*.)

[2] The sample sort exit routines will be link-edited into data set SMF1.EXITS.

[3] Link-edit 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 can 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 must be specified and can be larger than the BLKSIZE value because records might be segmented. 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 BUF parameter. Modify these parameters according to the installation's buffer size and the longest record to be 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 29. Sample JCL for Executing a Sort Procedure

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.

SYS1.ASAMPLIB has a sample assembler source report program, called SMFFRMT. Figure 30 illustrates sample output from the SMFFRMT program. To use this program to print selected types of SMF records, specify the record types to be printed, separated by commas, in the PARM field of the EXEC statement. To print all record types, the PARM parameter is not required.

```
RECORD                    HEADER/RECORD                                          DATE          720117
 TYPE                                                                            PAGE               1

 02     0102  00681590  0072017F  C2C2F4F5                                *.. .... .... BB45             *

 08     0108  0066D2E5  0072017F  C2C2F4F5                                *.. ..KV .... BB45             *
        002E  0823000908020000D0808000E0808000F2008023020080231200802    *.. ............................*
        3220080233200802342008023520080236                               *.................              *

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

 13     010D  0066D32F  0072017F  C2C2F4F5                                *.. ..L. .... BB45             *
        002E  000040000001404040404040404040404040404040C1  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  00000000000000000000000000000  *. ..... . .. A . ..............*
        000000000000  2E  D7C5D5C3C540D1E64040404040404040404040404040   *..... . PENCE JW              *
        000173  01  15  F7F6F0F1F0F1F3F1F6F2F0F2F7D5F0F2F2D7D9D6C4       *... . . 7601013162027NO22PROD *

 06     0106  006774D1  0072017F  C2C2F4F5                                *.. ...J .... BB45             *
        C1C1D1F2F5F6C1F0  00672FFE  0072017F  0000000000000000  C1       *AAJS56A0 .... .... ........ A *
        00677408  0072017F  00000008  00  01  40404040  C3C5D5E3D9C1      *.... .... .... . .      CENTRA*
        D340  0000000000000000  00000000  00000000  00000000  000000    *L  ........ .... .... .... ...*
        00  00000000  00000000  00  00                                    *. .... .... . .               *

 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  00000000000000000000000000000  *. ..... . .. A . ..............*
        000000000000  2E  D7C5D5C3C540D1E64040404040404040404040404040   *..... . PENCE JW              *
        00031C  01  15  F7F6F0F1F0F1F3F1F6F2F0F2F7D5F0F2F2D7D9D6C4       *... . . 7601013162027NO22PROD *

 06     0106  0067B073  0072017F  C2C2F4F5                                *.. .... .... BB45             *
        C1C1D1E2F5F6C2F0  00677F2F  0072017F  0000000000000000  C1       *AAJS56B0 .... .... ........ A *
        0067AF96  0072017F  00000008  00  01  40404040  C3C5D5E3D9C1      *.... .... .... . .      CENTRA*
        D340  0203010000000000  E3D54040  C8D54040  00000000  000000    *L  ........ TN   HN   .... ...*
        00  C3D6D7F1  C6D3C1F1  04 C0                                     *. COP1 FLA1 . .               *

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

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

Figure 30. Sample Output from SMFFRMT

Figure 31 shows sample JCL for executing the SMFFRMT program after you have assembled it and link-edited it into SYS1.LINKLIB.

```
//FORMAT    JOB   MSGLEVEL=1
//FRMT      EXEC  PGM=SMFFRMT,PARM='1,2,3,5,6,8,9,10,11,12,13'
//REPORT    DD    SYSOUT=A,DCB=(BLKSIZE=1089)[2]
//SMFDATA   DD    DISP=(OLD,KEEP),LABEL=(,SL),VOL=SER=xxxxxx,
//               UNIT=2400,DCB=(RECFM=VBS,BLKSIZE=200,LRECL=600)[1]
```

[1] The LRECL value must be specified and can be larger than the BLKSIZE value because records might be segmented. 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 BUF parameter. Modify these parameters according to the installation's buffer size and the longest record to be collected. See Figure 10 for the JCL defining the SMFDATA data set.

[2] The BLKSIZE value must be a multiple of 121. The record format is FBA.

Figure 31. Sample JCL for Executing SMFFRMT

This chapter fully describes all of the SMF-formatted records. Note that the fields in these records marked "Reserved" are for use by SMF and are not available for your use.

Standard SMF Record Header

Each record written to the SMF data set by the SMF writer routine contains the standard SMF record header. Each record written to the SMF data set by user-written routines should also include the standard record header. Figure 32 illustrates the header; its length is 14 bytes.

Offsets		Name	Length	Format	Source	Description
0	0	(The "Name" field in all of the SMF records contains the symbolic addresses defined by the IFASMFR macro instruction.)	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1		1	binary	internal	Record type
2	2		4	binary	SVC 11 or SVC 83	Time, in hundredths of a second, record was moved to SMF buffer[1]
6	6		4	packed	SVC 11 or SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign[1]
10	A		4	EBCDIC	JMRCPUID or SMCASID	System identification (taken from SID parameter

[1] In record types 2 and 3, these fields indicate the time and date that the record was moved to the dump data set. In record types 4 and 5, these fields indicate the time and date that the job or job step terminated.

Figure 32. Standard SMF Record Header

Note: When using the SMFWTM macro instruction to write a record to the SMF data set, you must add a record descriptor word (four bytes) to the beginning of the record header; the address passed to the macro instruction must point to the beginning of the record descriptor word (RDW). For a discussion of the RDW, see *OS/VS Data Management Services Guide.*

Summary of SMF Records

Figure 33 summarizes the SMF records according to the type of data they collect, such as VSAM or RTAM data, and the events (or status indicators) that cause SMF to generate the records. Also included in Figure 33 is a summary of the DSV and OPT SMFxxxxx parameters that control whether each record type is to be written to the SMF data set. To write all SMF-formatted records (except record types 2, 3 and 7) to the SMF data set, you must specify MAN=ALL and a value for the BUF parameter. To write record type 21 to the SMF data set, you must specify ESV=SMF in the SCHEDULR macro instruction at SYSGEN time.
To write record type 17 for temporary data sets, you must specify REC=2.

For a summary of the sizes of the SMF records, see "SMF Buffer" in the chapter "System Requirements and Considerations."

Catagory of Data	Event or Status		Record Type	SMFxxxxx Definition	
				DSV=	OPT=
Day Data	IPL		0		
	Devices online at IPL		8		
			19	1,3	
			22		
	Partition defined		13		
	End of day		12		
			19	1,3	
Machine Data	Accumulated wait time		1		
	Devices varied online		9		
			22		
	Devices recovered at allocation		10		
	Devices varied offline		11		
			22		
Auxiliary Storage Data	Space available on DASD volumes: at IPL, after HALT EOD or SWITCH SMF command, when demounted		19	1,3	
	Tape volume closed or processed by EOV		21		
	VSAM data space defined, extended or deleted		69	1,3	
Processing Data	Step processing		4		2
	Job processing		5		
	SYSOUT processing		6		
	Job initiated		20		
Non-VSAM Data Set Activity Data	Data set closed or processed by EOV	Data set opened for INPUT or RDBACK	14	2,3	2
		Data set opened for OUTPUT, UPDAT, INOUT, or OUTIN	15	2,3	2
	Data set scratched		17	2,3	2
	Data set renamed		18	2,3	2
VSAM Data Set Activity Data	Component or cluster opened		62	2,3	2
	Entry defined		63	2,3	2
	Component or cluster status		64	2,3	2
	Entry deleted		67	2,3	2
	Entry renamed		68	2,3	2
RTAM Data	RTAM initialized		43		
	MODIFY RTAM command issued		44		
	STOP RTAM command issued		45		
	LOGON record received by RTAM		47		
	LOGOFF record received by RTAM		48		
	Invalid LOGON record received by RTAM		49		
ACF/VTAM Data	ACF/VTAM Tuning Statistics		50		
Subsystem Data	Subsystem Data		51		
Record Management Data	Dump header		2		
	Dump trailer		3		
	SMF records lost		7		
	Record descriptor word (RDW)		ALL	N/A	N/A
	Block descriptor word (BDW)		N/A	N/A	N/A
Security Data	Cryptography (5740-XY5)		82		

Figure 33. Summary of SMF Records

Record Type 0 – IPL

Record type 0 is written by IEESMFI2 after every IPL of the system. It includes the virtual and real storage sizes and the SMF options in effect. Its length is 31 bytes.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF0FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF0RTY	1	binary	internal	Record type
2	2	SMF0TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF0DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF0SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF0JWT	4	binary	SMCAJWT	Limit, in minutes, of continuous wait time for the job (taken from JWT parameter)
18	12	SMF0BUF	4	binary	SMCABUF*2	Number of bytes in SMF buffer (taken from BUF parameter)
22	16	SMF0VST	4	binary	(CVTMZ00+1)/1024	Number of 1K bytes in virtual storage
26	1A	SMF0OPT	1	binary	SMCAOPT	SMF options *Bit* *Meaning When Set* 0 System and job accounting (OPT=1) 1 System, job and step accounting (OPT=2) 2 User exits will be taken (EXT=YES) 3 Data set accounting (DSV=2 or 3) 4 Volume accounting (DSV=1 or 3) 5 Reserved 6 Type 17 records will be written for temporary data sets (REC=2) 7 Reserved
27	1B	SMF0RST	4	binary	(CVTEORM+1)/1024	Number of 1K bytes in real storage

Record Type 1 – System Statistics

Record type 1 is written by IEESMFI2 after every IPL of the system and by IEFSMFLK at the first job or job step termination following the expiration of a ten-minute interval. Its length is 34 bytes.

This record contains the system wait time and paging counts accumulated during all the ten-minute intervals that expired since the last record type 1 or 12 was written. It also contains the expiration time of the last ten-minute interval.

Notes:

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

2. If the operator pushes the STOP button to suspend CPU processing on an IBM System/370, timing of the ten-minute interval is suspended but the time-of-day clock continues to run. The ten-minute interval is based on CPU-processing – not on the time-of-day clock. Therefore, the wait time interval reflected by the time stamp (offset 2) is equal to the ten-minute CPU-processing interval plus the time that the CPU processing was stopped.

3. The following algorithm shows the relationship among elapsed time, job time, wait time, and system time:

   Elapsed Time = Job Time + Wait Time + System Time

   Elapsed time is obtained by subtracting the time stamp value on the first type 1 record from the time stamp value on the type 12 record.

   Job time is obtained by summing the job CPU time values from all the type 5 records produced during the elapsed time.

   Wait time is obtained by summing the wait time values from the type 12 record and all but the first type 1 records.

   System time is calculated when the other three values of the above algorithm are known.

Wait Time and Paging Count Collection

SMF divides the CPU-processing time (elapsed time) into ten-minute intervals. The type 1 record written after every IPL of the system marks the beginning of the first ten-minute interval. At the termination of a job or job step, SMF determines whether at least one ten-minute interval has expired since the last type 1 record was written. Processing continues, as follows:

- If a ten-minute interval has expired, a type 1 record is produced and the wait time and paging counts accumulated during the expired interval are moved into the record.
- If more than one ten-minute interval has expired, the wait time and paging counts accumulated during the intervals that have expired since the last type 1 record are moved into the record.
- If a ten-minute interval has not expired, a type 1 record is not produced.

Figure 34 is an example of how SMF collects system wait time. (The process is similar for collecting system paging statistics.) In Figure 34, when job/step A terminates, three ten-minute intervals have expired. The total wait time collected in these intervals (783 seconds) is moved to a type 1 record. When job/step B terminates, a ten-minute interval has not expired since the last type 1 record was written; therefore, a type 1 record is not written. When job/step C terminates, three ten-minute intervals have expired, and the total wait time for these intervals (809 seconds) is moved to a type 1 record.

Wait time collected in successive ten-minute intervals	217	263	303	342	265	202		
Wait time collected in a type 1 record			783			809		
Job or job step termination			A	B		C		

Figure 34. Sample Wait Time Collection

The format of record type 1 is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF1FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-6 Reserved 7 VS1
1	1	SMF1RTY	1	binary	internal	Record type
2	2	SMF1TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF1DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF1SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF1SWT	4	binary	SMCAWAIT	System wait time, in hundredths of a second, for successive ten-minute intervals that have expired since the last record type 1 or 12 was written
18	12	SMF1TEXP	4	binary	SMCATEXP	Expiration time, in hundredths of a second, of the interval whose statistics are reported in this record
22	16	SMF1PGIN	4	binary	SMCAPGIN	Number of page-ins for the entire system during the interval
26	1A	SMF1PGOT	4	binary	SMCAPGOT	Number of page-outs for the entire system during the interval
30	1E	SMF1PGRL	4	binary	SMCAPGRL	Number of page reclaims for the entire system during the interval

Record Type 2 – Dump Header

Record type 2 is written directly to the dump data set by IFASMFDP, the SMF dump program. This record consists of only the standard SMF record header. It indicates the beginning of a dump of the SMF data set from a direct access device usually to a tape. Its length is 14 bytes.

Note: Even if MAN=USER is specified in the SMFxxxxx member of SYS1.PARMLIB, a type 2 record is written.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF2FLG	1	binary	IFASMFDP	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF2RTY	1	binary	internal	Record type
2	2	SMF2TME	4	binary	SVC 11 (Set by IFASMFDP)	Time, in hundredths of a second, record was moved to the dump data set
6	6	SMF2DTE	4	packed	SVC 11 (Set by IFASMFDP)	Date record was moved to the dump data set, in the form 00YYDDDF where F is the sign
10	A	SMF2SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)

Record Type 3 – Dump Trailer

Record type 3 is written directly to the dump data set by IFASMFDP, the SMF dump program. This record consists of only the standard SMF record header. It marks the end of a dump of the SMF data set from a direct access device usually to a tape. Its length is 14 bytes.

Note: Even if MAN=USER is specified in the SMFxxxxx member of SYS1.PARMLIB, a type 3 record is written.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF3FLG	1	binary	IFASMFDP	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF3RTY	1	binary	internal	Record type
2	2	SMF3TME	4	binary	SVC 11 (Set by IFASMFDP)	Time, in hundredths of a second, record was moved to the dump data set
6	6	SMF3DTE	4	packed	SVC 11 (Set by IFASMFDP)	Date record was moved to the dump data set, in the form 00YYDDDF where F is the sign
10	A	SMF3SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)

Record Type 4 – Step Termination

Record type 4 is built by IEFSMFWI and written by IEFSMFLK at the normal or abnormal termination of a job step, or when a job step is flushed during or after job initiation. It is not written for a job step that follows a CANCEL operator command. Its length is 117 bytes plus (1) eight bytes for each device entry and (2) the length of the step accounting fields.

This record identifies the job step by the job log identification, step name, number of the step within the job, user identification, and program name. If accounting numbers (which can be alphameric) were specified in the EXEC statement, they are included.

This record also contains operating information such as the job step start and end times, step CPU time, amount of storage allocated and used, step termination status, number of records in instream DD DATA and DD * data sets for the step, device allocation start time, problem program start time, and storage protect key. Record type 4 also has an entry for each non-spooled data set that was defined by a DD statement. Each entry lists the EXCP count for the data set, device class, unit type, channel address, and unit address.

Notes:

1. 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.)
2. The EXCP count indicates the input/output activity required by the job. It includes EXCPs issued via SVC 0, EXCPs issued in a user's program controlled interrupt (PCI), EXCPs issued in a user's channel-end and abnormal-end appendages, and the input/output for VSAM data sets. It does not include EXCPs for system services performed in system keys as a result of the problem program (for example, joblib/steplib processing, overlay supervisor processing, and checkpoint data set processing). The EXCP count also does not include PCIs when ADDRSPC=REAL is specified.
3. EXCPs for concatenated data sets residing on the same volume are accumulated in the device entry corresponding to the first occurrence of the UCB address for the device the volume is on.
