File Number S360-31
Form C28-6712-1

OS

IBM Systems Reference Library

IBM System/360 Operating System

Planning for System Management Facilities

System Management Facilities (SMF) is an optional feature of the System/360 Operating System that can be selected at system generation in conjunction with the Multiprogramming with a Fixed number of Tasks (MFT) configuration or the Multiprogramming with a Variable number of Tasks (MVT) configuration. SMF collects and optionally records system, job management, and data management information and provides control program exits to installation-supplied routines that can periodically monitor the operation of a job or job step.

This publication:

- Introduces basic SMF concepts.
- Describes SMF records, control program exits, and data management requirements.
- Tells how to plan, write and test the installation-supplied exit routines.
- Gives procedures for incorporating SMF into the operating system.
- Describes IPL and data management procedures that must be performed by the operator.
- Describes SMF record formats.
- Explains SMF messages.

System Management Facilities can be used with the Model 65 multiprocessing (M65MP) option of MVT and the Remote Job Entry (RJE) and Graphic Job Processor (GJP) options of MFT and MVT.

This publication is a planning aid only. It is intended for use prior to the availability of System Management