4. When chained scheduling is used, the EXCP count may not be constant between different runs of the same job. Chained scheduling is designed to optimize input/output activity; the number of EXCPs required will depend upon system and program interaction at the time the input/output is performed.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF4FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF4RTY	1	binary	internal	Record type
2	2	SMF4TME	4	binary	SVC 11 (Set by IEFSMFWI)	Time, in hundredths of a second, step terminated
6	6	SMF4DTE	4	packed	SVC 11 (Set by IEFSMFWI)	Date step terminated, in the form 00YYDDDF, where F is the sign
10	A	SMF4SID	4	EBCDIC	JMRCPUID	System identification (taken from SID parameter)
14	E	SMF4JBN	8	EBCDIC	JMRJOB	Job name[1]
22	16	SMF4RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job[1]
26	1A	SMF4RDS	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign[1]
30	1E	SMF4UIF	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)
38	26	SMF4STN	1	binary	LCTSNUMB	Step number (first step=1, etc.)
39	27	SMF4SIT	4	binary	JCTJMRSS (Set by IEFSMFIE)	Time, in hundredths of a second, initiator selected this step
43	2B	SMF4STID	4	packed	JCTSSD (Set by IEFSMFIE)	Date initiator selected this step, in the form 00YYDDDF where F is the sign
47	2F	SMF4NCI	4	binary	SCTSMF	Number of card-image records in instream DD DATA and DD* data sets read by the reader for the step. (This number does not include records from associated data sets.)

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

(Continued)

Offsets		Name	Length	Format	Source	Description
51	33	SMF4SCC	2	binary	LCTSMF	Step completion code: X'0ccc' indicates system ABEND in the job step where ccc is the system ABEND code. (See *OS/VS Message Library: VS1 System Codes.*) X'8ccc' indicates user ABEND in the job step where ccc is the user ABEND code. X'nnnn' indicates normal completion where nnnn is the contents of the two low-order bytes in register 15 at termination. X'0000' indicates either (1) the job step was flushed (not executed) because of an error during allocation or in a preceding job step, or (2) normal job completion with a return code of 0. Use this field in conjunction with the step termination indicator field (offset 83).
53	35	SMF4PRTY	1	binary	TCBDSP	Step priority
54	36	SMF4PGMN	8	EBCDIC	SCTPGMNM	Program name (taken from PGM= parameter on EXEC card). If a backward reference was used (SCTSTYPE equals X'80'), then this field contains *.DD.
62	3E	SMF4STMN	8	EBCDIC	SCTSNAME	Step name (taken from name on EXEC card)
70	46	SMF4RSH0	2	binary	TCTRSZ	Partition size requested, in 1K units. If ADDRSPC=REAL is specified, this field equals the amount of real storage requested (see offset 98).
72	48		2	binary		Reserved
74	4A	SMF4H0ST	2	binary	(TCTRSV–TCTMINC)*2	Storage used, in 1K units. If ADDRSPC=REAL is specified, this field equals the amount of real storage requested (see offset 98).
76	4C		6	binary		Reserved
82	52	SMF4SPK	1	binary	TCBPKF	Storage protect key, in the form xxxx0000 where xxxx is the key
83	53	SMF4STI	1	binary		Step termination indicator *Bit* *Meaning When Set* 0 Reserved JCTJMRCL 1 Canceled by exit IEFUJV[2] JCTJMRCL 2 Canceled by exit IEFUJI[2] JCTJMRCL 3 Canceled by exit IEFUSI[2] JCTJMRCL 4 Canceled by exit IEFACTRT[2] 5 Reserved TCBFA 6 If 0, normal completion. If 1, ABEND. If step completion code (offset 51) equals 0322 or 0522, IEFUTL caused ABEND. If step completion code equals 0722, IEFUSO caused ABEND. LCTTMWRK 7 If 0, normal completion. If 1, step was flushed.
84	54		2	binary		Reserved
86	56	SMF4AST	4	binary	TCTAST	Device allocation start time, in hundredths of a second
90	5A	SMF4PPST	4	binary	TCTPPST	Problem program start time, in hundredths of a second
94	5E		4	binary		Reserved

[2]Job steps canceled by IEFUJV, IEFUJI, and IEFUSI will not be executed; therefore bit 7 will also be on. Job steps canceled by IEFACTRT will cause subsequent job steps to be canceled; bit 7 will be on for subsequent job steps.

(Continued)

Offsets		Name	Length	Format	Source	Description
98	62	SMF4RIN	2	binary	SCTSSTAT	Record indicator *Bit* *Meaning When Set* 0-6 Reserved 7 If 0, storage is virtual. If 1, storage is real. 8-15 Reserved
100	64	SMF4RLCT	2	binary	internal	Offset from the beginning of the record header to the relocate section
102	66	SMF4LENN	2	binary	internal	Length of device entry portion of record. Calculated as: (8 times the number of devices) + 2
For each device assigned to each *non-spooled* data set, there is an eight-byte entry with the following format.[3]						
+0		SMF4DEVC	1	binary	UCBTBYT3	Device class
+1		SMF4UTYP	1	binary	UCBTBYT4	Unit type
+2		SMF4CUAD	1	binary	UCBCHA	Channel address. (A mass storage volume is indicated when the high-order bit of this field is on.)
+3			1	binary	UCBUA	Unit address
+4		SMF4EXCP	4	binary	TCTDCTR	EXCP count (see notes for this record)
After the device entries are the following fields:						
Accounting Section:						
+0		SMF4LNTH	1	binary	internal	Length of accounting section, excluding this field
+1		SMF4SETM	3	binary	ACTJTIME (in JCT)	Step CPU time, in hundredths of a second [4]
+4		SMF4NAF	1	binary	internal	Number of accounting fields
+5		SMF4ACTF	variable	EBCDIC	EXEC statement	Accounting fields. Each entry for an accounting field contains the length of the field (one byte, binary) followed by the field (EBCDIC). A zero indicates an omitted field.
Relocate Section:						
+0[5]		SMF4PGIN[5]	4	binary	TCTPGIN (Set by IEAPGSPP)	Number of page-ins done to the partition. This field includes page-ins required through page faults, specific page requests, and page fixes.
+4		SMF4PGOT	4	binary	TCTPGOUT (Set by IEAPGSPP)	Number of page-outs done to the partition. This field includes pages stolen by the paging supervisor as a result of infrequent use.

[3]For a DD DUMMY data set, the entry is zero. (A DD DUMMY entry results when a forward reference to a DD statement having that DD name is not found or when DD DUMMY is specified.)

[4]If TIME=1440 is specified on the JOB statement, all timing for the job is eliminated. If TIME=1440 is specified on the EXEC statement and a job time limit is not specified on the JOB statement, all timing for the job step is eliminated. For more information, see the IEFUTL exit.

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

[5]The displacement of this field depends upon the size of the accounting fields and the number of devices. Offset 100 contains the displacement for this field.

Record Type 5 – Job Termination

Record type 5 is built by IEFSMFWI and written by IEFSMFLK at the normal or abnormal termination of a job, or when a job step is flushed during or after job initiation. Its length is 117 bytes plus the length of the job accounting fields. (The maximum length of this record is 261 bytes.)

This record identifies the job by job log identification, user identification, priority, input class, SYSOUT class indicators, and programmer's name. If accounting numbers (which can be alphameric) were specified in the JOB statement, they are included.

This record also contains operating information such as the job start and end times, number of steps in the job, number of records in instream DD DATA and DD* data sets for the job, job termination status, reader device class and unit type, storage protect key, and job CPU time. (The job CPU time equals the sum of the job step times.)

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF5FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF5RTY	1	binary	internal	Record type
2	2	SMF5TME	4	binary	SVC 11 (Set by IEFSMFWI)	Time, in hundredths of a second, job terminated
6	6	SMF5DTE	4	packed	SVC 11 (Set by IEFSMFWI)	Date job terminated, in the form 00YYDDDF where F is the sign
10	A	SMF5SID	4	EBCDIC	JMRCPUID	System identification (taken from SID parameter)
14	E	SMF5JBN	8	EBCDIC	JMRJOB	Job name[1]
22	16	SMF5RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job[1]
26	1A	SMF5RSD	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign[1]
30	1E	SMF5UIF	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)
38	26	SMF5NST	1	binary	LCTSNUMB	Number of steps in the job
39	27	SMF5JIT	4	binary	JCTJMRJT (Set by IEFSMFIE)	Time, in hundredths of a second, initiator selected the job
43	2B	SMF5JID	4	packed	JCTJMRJD (Set by IEFSMFIE)	Date initiator selected the job, in the form 00YYDDDF where F is the sign
47	2F	SMF5NCI	4	binary	JMRJOBIN	Number of card-image records in instream DD DATA and DD* data sets read by the reader for the job. (This number does not include records from associated data sets.)

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

(Continued)

Offsets		Name	Length	Format	Source	Description
51	33	SMF5JCC	2	binary	LCTSMF	Job completion code: X'0ccc' indicates system ABEND in the last job step where ccc is the system ABEND code. (See *OS/VS Message Library: VS1 System Codes.*) X'8ccc' indicates user ABEND in the last job step where ccc is the user ABEND code. X'nnnn' indicates normal completion where nnnn is the contents of the two low-order bytes in register 15 at termination. X'0000' indicates either (1) a job step was flushed (not executed) because of an error during allocation or in a preceding job step, or (2) normal job completion with a return code 0. Use this field in conjunction with the job termination indicator field (offset 62).
53	35	SMF5JPTY	1	binary	JLPPRI (in JLPA)	Job priority. This field normally equals the user-assigned priority of 0 to 13, but if the job fails while being scheduled, this field equals 14.
54	36	SMF5RSTT	4	binary	JMRDRSTP	Time, in hundredths of a second, reader recognized the end of the job. (This does not include time to read records from associated data sets.)
58	3A	SMF5RSTD	4	packed	JMRDRSTP+4	Date reader recognized the end of the job, in the form 00YYDDDF where F is the sign.
62	3E	SMF5JBTI	1	binary	 JCTJMRCL JCTJMRCL JCTJMRCL JCTJMRCL TCBFA	Job termination indicator *Bit* *Meaning When Set* 0 Reserved 1 Canceled by exit IEFUJV 2 Canceled by exit IEFUJI 3 Canceled by exit IEFUSI 4 Canceled by exit IEFACTRT 5 Reserved 6 If 0, normal completion. If 1, ABEND. 7 Reserved
63	3F	SMF5SMCI	5	binary	JCTCTOJ	SYSOUT class indicator.[2] (This field is usually taken from the SYSOUT parameter on the DD statement.)
68	44		1	binary		Reserved
69	45	SMF5RDCL	1	binary	JMRRDR	Reader device class[3]
70	46	SMF5RUTY	1	binary	JMRRDR	Reader unit type[3]
71	47	SMF5JICL	1	EBCDIC	JLPID (in JLPA)	Job input class (default equals 'A')
72	48	SMF5SPK	1	binary	TCBPKF	Storage protect key, in the form xxxx0000 where xxxx is the key
73	49		3	binary		Reserved
76	4C	SMF5QID	7	EBCDIC	JMRRESID	User's logon identifier. Non-terminal oriented jobs have an identifier of 'CENTRAL'; terminal-oriented jobs use the contents of the JMRRESID field for the terminal through which the job was submitted.

(Continued)

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

Byte 0		*Byte 1*		*Byte 2*		*Byte 3*		*Byte 4*	
Bit	*Class*	*Bit*	*Class*	*Bit*	*Class*	*Bit*	*Class*	*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		

[3] This field is not filled in for jobs submitted via an internal reader.

Offsets		Name	Length	Format	Source	Description
83	53		1	EBCDIC		Reserved
84	54		8	binary		Reserved
92	5C	SMF5TLEN	1	binary	internal	Length of rest of record excluding this field
93	5D	SMF5PRGN	20	EBCDIC	ACTPRGNM (in JCT)	Programmer's name
113	71	SMF5JCPU	3	binary	ACTJTIME (in JCT)	Job CPU time, in hundredths of a second[4]
116	74	SMF5ACTF	1	binary	ACTJNFLD (in JCT)	Number of accounting fields
117	75	SMF5JSAF	variable	EBCDIC	JOB statement	Accounting fields. Each entry for an accounting field contains the length of the field (one byte, binary) followed by the field (EBCDIC). A zero indicates an omitted field.

[4] If TIME=1440 is specified on the JOB statement, all timing for the job is eliminated. If TIME=1440 is specified on an EXEC statement and a job time limit is not specified on the JOB statement, all timing for that job step is eliminated. For more information, see the IEFUTL exit.

CPU time may not 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, CPU architecture (such as storage buffering), cycle stealing with integrated channels, and queue searching (such an enqueue).

Record Type 6 – Output Writer

Record type 6 is written by IEFOSC4A (a subroutine called by IEFOSC01, IEFOSC02, IEFOSC05, and IEFOSC09) when the writer has finished processing a SYSOUT class, or form within a class, for a job. It is also written if the writer is held while writing a SYSOUT data set. If two or more forms are used within a class, one type 6 record is produced for each form. Its length is 99 bytes.

For output going to an IBM 3800, this record is written once for every data set. For data sets not going to an IBM 3800, fields SMF6CPS through SMF6BID are meaningless.

This record identifies the output writer by SYSOUT class and form number, and identifies the job by job log identification and user identification. It also contains information on the output writer activity such as the number of logical records processed, number of data sets processed, writer start and end times, and input/output status indicator. For output going to an IBM 3800, it contains information on how the output was processed.

Note: If an external writer or user supplied writer is used, an incomplete record type 6 is produced for each data set (if the writing of records is allowed). The number of logical records (offset 47) and the I/O status indicator (offset 51) fields are zero.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF6FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF6RTY	1	binary	internal (Set by IEFVMD)	Record type
2	2	SMF6TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF6DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF6SID	4	EBCDIC	SMCASID (SVC 83)	System identification (taken from SID parameter)
14	E	SMF6JBN	8	EBCDIC	EMRJOB (Set by IEFOSC4A)	Job name[1]
22	16	SMF6RST	4	binary	JMRENTRY (Set by IEFOSC4A)	Time, in hundredths of a second, reader recognized the JOB card for this job[1]
26	1A	SMF6RSD	4	packed	JMREDATE (Set by IEFOSC4A)	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign[1]

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

(Continued)

Offsets		Name	Length	Format	Source	Description
30	1E	SMF6UIF	8	EBCDIC	DSBUSEID or JMRUSEID (Set by IEFOSC4A)	User identification (taken from common exit parameter area). In the case of a forms change, the information in the DSBUSEID field is used; the DSBUSEID field is set for each job step from the JMRUSEID field *before* the IEFACTRT step/job termination exit is taken.
38	26	SMF6OWC	1	EBCDIC	DSBCLAS (Set by IEFOSC4A)	SYSOUT class
39	27	SMF6WST	4	binary	SVC 11 (Set by IEFOSC4A)	Time of SYSOUT start, in hundredths of a second
43	2B	SMF6WSD	4	packed	SVC 11 (Set by IEFOSC4A)	Date of SYSOUT start, in the form 00YYDDDF where F is the sign
47	2F	SMF6NLR	4	binary	Calculated by IEFOSC01 or IEFOSC02; moved to buffer by IEFOSC4A	Number of logical records written by the writer, by form number and class. This field includes job-related records if the JOBLOG=YES parameter is specified in the JES macro at system generation.
51	33	SMF6IOE	1	binary	Set during writer processing in IEFOSC4A	I/O status indicator *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
52	34	SMF6NDS	1	binary	internal (Set by IEFOSC01 or IEFOSC02)	Number of data sets completely written by the writer and included in this record. If a data set is held, it will be counted upon completion. For output going to an IBM 3800, this value is normally 1, because the record is written after each data set. The record is not written for an empty data set, but each data set is counted. The record for the non-empty data set following any empty data sets (or the last record for the job) reflects the empty data sets in this field.
53	35	SMF6FMN	4	EBCDIC	DSBFOR (Set by IEFOSC4A)	Form number
57	39	SMF6QID	8	EBCDIC	JMRRESID (Set by IEFVMD)	User's logon identifier. Non-terminal oriented jobs have an identifier of 'CENTRAL'; terminal-oriented RES jobs use the contents of the JMRRESID field for the terminal to which output was routed. Terminal-oriented CRJE jobs use the CRJE user ID for the terminal to which output was routed.
65	41	SMF6CPS	8	EBCDIC	Calculated by IEFOSC5B	Copies distribution
73	49	SMF6CHR	16	EBCDIC	SPPXLAT1 (Set by IEFOSC4A)	Translate table names from CHARS parameter
89	59	SMF6MID	4	EBCDIC	SPPMODPT (Set by IEFOSC4A)	Copy modification module name
93	5C	SMF6FLI	4	EBCDIC	SPPIMAGE (Set by IEFOSC4A)	Flash overlay name
97	61	SMF6FLC	1	binary	Calculated by IEFOSC5B	Number of copies flashed
98	62	SMF6BID	1	binary	Set by IEFOSC4A SPPBURST ACBOPTJ	Flag Byte *Bit* *Meaning When Set* 0 BTSS was used for output 1 OPTCD=J was used for output 2-7 Reserved

Record Type 7 – Data Lost

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

This record contains a count of the SMF records that were 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 offset 2.)

Notes:

1. In record types 4 and 5, the time stamp reflects the time that the job step or job ended instead of the time that the record was moved to the SMF buffer. Therefore, it is possible for these records, which follow the type 7 record in the SMF data set, to have a time stamp earlier than that of the type 7 record.
2. Even if MAN=USER is specified in the SMFxxxxx member of SYS1.PARMLIB, a type 7 record is built and written to the SMF data set.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF7FLG	1	binary	IEESMFWT	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF7RTY	1	binary	internal	Record type
2	2	SMF7TME	4	binary	SVC 11 (Set by IEESMFWT)	Time, in hundredths of a second, record was built in SMF buffer
6	6	SMF7DTE	4	packed	SVC 11 (Set by IEESMFWT)	Date record was built in SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF7SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF7NRO	2	binary	SMCADSCT	Number of SMF records lost because no SMF data sets were available for recording
16	10	SMF7STM	4	binary	SMCADSTM (Set by IEESMFWT)	Start time, in hundredths of a second, of period during which no SMF data sets were available for recording
20	14	SMF7STD	4	packed	SMCADSTM (Set by IEESMFWT)	Start date of period during which no SMF data sets were available for recording, in the form 00YYDDDF where F is the sign

Record Type 8 – I/O Configuration

Record type 8 is written by IEESMFI2 after the IPL of the system is completed and the SET DATE operator command is issued. This record identifies each device that is online at IPL by device class, unit type, channel address, and unit address. Its length is 16 bytes plus four bytes for each device online at IPL.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF8FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF8RTY	1	binary	internal	Record type
2	2	SMF8TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF8DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF8SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF8LENN	2	binary	internal	Length of rest of record including this field
For each online device, there is a four-byte entry with the following format:						
+0		SMF8IODV	1	binary	UCBTBYT3	Device class
+1			1	binary	UCBTBYT4	Unit type
+2			1	binary	UCBCHA	Channel address. (A mass storage volume is indicated when the high-order bit of this field is on.)
+3			1	binary	UCBUA	Unit address

Record Type 9 – VARY ONLINE

Record type 9 is written by ICBMSG00 (for the 3850 Mass Storage Control) and IEE2303D when a VARY ONLINE command is processed. This record identifies the device being added to the configuration by device class, unit type, channel address, and unit address. Its length is 16 bytes plus four bytes for each device varied online.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF9FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF9RTY	1	binary	internal	Record type
2	2	SMF9TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF9DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF9SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF9LENN	2	binary	internal	Length of rest of record including this field
For each device varied online, there is a four-byte entry with the following format:						
+0		SMF9DVAD	1	binary	UCBTBYT3	Device class
+1			1	binary	UCBTBYT4	Unit type
+2			1	binary	UCBCHA	Channel address. (A mass storage volume is indicated when the high-order bit of this field is on.)
+3			1	binary	UCBUA	Unit address

Record Type 10 – Allocation Recovery

Record type 10 is written by IEFXJIMP after a successful device allocation recovery. Its length is 40 bytes plus four bytes for each device that is made available.

This record identifies the device that is made available by device class, unit type, channel address, and unit address. It identifies the job requiring the allocation by job log identification and user identification.

Note: This record is not produced if the operator cancels the job instead of attempting recovery.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF10FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF10RTY	1	binary	internal	Record type
2	2	SMF10TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF10DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF10SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF10JBN	8	EBCDIC	JMRJOB	Job name. This field contains blanks if allocation recovery is for a system task [1]
22	16	SMF10RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job. This field equals zero if allocation recovery is for a system task [1]
26	1A	SMF10RSD	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign. This field equals zero if allocation recovery is for a system task [1]
30	1E	SMF10UIF	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)
38	26	SMF10LN	2	binary	internal	Length of rest of record including this field
For each device made available, there is a four-byte entry with the following format:						
+0		SMF10DEV	1	binary	UCBTBYT3	Device class
+1			1	binary	UCBTBYT4	Unit type
+2			1	binary	UCBCHA	Channel address. (A mass storage volume is indicated when the high-order bit of this field is on.)
+3			1	binary	UCBUA	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 identification.

Record Type 11 – VARY OFFLINE

Record type 11 is written when a VARY OFFLINE command is processed. It is written by ICBMSG00 (for the 3850 Mass Storage Control), IEFXCSSS, and IEFZHMSG. This record identifies the device being removed from the configuration by device class, unit type, channel address, and unit address. Its length is 16 bytes plus four bytes for each device varied offline.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF11FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF11RTY	1	binary	internal	Record type
2	2	SMF11TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF11DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF11SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF11LN	2	binary	internal	Length of rest of record including this field
For each device varied offline, there is a four-byte entry with the following format:						
+0		SMF11DEV	1	binary	UCBTBYT3	Device class
+1			1	binary	UCBTBYT4	Unit type
+2			1	binary	UCBCHA	Channel address. (A mass storage volume is indicated when the high-order bit of this field is on.)
+3			1	binary	UCBUA	Unit address

Record Type 12 – End-of-Day

Record type 12 is written by IEE1403D when a HALT EOD or SWITCH SMF command is processed. This record contains the system wait time and paging statistics accumulated between the expiration time recorded in the last record type 1 or 12, and the time the HALT or SWITCH command was issued. Its length is 34 bytes.

Note: If the system continues running after the HALT EOD or SWITCH SMF command is issued, the next record type 1 contains the wait time accumulated from the HALT or SWITCH command to the expiration of the next ten-minute interval. For more information about the collection of system wait time and paging statistics, see record type 1.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF12FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-6 Reserved 7 VS1
1	1	SMF12RTY	1	binary	internal	Record type
2	2	SMF12TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF12DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF12SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF12SWT	4	binary	SMCAWAIT	System wait time, in hundredths of a second, since the expiration time recorded in the last record type 1
18	12	SMF12TEX	4	binary	SVC 11 (Set by IEE1403D)	Expiration time, in hundredths of a second, of the collection period whose statistics are reported in this record
22	16	SMF12PGI	4	binary	SMCAPGIN	Number of page-ins for the entire system during the interval
26	1A	SMF12PGO	4	binary	SMCAPGOT	Number of page-outs for the entire system during the interval
30	1E	SMF12PGR	4	binary	SMCAPGRL	Number of page reclaims for the entire system during the interval

Record Type 13 – Dynamic Storage Configuration

Record type 13 is written by IEEDFINA at IPL and after each DEFINE operator command is processed. This record identifies each active problem program partition by the partition number, storage assigned to the partition, number of job classes, and job classes. Its length is 16 bytes plus 22 bytes for each partition entry.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF13FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-6 Reserved 7 VS1
1	1	SMF13RTY	1	binary	internal	Record type
2	2	SMF13TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF13DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF13SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF13LTH	2	binary	internal	Length of rest of record, including this field
For each active problem program partition, there is a 22-byte entry with the following format:						
+0		SMF13PN	1	binary	(TCBIBF–TCBID) for P0	Partition number
+1		SMF13SH0	2	binary	(BBXPTHIL – BBXPTLOL)/1024	Storage assigned to the partition, in 1K blocks
+3			2	binary		Reserved
+5		SMF13NJC	1	binary	internal	Number of job classes
+6		SMF13JC	16	EBCDIC	SD33GRP (in PIB)	Job classes specified by A-Z and 0-9; system task partitions are specified by *. The job classes are in their specified order, adjusted right, and padded with blanks.

Record Type 14 – INPUT or RDBACK Data Set Activity

Record type 14 is written for non-VSAM direct access or tape data sets that are defined by DD statements and opened for INPUT or RDBACK processing by problem programs. It is written by IFG0202H and IFG0202I when a data set, as described above, is closed or processed by EOV. Its length varies from 288 to 6,412 bytes, depending upon the number of volumes for the data set.

This record contains information (associated with both the access method used and the type of data set used) from the TIOT, JFCB, DCB, DEB, and UCB data areas. For more information about these data areas, see *OS/VS1 System Data Areas.*

Notes:

1. 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.
2. When opening generation data group (GDG) data sets using only one DD statement, the DD name is recorded only in the first type 14 record. The other type 14 records generated contain blanks in the DD name field.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF14FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF14RTY	1	binary	internal	Record type
2	2	SMF14TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF14DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF14SID	4	EBCDIC	JMRCPUID	System identification (taken from SID parameter)
14	E	SMF14JBN	8	EBCDIC	JMRJOB	Job name[1]
22	16	SMF14RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job [1]
26	1A	SMF14RSD	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign [1]
30	1E	SMF14UID	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)
38	26	SMF14RIN	2	binary		Record and data set indicator *Bit* *Meaning When Set* 0 Reserved
		SMF14EOV			internal	1 Record written by EOV (Register 14=0 if CLOSE; register 14=4 if EOV)
		SMF14DAD			UCBTYP	2 DASD device
		SMF14TDS			JFCBDSNM	3 Temporary data set. (A data set is temporary if it has a system-generated name, is created within a job or job step, and exists only for the duration of that job or job step.)
		SMF14DDA			DCBDSORG	4 DCBDSORG=DA. (The data set organization being used is direct access.)
		SMF14IS			DCBDSORG	5 DCBSORG=IS and DCBMACRF not EXCP. (The data set organization being used is indexed sequential and the EXCP access method is not being used.)
		SMF14JIS			JFCDSORG (in JFCB)	6 JFCDSORG=IS. (The data set organization being used is indexed sequential.) 7-15 Reserved

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

(Continued)

Offsets		Name	Length	Format	Source	Description
Section Sizes:						
40	28	SMF14SDC	1	binary	internal	Size of DCB/DEB section. This field equals 24.
41	29	SMF14NUC	1	binary	internal	Number of UCB sections. There is always one UCB section for each UCB currently processing except for ISAM and BPAM-concatenated data sets. For ISAM data sets, this field is calculated as: one for the index extent, one per volume for primary extents, and one for the overflow extent. For BPAM-concatenated data sets, there is one UCB section for each data set in the concatenated data set.
42	2A	SMF14SUC	1	binary	internal	Size of each UCB section. This field equals 24.
43	2B	SMF14SET	1	binary	internal	Size of ISAM extension section. This field equals 28 (or 0 if there are no ISAM data sets).
44	2C	SMF14RV1	4	binary		Reserved
TIOT Section — a portion of the TIOT, including:						
48	30	SMFTIOE1	1	binary	TIOELNGH	Length, in bytes, of the DD entry (including all device entries)
49	31	SMFTIOE2	1	binary	TIOESTTA	Status indicator. This field indicates the tape label processing to be performed; whether unallocating, rewinding, or unloading tape data sets is required; and whether this is the first DD entry for a split cylinder.
50	32	SMFTIOE3	1	binary	TIOEWTCT	Number of devices requested for this data set during allocation
51	33	SMFTIOE4	1	binary	TIOELINK	During allocation, this field indicates a link to the appropriate prime split, unit affinity, volume affinity, or suballocate TIOT entry. After allocation, it is a data set and device indicator.
52	34	SMFTIOE5	8	EBCDIC	TIOEDDNM	DDname
60	3C	SMFTIOE6	3	binary	TIOEJFCB	Relative track address (TTR) of the JFCB. During allocation, this field contains the TTR of the SIOT.
63	3F	SMFTIOE7	1	binary	TIOESTTC	Status indicator during allocation only and set to zeros at the end of allocation. This field indicates whether the unit affinity, volume affinity, and suballocate TIOT entry are primary or secondary.
JFCB Section:						
64	40	SMFJFCB1	176	binary	JFCB	The JFCB, excluding JFCB extensions
DCB/DEB Section — portions of the DCB and DEB, including:						
240	F0	SMFDCBOR	2	binary	DCBDSORG	Data set organization being used
242	F2	SMFDCBRF	1	binary	DCBRECFM	Record format
243	F3	SMFDCBMF	2	binary	DCBMACRF	Type of I/O macro instruction and options
245	F5	SMFDCBFL	1	binary	DCBOFLGS	Indicator used by the OPEN routine such as the type of the last I/O operation, and the return from the user's exit.
246	F6	SMFDCBOP	1	binary	DCBOPTCD	Option codes used by access-method interfaces
247	F7	SMF14RV2	1	binary		Reserved
248	F8	SMFDEBFL	1	binary	DEBOFLGS	Data set and device status indicator. This field indicates whether a data set is modified, new or old, and shows the status of DASD.
249	F9	SMFDEBOP	1	binary	DEBOPATB	Indicator showing both the method of I/O processing and the disposition that is to be performed when an end-of-volume (EOV) condition occurs.
250	FA	SMFDEBVL	2	binary	DEBVLSEQ	Volume sequence number. For direct access, the sequence number is relative to the first volume of the data set. For tape, the sequence number is relative to the first volume processed.

(Continued)

Offsets	Name	Length	Format	Source	Description
For each DCB/DEB tape extension, there is a 12-byte entry with the following format:					
+0	SMFDCBBL	4	binary	DCBBLKCT	Block count for each volume
+4	SMFDSSNO	6	binary	UCBSQC	Data set serial number
+10	SMF14RV3	2	binary		Reserved
The following 12 bytes apply to the DCB/DEB DASD extension:					
+0	SMF14NTU	4	binary	DCBFDAD	Relative track and concatenation number (TTRN) of the last record processed for a physical sequential or partitioned data set. N is always zero except for BPAM-concatenated data sets. If the last operation was a read (DCBOFLGS bit 0 is off) the TTR is: (1) zero, (2) the start of a BPAM member if there are two or more IOBs, or (3) the last write of the data set if there is one IOB. This is true because the access method maintains DCBFDAD while reading only if RECFM=FS, RECFM=FBS, or only one IOB (BUFNO, NCP) is used.
+4	SMF14NTR	4	binary	TCTTKRLD	Number of tracks released by the DADSM routine
+8	SMF14NER	1	binary	TCTEXRLD	Number of extents released by the DADSM routine
+9	SMF14RV4	3	binary		Reserved
UCB Section – a portion of the UCB (see offsets 41 and 42), including:					
+0	SMFUCBCH	1	binary	UCBCHA	Channel address. (A mass storage volume is indicated when the high-order bit of this field is on.)
+1	SMFUCBUA	1	binary	UCBUA	Unit address
+2	SMFSRTEV	6	binary	UCBVOLI	Volume serial number
+8	SMFUCBTY	4	binary	UCBTYP	Unit type
+12	SMFSRTES	1	binary	UCBSTAB	DASD volume status indicator. This field indicates whether this DASD volume is a private, public, storage, or control volume.
+13	SMF14NEX	1	binary	internal (in DEB)	Number of extents
+14	SMF14RV5	2	binary		Reserved
+16	SMFEXCP	4	binary	TCTDCTR	EXCP count for entire step. Note that if a data set is opened and closed twice during a single step, the count in the second type 14 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 record type 4. For more information about EXCP count, see record type 4.)
For each UCB tape extension, there is a four-byte entry with the following format:					
+0	SMFSRTEF	2	binary	UCBFSCT	Data set sequence count
+2	SMFSRTEQ	2	binary	UCBFSEQ	Data set sequence number
The following four bytes apply to the UCB DASD extension:					
+0	SMF14NTA	4	binary	DEBNMTRK for all extents	Number of tracks allocated on the device
ISAM Extension Section (DCBDSORG=IS and DCBMACRF not EXCP):					
+0	SMF14RV6	2	binary		Reserved
+2	SMFDCBMA	1	binary	DCBMAC	Extension of I/O macro instruction field (DCBMACRF) for ISAM
+3	SMFDCBNL	1	binary	DCBNLEV	Number of index levels
+4	SMFDCBR3	4	binary	DCBRORG3	For each use of the data set, number of read or write accesses to an overflow record which is not first in a chain of such records
+8	SMFDCBNR	4	binary	DCBNREC	Number of logical records in the prime data area
+12	SMFDCBR2	2	binary	DCBRORG2	Number of tracks (whole or partial) remaining in the overflow area

(Continued)

Offsets	Name	Length	Format	Source	Description
ISAM Extension Section: (Continued)					
+14	SMFDCBNO	2	binary	DCBNOREC	Number of logical records in the overflow area
+16	SMFDCBR1	2	binary	DCBRORG1	Number of cylinder overflow areas that are full
+18	SMF14RV7	1	binary		Reserved
+19	SMFDEBNI	1	binary	DEBNIEE	Number of extents in the independent index area
+20	SMFDEBNP	1	binary	DEBNPEE	Number of extents in the prime data area
+21	SMFDEBNO	1	binary	DEBNOEE	Number of extents in the independent overflow area
+22	SMFNCYLS	2	binary	internal (in DEB)	Number of cylinders in the independent index area
+24	SMFNPCYL	2	binary	internal (in DEB)	Number of cylinders in the prime data area
+26	SMFNOCYL	2	binary	internal (in DEB)	Number of cylinders in the independent overflow area

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

Record type 15 is written for non-VSAM direct access or tape data sets that are defined by DD statements and opened for OUTPUT, UPDAT, INOUT, or OUTIN processing by problem programs. It is written by IFG0202H and IFG0202I when a data set, as described above, is closed or processed by EOV. Its length varies from 288 to 6,412 bytes, depending upon the number of volumes for the data set.

This record contains information (associated with both the access method used and the type of data set used) from the TIOT, JFCB, DCB, DEB, and UCB data areas. For more information about these data areas, see *OS/VS1 System Data Areas*.

Note: 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 provided in record type 6.

The format for this record is the same as the format for record type 14.

Record Type 17 – Scratch Data Set Status

Record type 17 is written by IGG0290D when a non-VSAM data set that is defined by a DD statement (either explicitly or implicitly) is scratched. (When a DD statement defines a volume, all the data sets on that volume are implicitly defined.) This record contains the data set name, number of volumes, and volume serial numbers. Its length is 88 bytes plus eight bytes for each volume.

Notes:

1. If REC=0 was specified, record type 17 is generated for non-temporary data sets only. If REC=2 was specified, this record is generated for both temporary and non-temporary data sets.
2. The length of this record can vary from 96 to 2,136 bytes, depending upon the number of volumes for the data set.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF17FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF17RTY	1	binary	internal	Record type
2	2	SMF17TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF17DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF17SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF17JBN	8	EBCDIC	JMRJOB	Job name[1]
22	16	SMF17RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job [1]
26	1A	SMF17RSD	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign [1]
30	1E	SMF17UID	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)
38	26	SMF17RIN	2	binary		Reserved
40	28	SMF17DSN	44	EBCDIC	user's parameter list	Data set name
84	54	SMF17RV1	3	binary		Reserved
87	57	SMF17NVL	1	binary	user's parameter list	Number of volumes
For each volume, there is a eight-byte entry with the following format:						
+0		SMF17RV2	2	binary		Reserved
+2		SMF17FVL	6	EBCDIC	user's parameter list	Volume serial number

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

Record Type 18 – Rename Data Set Status

Record type 18 is written by IGG03001 when a non-VSAM data set that is defined by a DD statement (either explicitly or implicitly) is renamed. (When a DD statement defines a volume, all the data sets on that volume are implicitly defined.) This record contains the old data set name, new data set name, number of volumes, and volume serial numbers. Its length is 132 bytes plus eight bytes for each volume.

Note: The length of this record can vary from 140 to 2,180 bytes, depending upon the number of volumes for the data set.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF18FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF18RTY	1	binary	internal	Record type
2	2	SMF18TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF18DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF18SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF18JBN	8	EBCDIC	JMRJOB	Job name[1]
22	16	SMF18RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job [1]
26	1A	SMF18RSD	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign [1]
30	1E	SMF18UID	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)
38	26	SMF18RIN	2	binary		Reserved
40	28	SMF18ODS	44	EBCDIC	user's parameter list	Old data set name
84	54	SMF18NDS	44	EBCDIC	user's parameter list	New data set name
128	80	SMF18RV1	3	binary		Reserved
131	83	SMF18NVL	1	binary	user's parameter list	Number of volumes
For each volume, there is an eight-byte entry with the following format:						
+0		SMF18RV2	2	binary		Reserved
+2		SMF18FVL	6	EBCDIC	user's parameter list	Volume serial number

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

Record Type 19 – Direct Access Volume

Record type 19 is written by IGC0107H (1) for each direct access device that is online at IPL, (2) when a HALT EOD or SWITCH SMF command is processed, and (3) when a volume that is defined by a DD statement is demounted. Its length is 64 bytes.

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

Notes:

1. Record type 19 is not produced for DOS volumes used under the VS1 operating system.
2. In order to determine the latest status of a shared file, the CPU clocks must be synchronized.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF19FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF19RTY	1	binary	internal	Record type
2	2	SMF19TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF19DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF19SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF19RV1	2	binary		Reserved
16	10	SMF19VOL	6	EBCDIC	volume label[1] (VOLSERNO)	Volume serial number
22	16	SMF19OID	10	EBCDIC	volume label[1] (VOLOWNER)	Owner identification of direct access volume
32	20	SMF19DEV	4	binary	UCBTYP	Unit type
36	24	SMF19VTC	5	binary	volume label[1] (VOLVTOC)	VTOC address
41	29	SMF19VTI	1	binary	DS4VTOCI (in DSCB4) DS4DIRF DS4DICVT	VTOC indicator *Bit* *Meaning When Set* 0-4 Reserved 5 Possible VTOC error 6 VTOC error has been fixed 7 Reserved
42	2A	SMF19NDS	2	binary	internal	Number of DSCBs, calculated as: number of DSCBs per track times number of tracks in VTOC
44	2C	SMF19DSR	2	binary	DS4DSREC (in DSCB4)	Number of DSCB0s, that is, number of available DSCBs
46	2E	SMF19NAT	2	binary	DS4NOATK (in DSCB4)	Number of unused alternate tracks
48	30	SMF19SPC	2	binary	internal (DSCB5)	Number of unallocated cylinders
50	32		2	binary	internal (DSCB5)	Number of unallocated tracks
52	34	SMF19LEX	2	binary	internal (DSCB5)	Number of cylinders in the largest unallocated extent
54	36		2	binary	internal (DSCB5)	Number of tracks in the largest unallocated extent
56	38	SMF19NUE	2	binary	internal (DSCB5)	Number of unallocated extents
58	3A	SMF19RV2	2	binary		Reserved
60	3C	SMF19CUU	1	binary	UCBCHA	Channel address. (A mass storage volume is indicated when the high-order bit of this field is on.)
61	3D		1	binary	UCBUA	Unit address
62	3E	SMF19IND	2	binary	internal	Module identification or drive number indicating physical identity of devices having moveable address plugs. This field is taken from bits 2-7 of sense byte 4 for these devices. (See the component descriptions of these devices for the meaning of sense byte 4.)

[1]The volume label for the direct access device is record 3 of cylinder 0 of track 0.

Record Type 20 – Job Initiation

Record type 20 is written by IEFSMFIE at job initiation. This record contains the job log identification, user identification, programmer's name, number of accounting fields on the JOB statement, and accounting fields. Its length is 61 bytes plus the length of the JOB statement accounting fields.

Note: If the IEFUIV user-written exit routine cancels a job, the SMF data set will contain a record of the job only if the IEFUIV routine writes it.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF20FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF20RTY	1	binary	internal	Record type
2	2	SMF20TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF20DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF20SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF20JBN	8	EBCDIC	JMRJOB	Job name[1]
22	16	SMF20RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job[1]
26	1A	SMF20RSD	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign[1]
30	1E	SMF20UID	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)
38	26	SMF20RIN	2	binary		Reserved
40	28	SMF20PGM	20	EBCDIC	ACTPRGNM (in JCT)	Programmer's name
60	3C	SMF20NAF	1	binary	ACTJNFLD (in JCT)	Number of accounting fields
61	3D	SMF20ACT	variable	EBCDIC	JOB statement	Accounting fields. Each entry for an accounting field contains the length of the field (one byte, binary) followed by the field (EBCDIC). A zero indicates an omitted field.

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

Record Type 21 – Error Statistics by Volume

Record type 21 is written by IGC0009A when a user data set on magnetic tape is closed or processed by end-of-volume (EOV). Its length is 44 bytes.

This record contains the volume serial number, channel address, unit address, unit type, and tape density. It also contains the number of: temporary and permanent read and write errors, START I/Os, noise blocks, erase gaps, and cleaner actions.

Notes:

1. Record type 21 is written to the SMF data set only when ESV=SMF is specified in the SCHEDULR macro instruction at SYSGEN time.
2. The IFHSTATR utility program formats and prints the error-statistics-by-volume (ESV) information in this record. For a detailed description of this utility program, see *OS/VS1 Utilities.*

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF21FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF21RTY	1	binary	internal	Record type
2	2	SMF21TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF21DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF21SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF21LGH	2	binary	internal	Length of rest of record including this field. (This field is always 30.)
16	10	SMF21VOL	6	EBCDIC	UCBVOLI	Volume serial number
22	16	SMF21CA	1	binary	UCBCHA	Channel address
23	17		1	binary	UCBUA	Unit address
24	18	SMF21UCB	4	binary	UCBTYP	Unit type
28	1C	SMF21TR	1	binary	UCBTR	Number of temporary read errors
29	1D	SMF21TW	1	binary	UCBTW	Number of temporary write errors
30	1E	SMF21SIO	2	binary	UCBSIO	Number of START I/Os
32	20	SMF21PR	1	binary	UCBPR	Number of permanent read errors
33	21	SMF21PW	1	binary	UCBPW	Number of permanent write errors
34	22	SMF21NB	1	binary	UCBNB	Number of noise blocks
35	23	SMF21ERG	2	binary	UCBERG	Number of erase gaps
37	25	SMF21CLN	2	binary	UCBCLN	Number of cleaner actions
39	27	SMF21DEN	1	binary	DCBDEN	Tape density – 2400 and 3400 series magnetic tape units *Bits* *Meaning When Set* *Code* *7-Track* *9-Track* DCBMTDN0 6,7 0 200 BPI N/A DCBMTDN1 1,6,7 1 556 BPI N/A DCBMTDN2 0,6,7 2 800 BPI 800 BPI DCBMTDN3 0,1,6,7 3 N/A 1600 BPI DCBMTDN4 0,1,3,6,7 4 N/A 6250 BPI
40	28	SMF21BLS	2	binary	DCBBLKSI	Block size. This field is zero if DCBRECFM indicates variable or unblocked records, or if you are doing your own EXCP processing.
42	2A	SMF21RV0	2	binary		Reserved (not necessarily zero)

Record Type 22 – Configuration

Record type 22 is written (1) by ICBMSG00 after every IPL of the system, and (2) by IGC126 when a VARY ONLINE,S or VARY OFFLINE,S operator command changes the status of a mass storage device. This record describes the MSS units online at IPL, and indicates the subsystem identification (SSID) of each device that is varied online or offline. Its length is 18 bytes plus the length of the sections in the record.

Notes:

1. During system initialization, the Mass Storage Control (MSC) creates a configuration record (called Message 92) and puts it in a message buffer. After SMF is initialized, ICBMSG00 creates the "MSS IPL Configuration section" of record type 22 from the Message 92 in the MSC buffer.
2. By using the VARY ONLINE,S and VARY OFFLINE,S commands, the operator can modify the configuration. (The formats of these commands are described in *Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS.*) In a record type 22, a "VARY ONLINE,S section" or a "VARY OFFLINE,S section" is written for *each* unit varied by these commands.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF22FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF22RTY	1	binary	internal	Record type
2	2	SMF22TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF22DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF22SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF22IND	2	binary	ICBMSG00 IGC126 IGC126	Record creator indicator *Value* *Meaning* 4 MSS at IPL 5 VARY ONLINE,S 6 VARY OFFLINE,S
16	10	SMF22ECT	2	binary	internal	Number of sections following
MSS IPL Configuration Section:						
+0		SMF22RV4	1	binary		Reserved
+1		SMF22IID	1	binary	internal	MSS IPL configuration section identification. (This field is always 4.)
The following 40 bytes contain the bit pattern of MSS units online at IPL:						
+2		SMF22ION	1	binary		Data recording devices (DRDs) of MSF(0) *Bit* *SSID* 0 200 1 201 2 202 3 203 4 204 5 205 6 206 7 207

(Continued)

Offsets	Name	Length	Format	Source	Description
MSS IPL Configuration Section: (Continued)					
+3		1	binary		Data recording devices (DRDs) of MSF(1) *Bit* *SSID* 0 210 1 211 2 212 3 213 4 214 5 215 6 216 7 217
+4		2	binary		Reserved
+6		2	binary		Staging Adapters *Bit* *SSID* 0 800 1 810 2 820 3 830 ⋮ 15 8F0
+8		1	binary		Device recording controls (DRCs) of MSF(0) *Bit* *SSID* 0 400 1 401 2 402 3 403 Device recording controls (DRCs) of MSF(1) 4 410 5 411 6 412 7 413
+9		1	binary		Reserved
+10		28	binary		Staging spindles in staging data groups (SDGs) *SDG* *Bits* *SSIDs* 0 0-7 000-007 1 0-7 008-00F 2 0-7 010-017 3 0-7 018-01F ⋮ ⋮ ⋮ 27 0-7 0E8-0EF
+38		3	binary		Reserved
+41		1	binary		MSF indicator *Bit* *Meaning When Set* 0 MSF(0) with SSID of 101 1 MSF(0) with SSID of 102 2 MSF(1) with SSID of 111 3 MSF(1) with SSID of 112 4-7 Reserved
VARY ONLINE,S Section:					
+0	SMF22RV5	1	binary		Reserved
+1	SMF22NID	1	binary	internal	VARY ONLINE,S section identification. (This field is always 5.)
+2	SMF22RVA	1	binary		Reserved
+3	SMF22NSI	3	binary	VARY command	Subsystem identification of device

(Continued)

Offsets	Name	Length	Format	Source	Description
VARY OFFLINE,S Section:					
+0	SMF22RV6	1	binary		Reserved
+1	SMF22FID	1	binary	internal	VARY OFFLINE,S section identification. (This field is always 6.)
+2	SMF22RVB	1	binary		Reserved
+3	SMF22FSI	3	binary	VARY command	Subsystem identification of device

Record Type 43 – RTAM Start

Record type 43 is written by IFSINIT during RTAM initialization. This record contains the RTAM start procedure name, maximum number of readers and writers supported, number of line and logical port DCTs, and number of lines to start at this time. This record also identifies each line DCT created by line number, channel address, and unit address. Its length is 42 bytes plus six bytes for each line DCT entry.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF43FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-6 Reserved 7 VS1
1	1	SMF43RTY	1	binary	internal	Record type
2	2	SMF43TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF43DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF43SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF43SBS	2	binary	internal	Subsystem identification – X'0001' signifies RTAM
16	10		2	binary		Reserved
18	12	SMF43LRR	2	binary	internal	Length of rest of record, excluding this field
20	14	SMF43PNM	8	EBCDIC	QIDPROC	RTAM start procedure name
28	1C		8	binary		Reserved
36	24	SMF43RDR	1	binary	RTMRDR	Maximum number of readers supported
37	25	SMF43WRT	1	binary	RTMPTR + RTMPNCH	Maximum number of writers supported
38	26	SMF43TAB	1	binary	$NUMLNES (in IFSSETS)	Number of entries in LINE table
39	27	SMF43DCT	1	binary	NUMLNDCT + RTMPORTS	Number of line and logical port DCTs
40	28	SMF43ACT	1	binary	RTMPORTS	Number of logical port DCTs
41	29	SMF43STR	1	binary	Indicated by message IFS102I	Number of lines to start at this time
For each line DCT created, there is a six-byte entry with the following format:						
+0		SMF43LNM	3	EBCDIC	DCTDEVN	Line number
+3		SMF43UAD	3	EBCDIC	UCBNAME	Channel and unit address, in the form cuu where c is the channel and uu is the unit

Record Type 44 – RTAM Modify

Record type 44 is written by IFSCMD when a MODIFY RTAM operator command is issued. This record contains the RTAM start procedure name, type of modification, and number of lines modified at this time. It identifies each line modified by line number, channel address, and unit address; it identifies each user modified by user identification. Its length is 30 bytes plus either six bytes for each modified line entry or nine bytes for each modified user entry.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF44FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-6 Reserved 7 VS1
1	1	SMF44RTY	1	binary	internal	Record type
2	2	SMF44TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF44DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF44SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF44SBS	2	binary	internal	Subsystem identification – X'0001' signifies RTAM
16	10		2	binary		Reserved
18	12	SMF44LRR	2	binary	internal	Length of rest of record, excluding this field
20	14	SMF44PNM	8	EBCDIC	QIDPROC	RTAM start procedure name
28	1C	SMF44COD SMF44STR SMF44STP SMF44RST SMF44VEY SMF44VEN SMF44SPS SMF44USR SMF44RCH	1	binary	internal	MODIFY command type *Value* *Meaning* 1 START 2 STOP 3 RESTART 4 VTAM=YES (V=Y) 5 VTAM=NO (V=N) 6 STOP=S 7 STOP=userid 8 Change remote characteristics
29	1D	SMF44LMD	1	binary	internal	Number of lines modified at this time
For each modified line, there is a six-byte entry with the following format:						
+0		SMF44LNM	3	EBCDIC	DCTDEVN	Line number
+3		SMF44UAD	3	EBCDIC	UCBNAME	Channel and unit address, in the form cuu where c is the channel and uu is the unit
For each modified user, there is a nine-byte entry with the following format:						
+0		SMF44PRT	7	EBCDIC	QIDUSRID	User identification (taken from MODIFY command)
+7		SMF44RSV	1	EBCDIC		Reserved
+8		SMF44RCN	1	binary	DCTRCHAR	This field identifies the remote characteristics of the specified user after the MODIFY command is complete. This field is used only if the MODIFY command type is 8; if STOP=userid is used, this field is all zeros. *Bit* *Meaning When Set* 0 Compression (CPRES) is supported 1 Compaction (CPACT) is supported 2 Transparency (TRAN) is supported 3 Horizontal format control (HFC) is supported 4 Workstation is unattended (UNATT) 5-7 Reserved

Record Type 45 – RTAM Stop

Record type 45 is written by IFSCMD when a STOP RTAM operator command is issued and by IFSSTAE when the STOP processing is completed. This record contains the RTAM start procedure name, STOP status, and number of lines started and logical ports active when the STOP was received. Its length is 31 bytes.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF45FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-6 Reserved 7 VS1
1	1	SMF45RTY	1	binary	internal	Record type
2	2	SMF45TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF45DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF45SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF45SBS	2	binary	internal	Subsystem identification – X'0001' signifies RTAM
16	10		2	binary		Reserved
18	12	SMF45LRR	2	binary	internal	Length of rest of record, excluding this field
20	14	SMF45PNM	8	EBCDIC	QIDPROC	RTAM start procedure name
28	1C	SMF45STP SMF45BEG SMF45END	1	binary	internal	STOP command processing indicator *Value* *Meaning* 0 Stop begun 1 Stop ended
29	1D	SMF45USR	1	binary	RESLGCT2 (in RESCT)	Number of lines started when STOP command was received
30	1E	SMF45PAS	1	binary	internal	Number of logical ports active when STOP command was received

Record Type 47 – RTAM Logon

Record type 47 is written by IEELGON1 when a valid LOGON record is received by RTAM. This record contains the QID entry, passback area, and LOGON record. Its length is 152 bytes.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF47FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-6 Reserved 7 VS1
1	1	SMF47RTY	1	binary	internal	Record type
2	2	SMF47TME	4	binary	SVC 83	Time in hundredths of a second, record was moved to SMF buffer
6	6	SMF47DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF47SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF47SBS	2	binary	internal	Subsystem identification – X'0001' signifies RTAM
16	10		2	binary		Reserved
18	12	SMF47LRR	2	binary	internal	Length of rest of record, excluding this field
20	14	SMF47QID	48	binary	QID table	QID entry
68	44	SMF47PSB	4	binary	internal	Passback area – X'8000 0000' signifies a valid LOGON record
72	48	SMF47LGN	80	EBCDIC	Command buffer built by SVC 34	LOGON record

Record Type 48 – RTAM Logoff

Record type 48 is written by IFSLOGON when a LOGOFF record is received by RTAM. This record contains the QID entry; its length is 68 bytes.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF48FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-6 Reserved 7 VS1
1	1	SMF48RTY	1	binary	internal	Record type
2	2	SMF48TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF48DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF48SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF48SBS	2	binary	internal	Subsystem identification – X'0001' signifies RTAM
16	10		2	binary		Reserved
18	12	SMF48LRR	2	binary	internal	Length of rest of record, excluding this field
20	14	SMF48QID	48	binary	QID table	QID entry

Record Type 49 – RTAM Integrity

Record type 49 is written by IFSLOGON and IEELGON1 when an invalid LOGON record is received by RTAM. This record contains the QID entry, passback area and LOGON record. Its length is 152 bytes.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF49FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-6 Reserved 7 VS1
1	1	SMF49RTY	1	binary	internal	Record type
2	2	SMF49TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF49DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF49SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF49SBS	2	binary	internal	Subsystem identification – X'0001' signifies RTAM
16	10		2	binary		Reserved
18	12	SMF49LRR	2	binary	internal	Length of rest of record, excluding this field
20	14	SMF49QID	48	binary	QID table	QID entry
68	44	SMF49PSB	4	binary	internal	Passback area – X'FF00 0000' signifies an invalid LOGON record
72	48	SMF49LGN	80	EBCDIC	If IFSLOGON, from BSC TP buffer or SNA session parameter area. If IEELGON1, from command buffer built by SVC 34	LOGON record

Record Type 50 – ACF/VTAM Tuning Statistics

Record type 50 maps data useful in adjusting certain ACF/VTAM and NCP sysgen and START command variables to improve ACF/VTAM performance. Its length is 58 bytes. For additional information, refer to the *ACF/VTAM System Programmer's Guide* listed in the Preface.

Offsets		Name	Length	Format	Source	Description
0	0	SMF50FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF50RTY	1	binary	internal	Record type
2	2	SMF50TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF50DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF50SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF50NME	8	EBCDIC		Name of locally attached intelligent controller
22	16	SMF50DLR	4	binary		Number of TCBs needed for dump-load-restart operation[1]
26	1A	SMF50CWR	4	binary		Number of write channel programs
30	1E	SMF50CRD	4	binary	ISTZFADB	Number of read channel programs
34	22	SMF50ATN	4	binary	ISTZFAAB	Number of attentions received
38	26	SMF50ATR	4	binary	ISTZFA2B	Number of times ending status of read channel program included ATTN
42	2A	SMF50PUI	4	binary	ISTZFA4F ISTZFA4B	Number of PIUs inbound
46	2E	SMF50PUO	4	binary		Number of PIUs outbound
50	32	SMF50BUF	4	binary	ISTZFA4F ISTZFA4B	Number of buffers used in read channel programs
54	36	SMF50SLD	4	binary	ISTZFA3B	Number of times a locally attached intelligent controller enters slowdown

[1]With the exception of the dump-load-restart TCB count, all values are accumulated in the ICNCB. The counts represent activity since the last trace entry.

Record Type 51 – Subsystem Data

Record type 51 is for subsystem or user data. Invocation of the IFASMFR macro instruction for type 51 causes the assembly of a DSECT mapping the standard type 51 record header. The DSECT ends with a label that marks the beginning of the user's data area. The total record length is variable.

Offsets		Name	Length	Format	Source	Description
0	0	SMF51 FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF51 RTY	1	binary	internal	Record type
2	2	SMF51 TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF51 DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF51 SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF51 UID	4	EBCDIC	User	User interface identification
18	12	SMF51 USR	*		User	Beginning of user/subsystem segment. Total record length is variable.

Record Type 62 – VSAM Component or Cluster Opened

Record type 62 is written by IDA0192A and IDA0192S at the successful or unsuccessful opening of a VSAM component or cluster. Its length is 138 bytes plus ten 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 object is defined and the volumes on which the catalog and object are stored. It also identifies the job that issued the OPEN macro by job log identification and user identification.

Note: This record is not generated when a system task issues the OPEN macro.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF62FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF62RTY	1	binary	internal	Record type
2	2	SMF62TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF62DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF62SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF62JBN	8	EBCDIC	JMRJOB	Job name[1]
22	16	SMF62RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job[1]
26	1A	SMF62RSD	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign[1]
30	1E	SMF62UIF	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)
38	26	SMF62IND	4	binary	Set by IDA0192A and passed to IDA0192S in parameter list	Open status indicator *Bit* *Meaning When Set* 0 Component or cluster was successfully opened 1 Security violation, that is, invalid password 2-31 Reserved
42	2A	SMF62CNM	44	EBCDIC	CAXCNAM (in IGGCAXWA)	Name of the catalog in which the component or cluster is defined
86	56	SMF62CVS	6	EBCDIC	UCBVOLI	Volume serial number of the volume containing the catalog
92	5C	SMF62DNM	44	EBCDIC	JFCBDDNM	Name of the component or cluster being opened
136	88	SMF62VCT	2	binary	OPEN routine calculates from VMT entries	Number of online volumes containing the component or cluster. (This field is also the number of ten-byte fields that list the volumes.)
For each online volume, there is a ten-byte entry with the following format:						
+0		SMF62VSR	6	EBCDIC	VMTVLSER	Volume serial number of the volume containing the component or cluster
+6		SMF62DTY	4	binary	UCBTYP	Unit type of the volume containing the component or cluster

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

Record Type 63 – VSAM Entry Defined

Record type 63 is written by IGG0CLBV when a VSAM catalog entry (a component, cluster, catalog, alternate index, path, or non-VSAM data set) (1) is defined by the DEFINE Access Method Service command, (2) is altered with new space allocation information (that is, when the VSAM end-of-volume (EOV) routine extends the entry's object), and (3) is changed by the ALTER Access Method Services command. One record type 63 is written for each newly created or altered entry. Its length is 132 bytes plus the length of the catalog records required to describe the entry.

Record type 63 identifies the catalog in which the object is defined, gives the catalog record for the newly defined object, and, for an alteration, gives the parts of the old catalog record before they were altered. It identifies the job by job log identification and user identification.

Note: The length of this record can be from 1000 to 4000 bytes or more, depending upon the sizes of the new and old catalog records (offsets 40 and 42, respectively). If this record is to be written to the SMF data set, be sure to include the sizes of these catalog records when estimating the additional storage required for the SMF buffer and the SMF data sets.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF63FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF63RTY	1	binary	internal	Record type
2	2	SMF63TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF63DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF63SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF63JBN	8	EBCDIC	JMRJOB	Job name[1]
22	16	SMF63RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job[1]
26	1A	SMF63RSD	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign[1]
30	1E	SMF63UIF	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)[1]
38	26	SMF63FDT	1	binary	VSAM catalog record entry type	Record creator/entry type indicator *Bit* *Meaning When Set* 0 New definition 1 Altered definition 2-5 Reserved 6 Path defined or altered 7 Alternate index defined or altered

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

(Continued)

Offsets		Name	Length	Format	Source	Description
39	27	SMF63TYP	1	binary	VSAM catalog record entry type	Entry type indicator *Bit* *Meaning When Set* 0 VSAM cluster 1 VSAM data component 2 VSAM index component 3 VSAM catalog 4 Non-VSAM data set 5-7 Reserved
40	28	SMF63NSZ	2	binary	internal	Size of new catalog record. (Be sure to include the contents of this field when estimating the additional storage required by SMF.)[2]
42	2A	SMF63OSZ	2	binary	internal	Size of old catalog record. This field contains the size of the old records before they were altered. (Be sure to include the contents of this field when estimating the additional storage required by SMF.)[2]
44	2C	SMF63CNM	44	EBCDIC	CAXCNAM (in IGGCAXWA)	Name of catalog in which the entry is defined
88	58	SMF63ENM	44	EBCDIC	Name field of VSAM catalog record	Entry name
132	84	SMF63NCR	variable	binary	VSAM catalog entry records	New catalog record followed by old catalog record.[2] For the new catalog record, the complete new entry is recorded. For the old catalog record, this field contains only those old records that were altered; it shows what these records were before they were altered.

[2] A VSAM catalog record is contained in one or more physical catalog records. Offsets 40 and 42 are the sums of the sizes of the physical catalog records that constitute the total logical VSAM catalog record. Catalog record formats are shown in the *VS1 VSAM Logic* manual cited in the Preface.

Record Type 64 – VSAM Component or Cluster Status

Record type 64 is written when (1) a VSAM component or cluster is closed, (2) VSAM must switch to another volume to continue to read or write, and (3) there is no more space available for VSAM to continue processing. If a cluster is closed, one record is written for each component in the cluster. This record is written by IDA0192S, IDA0200B, IDA0231B, and IDA0557A. Its length is 250 bytes plus 26 bytes for each extent.

Record type 64 indicates why the record was created (a component was closed, another volume was switched to, or no additional space was available). It describes the device and volume(s) on which the object is stored, and gives the extents of the object on the volume(s). It also gives statistics about various processing events that have occurred since the object was defined, such as the number of records in the data component, the number of records that were inserted, and the number of control intervals that were split. This record identifies the job by job log identification and user identification.

Note: This record is not generated for system tasks.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF64FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF64RTY	1	binary	internal	Record type
2	2	SMF64TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF64DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF64SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF64JBN	8	EBCDIC	JMRJOB	Job name[1]
22	16	SMF64RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job[1]
26	1A	SMF64RSD	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign[1]
30	1E	SMF64UIF	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)
38	26	SMF64RIN	1	binary	internal	Situation indicator *Bit* *Meaning When Set* 0 Component closed 1 Volume switched 2 No space available 3-7 Reserved
39	27	SMF64DTY	1	binary	AMBTYPE	Indicator of component being processed *Bit* *Meaning When Set* 0 Data component 1 Index component 2-7 Reserved
40	28	SMF64CNM	44	EBCDIC	CAXCNAM (in IGGCAXWA)	Name of the catalog in which the component is defined
84	54	SMF64DNM	44	EBCDIC	VSAM catalog (ENTNAME)	Name of the component being processed
128	80	SMF64NTR	2	binary	VSAM catalog (PRIMSPAC for primary allocation; SCONSPAC for secondary allocation)	Number of tracks that were requested but could not be allocated
130	82	SMF64CHR	4	binary	ARDHRBA (in ARDB)	Highest used relative byte address (RBA) of the component
134	86	SMF64ESL	2	binary	DEBNMEXT * 26	Length of extent entry portion of record, excluding this field
For each extent, there is a 26-byte entry with the following format:						
+0		SMF64FCC	4	binary	DEBSTRCC	Beginning cylinder and track, in the form CCHH where CC is the cylinder number and HH is the track number
+4		SMF64TCC	4	binary	DEBENDCC	Ending cylinder and track, in the form CCHH where CC is the cylinder number and HH is the track number
+8		SMF64VSN	6	EBCDIC	UCBVOLI	Volume serial number of the volume containing the extent
+14		SMF64CUU	1	binary	UCBCHA	Channel address. (A mass storage volume is indicated when the high-order bit of this field is on.)
			1	binary	UCBUA	Unit address
+16		SMF64IND	2	binary	internal	Spindle identification

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

(Continued)

Offsets		Name	Length	Format	Source	Description
+18		SMF64UTY	4	binary	UCBTYP	Unit type
+22			4	binary		Reserved
Statistics Section:[2]						
Accumulative Statistics from Creation Until the Current OPEN:						
+0		SMF64SLN	4	binary	internal	Length of the statistics section, including this field
+4		SMF64NIL	4	binary	AMDNIL	Number of levels in the index
+8		SMF64NEX	4	binary	AMDNEXT	Number of extents
+12		SMF64NLR	4	binary	AMDNLR	Number of records in the component
+16		SMF64NDE	4	binary	AMDDELR	Number of records that were deleted from the component
+20		SMF64NIN	4	binary	AMDIREC	Number of records that were inserted into the component
+24		SMF64NUP	4	binary	AMDURR	Number of records that were updated in the component
+28		SMF64NRE	4	binary	AMDRETR	Number of records that were retrieved from the component
+32		SMF64NFS	4	binary	AMDFSCA	Number of unused control intervals in the component
+36		SMF64NCS	4	binary	AMDNCIS	Number of control intervals that were split in the component
+40		SMF64NAS	4	binary	AMDNCAS	Number of control areas that were split in the component
+44		SMF64NEP	4	binary	AMDEXCP	Number of EXCPs. (For more information about EXCP count, see record type 4.)
Change in Statistics from OPEN to Time of EOV and CLOSE:						
+48		SMF64DIL	4	binary	AMDNIL	Change in number of levels in the index
+52		SMF64DEX	4	binary	AMDNEXT	Change in number of extents
+56		SMF64DLR	4	binary	AMDNLR	Change in number of records in the component
+60		SMF64DDE	4	binary	AMDDELR	Change in number of records that were deleted from the component
+64		SMF64DIN	4	binary	AMDIREC	Change in number of records that were inserted into the component
+68		SMF64DUP	4	binary	AMDUPR	Change in number of records that were updated in the component
+72		SMF64DRE	4	binary	AMDRETR	Change in number of records that were retrieved from the component
+76		SMF64DFS	4	binary	AMDFSCA	Change in number of unused control intervals in the component. (This field may be negative.)
+80		SMF64DCS	4	binary	AMDNCIS	Change in number of control intervals that were split in the component
+84		SMF64DAS	4	binary	AMDNCAS	Change in number of control areas that were split in the component
+88		SMF64DEP	4	binary	AMDEXCP	Change in number of EXCPs. (For more information about EXCP count, see record type 4.)
Data Set Characteristics Section:						
+92		SMF64DBS	4	binary	LPMBLKSZ (in IDALPMB)	Physical block size
+96		SMF64DCI	4	binary	AMDCINV (in AMDSB)	Control interval size
+100		SMF64DLS	4	binary	AMDLRECL (in AMDSB)	Maximum logical record size
+104		SMF64DKL	2	binary	AMDKEYLN (in AMDSB)	Key length
+106		SMF64DDN	8	EBCDIC	TIOEDDNM	DD name. When the record is written for a VSAM catalog or catalog recovery area, this field may contain zeros. When the record is written for a volume switch or no space available condition, and the volume is associated with a concatenated TIOT entry, this field contains blanks.

[2]All the fields in this section are present and are taken from the AMDSB data area; inapplicable fields contain zeros.

Record Type 67 – VSAM Entry Deleted

Record type 67 is written by IGG0CLBV when a VSAM catalog entry (a component, cluster, catalog, alternate index, path, or non-VSAM data set) is deleted. A type 63 record is written for each entry affected by the DELETE Access Method Services command. For example, three records are written for an indexed cluster: one for the relationship between the components of the cluster, one for the data component, and one for the index component. Its length is 130 bytes plus the length of the catalog records required to describe the entry.

Record type 67 identifies the deleted entry, the VSAM catalog in which the entry was defined, and the deleted catalog records. It identifies the job by job log identification and user identification.

Note: The length of this record can be from 1000 to 4000 bytes or more, depending upon the sizes of the catalog records that describe the entry (offset 128). If this record is to be written to the SMF data set, be sure to include the sizes of these catalog records when estimating the additional storage required for the SMF buffer and the SMF data sets.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF67FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF67RTY	1	binary	internal	Record type
2	2	SMF67TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF67DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF67SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF67JBN	8	EBCDIC	JMRJOB	Job name[1]
22	16	SMF67RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job[1]
26	1A	SMF67RSD	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign[1]
30	1E	SMF67UIF	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)
38	26	SMF67FDT	1	binary	VSAM catalog record entry type	Record creator/entry type indicator *Bit* *Meaning When Set* 0 Uncataloged[2] 1 Scratched[2] 2-5 Reserved 6 Path deleted 7 Alternate index deleted

(Continued)

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

[2]Both indicators are set for VSAM component or cluster entries. For all other VSAM entries, only the uncataloged bit is set. For non-VSAM entries, the uncatalog bit is always set and the scratched bit is set if the physical non-VSAM space was deleted.

Offsets		Name	Length	Format	Source	Description
39	27	SMF67IOD	1	binary	VSAM catalog record entry type	Entry type indicator *Bit* *Meaning When Set* 0 VSAM cluster 1 VSAM data component[3] 2 VSAM index component[3] 3 VSAM catalog 4 Non-VSAM data set 5-7 Reserved
40	28	SMF67CNM	44	EBCDIC	CAXCNAM (in IGGCAXWA)	Name of catalog in which the entry was defined
84	54	SMF67DEN	44	EBCDIC	Name field of VSAM catalog record	Entry name
128	80	SMF67RSZ	2	binary	internal	Size of catalog record that defined the entry.[4] (Be sure to include the contents of this field when estimating the additional storage required by SMF.)
130	82	SMF67CRC	variable	binary	VSAM catalog entry records	Catalog record[4]

[3]A data or index component can only be deleted as one of the three catalog records deleted when a cluster is deleted.

[4]A VSAM catalog record is contained in one or more physical catalog records. Offset 128 is the sum of the sizes of the physical catalog records that constitute the total logical VSAM catalog record. Catalog record formats are shown in the *OS/VS1 VSAM Logic* manual cited in the Preface.

Record Type 68 – VSAM Entry Renamed

Record type 68 is written by IGG0CLBV when a VSAM catalog entry (a component, cluster, catalog, alternate index, path, or non-VSAM data set) is renamed using the ALTER Access Method Services command. This record identifies the VSAM catalog in which the object is defined, and gives the old and new names for the object. It also identifies the job by job log identification and user identification. Its length is 170 bytes.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF68FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF68RTY	1	binary	internal	Record type
2	2	SMF68TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF68DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF68SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF68JBN	8	EBCDIC	JMRJOB	Job name[1]
22	16	SMF68RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job[1]
26	1A	SMF68RSD	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign[1]
30	1E	SMF68UIF	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)[1]
38	26	SMF68CNM	44	EBCDIC	CAXCNAM (in IGGCAXWA)	Name of catalog in which the entry is defined
82	52	SMF68ONM	44	EBCDIC	Pointed to by CTGFVENT field in IEZCTGFV parameter list	Old name of the entry. (AMS obtains this name from the ALTER command.)
126	7E	SMF68NNM	44	EBCDIC	Pointed to by CTGNEWNM field in IEZCTGPL parameter list	New name of the entry. (AMS obtains this name from the ALTER command.)

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

Record Type 69 – VSAM Data Space Defined, Extended, or Deleted

Record type 69 is written by IGG0CLBV when a VSAM data space is defined, extended, or deleted using the DEFINE or DELETE Access Method Services commands. Record type 69 is not written when a catalog or a unique data set is defined or deleted. Its length is 102 bytes.

This record identifies the catalog in which the data space is defined and the volume on which it is (or was) allocated. It also gives the number of free data space extents and the amount of unallocated space on the affected volume after the definition, extension, or deletion. It identifies the job by job log identification and user identification.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF69FLG	1	binary	SVC 83	System indicator *Bit* *Meaning When Set* 0-5 Reserved 6 VS2 7 VS1
1	1	SMF69RTY	1	binary	internal	Record type
2	2	SMF69TME	4	binary	SVC 83	Time, in hundredths of a second, record was moved to SMF buffer
6	6	SMF69DTE	4	packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF69SID	4	EBCDIC	SMCASID	System identification (taken from SID parameter)
14	E	SMF69JBN	8	EBCDIC	JMRJOB	Job name[1]
22	16	SMF69RST	4	binary	JMRENTRY	Time, in hundredths of a second, reader recognized the JOB card for this job[1]
26	1A	SMF69RSD	4	packed	JMREDATE	Date reader recognized the JOB card for this job, in the form 00YYDDDF where F is the sign[1]
30	1E	SMF69UIF	8	EBCDIC	JMRUSEID	User identification (taken from common exit parameter area)[1]
38	26	SMF69CUU	1	binary	UCBCHA	Channel address
39	27		1	binary	UCBUA	Unit address
40	28	SMF69IND	2	binary	EXCP of sense data	Spindle identification
42	2A	SMF69NDS	2	binary	VSAM catalog volume entry	Number of free data space extents on the affected volume after the data space is defined, extended, or deleted
44	2C	SMF69NUC	2	binary	VSAM catalog volume entry	Number of unallocated cylinders in all of the data spaces on the volume
46	2E	SMF69NUT	2	binary	VSAM catalog volume entry	Number of unallocated tracks in all of the data spaces on the volume in addition to the number of unallocated cylinders
48	30	SMF69LNC	2	binary	VSAM catalog volume entry	Number of cylinders in the largest continuous unallocated area in any data space on the volume
50	32	SMF69LNT	2	binary	VSAM catalog volume entry	Number of tracks (in addition to the number of cylinders) in the largest continuous unallocated area in any data space on the volume
52	34	SMF69CNM	44	EBCDIC	CAXCNAM (in IGGCAXWA)	Name of catalog in which the data space is defined
96	60	SMF69VSR	6	EBCDIC	Name field of VSAM catalog volume entry	Volume serial number of the volume on which the data space is defined

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

Record Type 82 – Security

Record type 82 is a security record used to record information about the events and operations of the Programmed Cryptographic Facility, Program Number 5740-XY5. The record length is 41 bytes plus the length of the variable relocate sections.

Record type 82 is written to the SMF data set at the completion of each of the following cryptography functions:

- Initialization.

  The record is written when the Programmed Cryptographic Facility is initialized, either when cryptography is started or as a part of the key generator utility program.

- Start.

  The record is written by ICTVKM01 when a START command is issued for cryptography.

- Stop.

  The record is written by ICTVKM01 when a STOP command is issued for cryptography.

- Generation of an operational key.

  If specified in the initialization options for cryptography, ICTVKM04 writes a record after processing each GENKEY macro instruction.

- Transformation of an operational key.

  If specified in the initialization options for cryptography, ICTVKM04 writes the record after processing each RETKEY macro instruction.

- Execution of the key generator utility.

  The record is written by ICTVKG00 after the execution of the key generator utility program, thus providing a record of changes to the cryptographic key data set (CKDS).

Record type 82 consists of a header section and five possible relocate sections. The header section identifies the job and step name of the cryptography user, the cryptography function that the record describes, and the return code issued by the function. The header section is 41 bytes long.

The five possible variable relocate sections are:

- Key generator utility, which indicates changes made by the utility to the host system master key, the local keys, the cross keys, and the remote keys. The length of the section is 6 bytes.

- GENKEY function, which indicates the action taken in response to a GENKEY macro instruction. The length of the section is 27 bytes.

- RETKEY function, which indicates the action taken in response to a RETKEY macro instruction. The length of the section is 11 bytes.

- Cryptography initialization, which describes the SMF recording options in effect at initialization and the cryptography function and key manager user SVC numbers. The length of the section is 7 bytes.

- Installation data, which contains any information supplied by an installation user exit routine. The maximum length of the section is 66 bytes.

Note: The number of relocate sections depends on the type of action taken. For instance, the record written when the Programmed Cryptographic Facility stops consists only of the header section. When the Programmed Cryptographic Facility has previously been initialized within the same IPL, the record written when cryptography starts also consists of only the header section. The number of relocate sections is indicated in CRY82VCT(offset 37).

Using the ICTSMF82 and ICTCRY82 macros: Because record type 82 cannot be mapped using the IFASMFR macro, you must use two mapping macros to symbolically address record type 82. The macros are supplied on SYS1.MACLIB. ICTSMF82 maps the fields in record type 82 whose names start with SMF. For the Programmed Cryptographic Facility, ICTSMF82 maps the fields addressed by offsets X'0' through X'12'. ICTCRY82 maps the fields in record type 82 whose names start with CRY. For the Programmed Cryptographic Facility, ICTCRY82 maps the fields addressed by offsets X'14' through X'28' and the variable relocate sections.

The format is:

Offsets		Name	Length	Format	Source	Description
0	0	SMF82FLG	1	binary	SVC 83	System indicator Bit — Meaning When Set 0-5 — Reserved 6 — VS2 7 — VS1
1	1	SMF82RTY	1	binary	internal	Record type
2	2	SMF82TME	4	binary	SVC 11 or SVC 83	Time, in hundredths of a second, record was moved to the SMF buffer
6	6	SMF82DTE	4	packed	SVC 11 or SVC 83	Date record was moved to the SMF buffer, in the form 00YYDDDF where F is the sign
10	A	SMF82SID	4	EBCDIC	JMRCPUID or SMCASID	System identification (taken from SID parameter)
14	E	SMF82LNG	4	binary	internal	Length of record header
18	12	SMF82TID	2	binary	internal	Security product identifier: X'0001' for Programmed Cryptographic Facility (Program number 5740-XY5)
20	14	CRY82USR	8	EBCDIC	TIOCNJOB in TIOT	Jobname
28	1C	CRY82GRP	8	EBCDIC	TIOCSTEP in TIOT	Stepname
36	24	CRY82FLG	1	binary	internal	Reserved
37	25	CRY82VCT	2	binary	internal	Number of variable relocate sections
39	27	CRY82FTN	1	binary	internal	Function code Code — Meaning 1 — Key generator 2 — GENKEY function 3 — RETKEY function 4 — Start cryptography 5 — Stop cryptography
40	28	CRY82RTC	1	binary	internal	Return code issued by function or X'FF' if function terminated abnormally

(Continued)

Offsets	Name	Length	Format	Source	Description
Key Generator Utility Relocate Section:					
+0	CRY82DTP	1	binary	ICTVKG00	Data type indicator: X'01' for key generator utility
+1	CRY82DLN	1	binary	ICTVKG00	Length of the data that follows
+2	CRY82SMK	1	binary	ICTVKG00	Host system master key flags Bit Meaning When Set 0 Host system master key was successfully changed 1-7 Reserved
+3	CRY82LMK	1	binary	ICTVKG01	Local key flags Bit Meaning When Set 0 At least one local key was updated 1 At least one local key was added 2 At least one local key was deleted from the CKDS 3-7 Reserved
+4	CRY82CMK	1	binary	ICTVKG01	Cross key flags Bit Meaning When Set 0 At least one pair of cross keys was updated 1 At least one pair of cross keys was added 2 At least one pair of cross keys was deleted from the CKDS 3-7 Reserved
+5	CRY82RMK	1	binary	ICTVKG01	Remote key flags Bit Meaning When Set 0 At least one remote key was updated 1 At least one remote key was added 2 At least one remote key was deleted from the CKDS 3-7 Reserved
GENKEY Function Relocate Section:					
+0	CRY82DTP	1	binary	ICTVKM04	Data type indicator: X'02' for GENKEY function
+1	CRY82DLN	1	binary	ICTVKM04	Length of the data that follows
+2	CRY82GFG	1	binary	ICTVKM04	GENKEY activity flags Bit Meaning When Set 0 'LOCKEY' parameter was in error 1 'LOCKEY2' parameter was in error 2 'REMKEY' parameter was in error 3 'OPKEY' was generated by the key manager. When bit 3 is off, 'OPKEY' was supplied to the key manager. 4 Installation data relocate section was omitted from this record because the data supplied by installation exit exceeded the length of CRY82ID (64 bytes). 5-7 Reserved
+3	CRY82LK1	8	EBCDIC	GENKEY LOCKEY parameter	'LOCKEY' key name
+11	CRY82LK2	8	EBCDIC	GENKEY LOCKEY2 parameter	'LOCKEY2' key name
+19	CRY82REM	8	EBCDIC	GENKEY REMKEY parameter	'REMKEY' key name

(Continued)

Offsets	Name	Length	Format	Source	Description
RETKEY Function Relocate Section:					
+0	CRY82DTP	1	binary	ICTVKM04	Data type indicator: X'03' for RETKEY function
+1	CRY82DLN	1	binary	ICTVKM04	Length of the data that follows
+2	CRY82RFG	1	binary	ICTVKM04	RETKEY activity flags Bit Meaning When Set 0 Installation data relocate section was omitted from this record because the data supplied by the installation exit exceeded the length of CRY82ID (64 bytes) 1-7 Reserved
+3	CRY82RKN	8	EBCDIC	RETKEY REMKEY parameter	'REMKEY' key name
Cryptography Initialization Relocate Section:					
+0	CRY82DTP	1	binary	ICTVKG00 ICTVKM01	Data type indicator: X'04' for initialization
+1	CRY82DLN	1	binary	ICTVKG00 ICTVKM01	Length of the data that follows
+2	CRY82SMF	1	binary	ICTOPTNS	SMF option flags Bit Meaning When Set 0 SMF records not written for GENKEY function 1 SMF records not written for RETKEY function 2-7 Reserved
+3	CRY82SIM	2	binary	ICTOPTNS	Cryptography function user SVC number in the form X'cccc'
+5	CRY82KMG	2	binary	ICTOPTNS	Key manager user SVC number in the form X'cccc'
Installation Data Relocate Section:					
+0	CRY82DTP	1	binary	ICTVKG00 ICTVKM04	Data type indicator: X'05' for installation data
+1	CRY82DLN	1	binary	ICTCKG90 ICTMGR90 ICTMGR95	Length of the data that follows
+2	CRY82ID	VAR	EBCDIC	ICTCKG90 ICTMGR90 ICTMGR95	Installation data written by an installation user exit routine. The maximum length is 64 bytes.

Appendix A: Field-to-Record Cross-Reference

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

Some records have sections of fields that are generated only when specific events occur, such as when IPL is completed or when a device is varied online or offline. The following abbreviations will appear under the "offset" column to indicate the section of the record where the field is found:

ACT	Accounting section
CRYP	Cryptography initialization relocate section (5740-XY5)
DCB	DCB/DEB DASD entension entry
DCT	Line DCT and modified line entry
DEV	Device entry
DSC	Data set characteristics section
EXT	Extent entry
GENK	GENKEY relocate section (5740-XY5)
IDAT	Installation data relocate section (5740-XY5)
ISAM	ISAM extension section
KG	Key generator relocate section (5740-XY5)
MSS	MSS IPL configuration section
PAR	Partition entry
REL	Relocate section
RETK	RETKEY relocate section (5740-XY5)
STAT	Statistics section
TAPD	DCB/DEB tape extension entry
TAPU	UCB tape extension entry
UCB	UCB section
VAR	VARY ONLINE,S and VARY OFFLINE,S section
VOL	Volume entry
VTAM	Modified VTAM user entry

Field Description	Record Type	Offset Dec.	Offset Hex.
Access-method interfaces option codes	14	246	F6
	15	246	F6
Accounting fields, job	5	117	75
	20	61	3D
Accounting fields, step	4	ACT	
Address, VTOC	19	36	24
Allocation, device start time	4	86	56
Allocation status indicator	14	63	3F
	15	63	3F
Alternate tracks, number of unused	19	46	2E
Attentions received, number of	50	34	22
Beginning cylinder and track	64	EXT	
Block count volume	14	TAPD	
	15	TAPD	
Block size	21	40	28
	64	DSC	
BTSS indicator	6	98	62
BUF parameter	0	18	12
Buffer, date record was moved to	ALL[1]	6	6
Buffer, number of bytes in	0	18	12
Buffer, time record was moved in	ALL[1]	2	2
Buffers used in read channel programs, number of	50	50	32
Card-image records in DD DATA and DD* data sets read for step/job	4	47	2F
	5	47	2F
Catalog name	62	42	2A
	63	44	2C
	64	40	28
	67	40	28
	68	38	26
	69	52	34
Catalog record size	63	40	28
	63	42	2A
	67	128	80
Catalog records	63	132	84
	67	130	82
CCHH, beginning	64	EXT	
CCHH, ending	64	EXT	

[1] Except 2, 3, 4, and 5

Field Description	Record Type	Offset Dec.	Offset Hex.
Channel address	4	DEV	
	8	DEV	
	9	DEV	
	10	DEV	
	11	DEV	
	14	UCB	
	15	UCB	
	19	60	3C
	21	22	16
	43	DCT	
	44	DCT	
	64	EXT	
	69	38	26
Class, job	5	71	47
	13	PAR	
Class, SYSOUT	5	63	3F
	6	38	26
Classes, number of job	13	PAR	
Cleaner actions, number of	21	37	25
Completion code, job	5	51	33
Completion code, step	4	51	33
Component indicator	64	39	27
Component or cluster name	62	92	5C
	64	84	54
Continuous wait time limit	0	14	E
Control areas, number that were split	64	STAT	
Control interval size	64	DSC	
Control intervals, number of unused	64	STAT	
Control intervals, number that were split	64	STAT	
Copies distribution	6	65	41
Copies flashed, number of	6	97	61
Copy modification module name	6	89	59
CPU time	4	ACT	
	5	113	71
Cylinder overflow areas that are full, number of	14	ISAM	
	15	ISAM	
Cylinders, number of unallocated	19	48	30
	69	44	2C
Cylinders in independent index, prime data, and independent overflow areas, number of	14	ISAM	
	15	ISAM	
Cylinders in largest continuous unallocated area, number of	69	48	30
Cylinders in largest unallocated extent, number of	19	52	34
DASD volume status indicator	14	UCB	
	15	UCB	

Field Description	Record Type	Offset Dec.	Offset Hex.
Data set indicator	14	38	26
	14	51	33
	14	248	F8
	15	38	26
	15	51	33
	15	248	F8
Data set name	17	40	28
	18	40	28
	18	84	54
Data set organization	14	240	F0
	15	240	F0
Data set sequence count	14	TAPU	
	15	TAPU	
Data set sequence number	14	TAPU	
	15	TAPU	
Data set serial number	14	TAPD	
	15	TAPD	
Data sets processed by writer, number of	6	52	34
Data space extents, number of free	69	42	2A
Date initiator selected step/job	4	43	2B
	5	43	2B
Date job terminated	5	6	6
Date reader recognized end of job	5	58	3A
Date reader recognized the JOB card	4	26	1A
	5	26	1A
	6	26	1A
	10	26	1A
	14	26	1A
	15	26	1A
	17	26	1A
	18	26	1A
	20	26	1A
	62	26	1A
	63	26	1A
	64	26	1A
	67	26	1A
	68	26	1A
	69	26	1A
Date record was moved to dump data set	2	6	6
	3	6	6
Date record was moved to SMF buffer	ALL[1]	6	6
Date recording was started when SMF data set became available	7	20	14
Date step terminated	4	6	6
Date SYSOUT was started	6	43	2B
DD DATA and DD* data set records read for step/job	4	47	2F
	5	47	2F
DD entry length	14	48	30
	15	48	30
DD name	14	52	34
	15	52	34
	64	DSC	

[1]Except 2, 3, 4, and 5

Field Description	Record Type	Offset Dec.	Offset Hex.
Deleted records, number of	64	STAT	
Density, tape	21	39	27
Device allocation requests, number of	14	50	32
	15	50	32
Device class	4	DEV	
	5	69	45
	8	DEV	
	9	DEV	
	10	DEV	
	11	DEV	
Device indicator	14	51	33
	14	248	F8
	15	51	33
	15	248	F8
Device recording controls, MSF	22	MSS	
Device SSID	22	VAR	
Devices, MSF data recording	22	MSS	
Devices requested during allocation, number of	14	50	32
	15	50	32
Device allocation start time	4	86	56
Drive number	19	62	3E
DSCBs, number of	19	42	2A
DSCB0s, number of	19	44	2C
Dump data set, date record was moved to	2	6	6
	3	6	6
Dump data set, time record was moved to	2	2	2
	3	2	2
End-of-volume indicator	14	249	F9
	15	249	F9
Ending cylinder and track	64	EXT	
Ending status of read channel program, number of times ATTN included	50	38	26
Entries in LINE table, number of	43	38	26
Entry name	63	88	58
	67	84	54
	68	82	52
	68	126	7E
Entry type indicator	63	38	26
	63	39	27
	67	38	26
	67	39	27
Erase gaps, number of	21	35	23
EXCP count	4	DEV	
	14	UCB	
	15	UCB	
	64	STAT	

Field Description	Record Type	Offset Dec.	Offset Hex.
Expiration time of ten-minute intervals	1	18	12
	12	18	12
Extents, number of	14	UCB	
	15	UCB	
	64	STAT	
Extents, number of unallocated	19	56	38
Extents in independent index, prime data, and independent overflow areas, number of	14	ISAM	
	15	ISAM	
Extents released by DADSM routine, number of	14	DCB	
	15	DCB	
Flash overlay name	6	93	5C
Form number	6	53	35
GENKEY flags	82	GENK[2]	
GENKEY key name	82	GENK[2]	
Highest used relative byte address	64	130	82
I/O processing method indicator	14	249	F9
	15	249	F9
I/O status	6	51	33
Index levels, number of	14	ISAM	
	15	ISAM	
	64	STAT	
Initiator select date for step/job	4	43	2B
	5	43	2B
Initiator select time for step/job	4	39	27
	5	39	27
Input class for job	5	71	47
Input/output (*see* I/O)			
Inserted records, number of	64	STAT	
Intelligent controller name	50	14	E
JFCB	14	64	40
	15	64	40
JFCB TTR address	14	60	3C
	15	60	3C
Job accounting fields	5	117	75
	20	61	3D
Job class	5	71	47
	13	PAR	
Job classes, number of	13	PAR	
Job completion code	5	51	33
Job CPU time	5	113	71

[2] Programmed Cryptographic Facility (5740-XY5)

Field Description	Record Type	Offset Dec.	Offset Hex.
Job log identification	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
	62	14	E
	63	14	E
	64	14	E
	67	14	E
	68	14	E
	69	14	E
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
	62	14	E
	63	14	E
	64	14	E
	67	14	E
	68	14	E
	69	14	E
	82	20[2]	
Job priority	5	53	35
Job termination date	5	6	6
Job termination indicator	5	62	3E
Job termination time	5	2	2
Job wait time limit	0	14	E
JWT parameter	0	14	E
Key, storage protect	4	82	52
	5	72	48
Key length	64	DSC	
Length of DD entry	14	48	30
	15	48	30
Length of security record header	82	14	E[2]
Limit, continuous wait time	0	14	E
Line DCTs, number of	43	39	27
Line number	43	DCT	
	44	DCT	
LINE table entries, number of	43	38	26
Lines modified, number of	44	29	1D
Lines started when STOP was received, number of	45	29	1D

[2] Programmed Cryptographic Facility (5740-XY5)

Field Description	Record Type	Offset Dec.	Hex.
Lines to start, number of	43	41	29
Logical port DCTs, number of	43	39	27
	43	40	28
Logical ports active when STOP was received, number of	45	30	1E
Logical record size, maximum	64	DSC	
Logical records, number written	6	47	2F
Logical records in overflow and prime data areas, number of	14	ISAM	
	15	ISAM	
logon identifier of user	5	76	4C
	6	57	39
LOGON record	47	72	48
	49	72	48
Macro instruction and option types	14	243	F3
	14	ISAM	
	15	243	F3
	15	ISAM	
Maximum logical record size	64	DSC	
Maximum number of readers supported	43	36	24
Maximum number of writers supported	43	37	25
Modified lines, number of	44	29	1D
Modified remote characteristics	44	VTAM	
MODIFY command type	44	28	1C
Module identification	19	62	3E
MSF data recording devices	22	MSS	
MSF device recording controls	22	MSS	
MSF indicator	22	MSS	
Name of catalog	62	42	2A
	63	44	2C
	64	40	28
	67	40	28
	68	38	26
	69	52	34
Name of component or cluster	62	92	5C
	64	84	54
Name of copy modification module	6	89	59
Name of data set	17	40	28
	18	40	28
	18	84	54
Name of entry	63	88	58
	67	84	54
	68	82	52
	68	126	7E
Name of flash overlay	6	93	5C

Field Description	Record Type	Offset Dec.	Offset Hex.
Name of job	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
	62	14	E
	63	14	E
	64	14	E
	67	14	E
	68	14	E
	69	14	E
	82	20	14[2]
Name of locally attached intelligent controller	50	14	E
Name of program	4	54	36
Name of programmer	5	93	5D
	20	40	28
Name of RTAM start procedure	43	20	14
	44	20	14
	45	20	14
Name of step	4	62	3E
	82	28	1C[2]
Names of translate table	6	73	49
New catalog record	63	132	84
	67	130	82
New catalog record size	63	40	28
	67	128	80
New data set name	18	84	54
New entry name	68	126	7E
Noise blocks, number of	21	34	22
Number of attentions received	50	34	22
Number of blocks per volume	14	TAPD	
	15	TAPD	
Number of buffers used in read channel programs	50	50	32
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 card-image records in DD DATA and DD* data sets read for step/job	4	47	2F
	5	47	2F
Number of cleaner actions	21	37	25
Number of copies flashed	6	97	61
Number of cylinders in independent index, prime data, and independent overflow areas	14	ISAM	
	15	ISAM	
Number of cylinders in largest continuous unallocated area	69	48	30

[2] Programmed Cryptographic Facility (5740-XY5)

Field Description	Record Type	Offset Dec.	Offset Hex.
Number of cylinders in largest unallocated extent	19	52	34
Number of data set sequences	14	TAPU	
	15	TAPU	
Number of data sets processed by writer	6	52	34
Number of devices requested during allocation	14	50	32
	15	50	32
Number of DSCBs	19	42	2A
Number of DSCB0s	19	44	2C
Number of erase gaps	21	35	23
Number of EXCPs	4	DEV	
	14	UCB	
	15	UCB	
	64	STAT	
Number of extents	14	UCB	
	15	UCB	
	64	STAT	
Number of extents in independent index, prime data, and independent overflow areas	14	ISAM	
	15	ISAM	
Number of extents released by DADSM routine	14	DCB	
	15	DCB	
Number of free data space extents	69	42	2A
Number of full cylinder overflow areas	14	ISAM	
	15	ISAM	
Number of index levels	14	ISAM	
	15	ISAM	
	64	STAT	
Number of job classes	13	PAR	
Number of line DCTs	43	39	27
Number of LINE table entries	43	38	26
Number of lines modified	44	29	1D
Number of lines started when STOP was received	45	29	1D
Number of lines to start	43	41	29
Number of logical port DCTs	43	39	27
	43	40	28
Number of logical ports active when STOP was received	45	30	1E
Number of logical records in overflow and prime data areas	14	ISAM	
	15	ISAM	
Number of logical records written	6	47	2F
Number of noise blocks	21	34	22
Number of partition	13	PAR	
Number of partition page-ins	4	REL	
Number of partition page-outs	4	REL	
Number of permanent read errors	21	32	20

Field Description	Record Type	Offset Dec.	Offset Hex.
Number of permanent write errors	21	33	21
Number of PIUs inbound	50	42	2A
Number of PIUs outbound	50	46	2E
Number of read channel programs	50	30	1E
Number of read or write accesses to an overflow record	14	ISAM	
	15	ISAM	
Number of readers supported, maximum	43	36	24
Number of records deleted, inserted, updated, and retrieved from component	64	STAT	
Number of records in component	64	STAT	
Number of records lost	7	14	E
Number of slowdowns entered by locally attached intelligent controller	50	54	36
Number of split control areas	64	STAT	
Number of split control intervals	64	STAT	
Number of START I/Os	21	30	1E
Number of step	4	38	26
Number of steps in job	5	38	26
Number of system page-ins	1	22	16
	12	22	16
Number of system page-outs	1	26	1A
	12	26	1A
Number of system page reclaims	1	30	1E
	12	30	1E
Number of TCBs needed for dump-load-restart operation	50	22	16
Number of temporary read errors	21	28	1C
Number of temporary write errors	21	29	1D
Number of times ending status of read channel program includes ATTN	50	38	26
Number of tracks allocated on device	14	UCB	
	15	UCB	
Number of tracks in largest unallocated extent	19	54	36
Number of tracks in largest continuous unallocated area	69	50	32
Number of tracks in overflow area	14	ISAM	
	15	ISAM	
Number of tracks released by DADSM routine	14	DCB	
	15	DCB	
Number of tracks requested but not allocated	64	128	80
Number of unallocated extents	19	56	38
Number of unallocated cylinders	19	48	30
	69	44	2C

Field Description	Record Type	Offset Dec.	Offset Hex.
Number of unallocated tracks	19	50	32
	69	46	2E
Number of unused alternate tracks	19	46	2E
Number of unused control intervals	64	STAT	
Number of volumes	17	87	57
	18	131	83
	62	136	88
Number of write channel programs	50	26	1A
Number of writers supported, maximum	43	37	25
Old catalog record	63	132	84
Old catalog record size	63	42	2A
Old data set name	18	40	28
Old entry name	68	82	52
OPEN routine indicator	14	245	F5
	15	245	F5
OPTCD=J output indicator	6	98	62
Open status indicator	62	38	26
Options, SMF	0	26	1A
Owner identification of direct access volume	19	22	16
Page-ins, partition	4	REL	
Page-ins, system	1	22	16
	12	22	16
Page-outs, partition	4	REL	
Page-outs, system	1	26	1A
	12	26	1A
Page reclaims, system	1	30	1E
	12	30	1E
Partition page-ins	4	REL	
Partition page-outs	4	REL	
Partition number	13	PAR	
Partition size requested	4	70	46
Passback area	47	68	44
	49	68	44
Permanent read errors, number of	21	32	20
Permanent write errors, number of	21	33	21
PIUs inbound, number of	50	42	2A
PIUs outbound, number of	50	46	2E
Priority, job	5	53	35
Priority, step	4	53	35

Field Description	Record Type	Offset Dec.	Offset Hex.
Program name	4	54	36
Program start time	4	90	5A
Programmed Cryptographic Facility	82[2]		
CKDS activity flags	82	KG[2]	
function code	82	39	27[2]
installation exit data	82	IDAT[2]	
number of variable relocate sections	82	37	25[2]
options	82	CRYP[2]	
return code	82	40	28[2]
Programmer's name	5	93	5D
	20	40	28
QID entry	47	20	14
	48	20	14
	49	20	14
RBA, highest used	64	130	82
Read channel programs, number of	50	30	1E
Read errors, number of	21	28	1C
	21	32	20
Read or write accesses to an overflow record, number of	14	ISAM	
	15	ISAM	
Reader device class	5	69	45
Reader end date	5	58	3A
Reader end time	5	54	36
Reader start date	4	26	1A
	5	26	1A
	6	26	1A
	10	26	1A
	14	26	1A
	15	26	1A
	17	26	1A
	18	26	1A
	20	26	1A
	62	26	1A
	63	26	1A
	64	26	1A
	67	26	1A
	68	26	1A
	69	26	1A
Reader start time	4	22	16
	5	22	16
	6	22	16
	10	22	16
	14	22	16
	15	22	16
	17	22	16
	18	22	16
	20	22	16
	62	22	16
	63	22	16
	64	22	16
	67	22	16
	68	22	16
	69	22	16
Reader unit type	5	70	46

[2] Programmed Cryptographic Facilty (5740-XY5)

Field Description	Record Type	Offset Dec.	Offset Hex.
Readers supported, maximum number of	43	36	24
Real storage, number of bytes in	0	27	1B
Real storage requested	4	74	4A
Record format	14	242	F2
	15	242	F2
Record indicator	4	98	62
	14	38	26
	15	38	26
	22	14	E
	63	38	26
	67	38	26
Record size, maximum logical	64	DSC	
Records, logical, number written	6	47	2F
Records deleted, inserted, updated, and retrieved from component, number of	64	STAT	
Records in component, number of	64	STAT	
Records in overflow and prime data areas, number of logical	14	ISAM	
	15	ISAM	
Record type	ALL	1	1
Records lost, number of	7	14	E
Relative address of JFCB or SIOT	14	60	3C
	15	60	3C
Relative address of last record processed for a physical sequential or partitioned data set	14	DCB	
	15	DCB	
Relative byte address, highest used	64	130	82
Remote characteristics modified	44	VTAM	
RETKEY flags	82	RETK[2]	
RETKEY key name	82	RETK[2]	
RTAM start procedure name	43	20	14
	44	20	14
	45	20	14
Security Product ID	82	18	12[2]
SID parameter	ALL	10	A
Situation indicator	64	38	26
SIOT TTR address	14	60	3C
	15	60	3C
Size, partition requested	4	70	46
Size of catalog record	63	40	28
	63	42	2A
	67	128	80
Size of control interval	64	DSC	

[2] Programmed Cryptographic Facility (5740-XY5)

Field Description	Record Type	Offset Dec.	Hex.
Slowdown entered by locally attached intelligent controller, number of times	50	54	36
SMF options	0	26	1A
Spindle identification	64	EXT	
	69	40	28
SSID of device	22	VAR	
Staging adapters	22	MSS	
Staging spindles in staging data groups	22	MSS	
Start date of recording when SMF data set became available	7	20	14
START I/Os, number of	21	30	1E
Start procedure name, RTAM	43	20	14
	44	20	14
	45	20	14
Start time of recording when SMF data set became available	7	16	10
Step accounting fields	4	ACT	
Step completion code	4	51	33
Step CPU time	4	ACT	
Step name	4	62	3E
	82	28	1C[2]
Step number	4	38	26
Step priority	4	53	35
Step termination date	4	6	6
Step termination indicator	4	83	53
Step termination time	4	2	2
STOP command processing indicator	45	28	1C
Storage, real, number of bytes in	0	27	1B
Storage, real requested	4	74	4A
Storage, used	4	74	4A
Storage, virtual, number of bytes in	0	22	16
Storage protect key	4	82	52
	5	72	48
Subsystem identification	43	14	E
	44	14	E
	45	14	E
	47	14	E
	48	14	E
	49	14	E
Subsystem identification of device	22	VAR	

[2] Programmed Cryptographic Facility (5740-XY5)

Field Description	Record Type	Offset Dec.	Offset Hex.
SYSOUT class indicator	5	63	3F
	6	38	26
SYSOUT start date	6	43	2B
SYSOUT start time	6	39	27
System identification	ALL	10	A
System indicator	ALL	0	0
System page-ins	1	22	16
	12	22	16
System page-outs	1	26	1A
	12	26	1A
System page reclaims	1	30	1E
	12	30	1E
System wait time	1	14	E
	12	14	E
Tape density	21	39	27
TCBs, number of	50	22	16
Temporary read errors, number of	21	28	1C
Temporary write errors, number of	21	29	1D
Termination indicator, job	5	62	3E
Termination indicator, step	4	83	53
Time, continuous wait	0	14	E
Time, expiration of ten-minute intervals	1	18	12
	12	18	12
Time, job CPU	5	113	71
Time, step CPU	4	ACT	
Time, system wait	1	14	E
	12	14	E
Time device allocation started	4	86	56
Time initiator selected step/job	4	39	27
	5	39	27
Time job terminated	5	2	2
Time program was started	4	90	5A
Time reader recognized end of job	5	54	36
Time reader recognized the 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
	20	22	16
	62	22	16
	63	22	16

Field Description	Record Type	Offset Dec.	Hex.
Time reader recognized the JOB card (cont'd)	64	22	16
	67	22	16
	68	22	16
	69	22	16
Time record was moved to dump data set	2	2	2
	3	2	2
Time record was moved to SMF buffer	ALL[1]	2	2
Time recording was started when SMF data set became available	7	16	10
Time step terminated	4	2	2
Time SYSOUT was started	6	39	27
TIOT status indicator	14	49	31
	15	49	31
Tracks, number in largest unallocated extent	19	54	36
Tracks, number of unallocated	19	50	32
	69	46	2E
Tracks, number of unused alternate	19	46	2E
Tracks allocated on device, number of	14	UCB	
	15	UCB	
Tracks in largest continuous unallocated area, number of	69	50	32
Tracks in overflow area, number of	14	ISAM	
	15	ISAM	
Tracks released by DADSM routine, number of	14	DCB	
	15	DCB	
Tracks requested but not allocated, number of	64	128	80
Translate table names	6	73	49
TTR of JFCB or SIOT	14	60	3C
	15	60	3C
TTRN of last record processed for a physical sequential or partitioned data set	14	DCB	
	15	DCB	
Type of I/O macro instruction and options	14	243	F3
	15	243	F3
Type of MODIFY command	44	28	1C
Unit address	4	DEV	
	8	DEV	
	9	DEV	
	10	DEV	
	11	DEV	
	14	UCB	
	15	UCB	
	19	61	3D
	21	23	17
	43	DCT	
	44	DCT	
	64	EXT	
	69	39	27

[1] Except 2, 3, 4, and 5

Field Description	Record Type	Offset Dec.	Offset Hex.
Unit type	5	70	46
	8	DEV	
	9	DEV	
	10	DEV	
	11	DEV	
	14	UCB	
	15	UCB	
	19	32	20
	21	24	18
	62	VOL	
	64	EXT	
Updated records, number of	64	STAT	
User identification	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
	44	VTAM	
	62	30	1E
	63	30	1E
	64	30	1E
	67	30	1E
	68	30	1E
	69	30	1E
User interface identification	51	14	E
User's logon identifier	5	76	4C
	6	57	39
Virtual storage, number of bytes in	0	22	16
Volume owner identification	19	22	16
Volume sequence number	14	250	FA
	15	250	FA
Volume serial number	14	UCB	
	15	UCB	
	17	VOL	
	18	VOL	
	19	16	10
	21	16	10
	62	86	56
	62	VOL	
	64	EXT	
	69	96	60
Volumes, number of	17	87	57
VTOC address	19	36	24
VTOC indicator	19	41	29
Write channel programs, number of	50	26	1A
Write errors, number of	21	29	1D
	21	33	21
Writers supported, maximum number of	43	37	25
3800 accounting information	6	65	41

Appendix B: SMF Compatibility Between VS1 and MFT

VS1 SMF and MFT SMF are compatible with the following exceptions:

- SYSGEN procedures – The MFT procedures for adding SMF and user-written exit routines to the operating system differ from the VS1 SYSGEN procedures.
- SMF data sets – You must catalog the VS1 SYS1.MANX and SYS1.MANY data sets on direct access devices (DASD).
- SMF exits – Three new SMF exits, IEFUIV, IEFU83, and IEFUJP, have been added for VS1.
- SMF SYS1.PARMLIB member – The MFT SMFDEFLT member of SYS1.PARMLIB has been replaced with the VS1 SMFxxxxx member. An IBM-default member, SMFPRM00, has also been added.
- SMF parameters – The MFT SMFDEFLT parameters ALT, MDL, and PRM are not supported by VS1. The VS1 SID parameter specifies both the system and the model on which SMF is active; its length is four bytes.
- SMF buffer size – The MFT BUF parameter can have a value from 400 to 65,536 bytes; the VS1 BUF parameter can have a value from 400 to 8,192 bytes.
- Common exit parameter area – VS1 data statistics (88 bytes) have been added to the common exit parameter area, especially for use by the IEFUJP exit.
- Sample SMF routines – MFT sample routines of SYS1.SAMPLIB have been added to the VS1 SYS1.ASAMPLIB data set. The MFT sample source report program, SMFPOST, has been changed to the VS1 sample program called SMFFRMT. The MFT sample sort exit routines SMFE35A and SMFE35B have been changed to the VS1 sample routine called SMFE35.
- Writing to installation-defined data sets – In MFT, exits IEFUJI, IEFUSI, and IEFACTRT cannot write to installation-defined data sets; in VS1, they can.
- SMF records – Several record types have been added, deleted and modified for VS1. Figure 35 lists these record types and the corresponding changes.
- SMF record fields – The VS1 values for fields such as CPU time, storage allocated and used, and EXCP count, may differ from the MFT values for these fields.

Record Types	Additions, Deletions, and Modifications from MFT for VS1
Header	System indicator added.
0	Size of virtual storage replaces size of main storage. Size of real storage added.
1	Expiration time of collection period and system page-ins, page-outs, and page reclaims for the interval added.
4	Requested partition size replaces allocation for hierarchy 0. Storage used replaces hierarchy 0 storage used. Device entries for spooled data sets deleted. Step termination indicator and record indicator replace reserved fields.
5	ABEND bit in job termination indicator is set when any step abnormally terminates. Termination indicator bit for IEFUJV exit added. Job completion code contains ABEND code for the last step that abnormally terminated, regardless of normal processing by successive steps. Reserved field replaces checkpoint/restart indicator. User's logon identifier added.
6	User's logon identifier added. Accounting information for IBM 3800 added.
9	Record written only for *devices* varied online.
11	Record written only for *devices* varied offline.
12	Expiration time of collection period and system page-ins, page-outs, and page reclaims for the interval added.
13	Record written only for storage assigned to *problem program* partitions.
22	Record added for configuration of mass storage devices.
30-35, 38 and 40-42	Records deleted.
43-45 and 47-49	Records added for RTAM
50	Record added for ACF/VTAM.
51	Record added for subsystems.
62-64 and 67-69	Records added for VSAM.
82	Record added for Programmed Cryptographic Facility (5740-XY5).

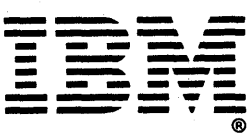
Figure 35. Record Modifications from MFT for VS1

Index

Indexes to OS/VS1 publications are consolidated in *OS/VS1 Master Index*. This master index references other publications that contain additional information about the subjects listed here.

GC24-5115-2

OS/VS1 System Management Facilities (SMF) (File No. S370-34) Printed in U.S.A. GC24-5115-2

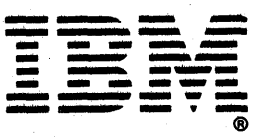


International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N. Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N. Y., U. S. A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N. Y., U. S. A. 10601

OS/VS1 System Management Facilities (SMF) (File No. S370-34) Printed in U.S.A. GC24-5115-2



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N. Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N. Y., U. S. A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N. Y., U. S. A. 10601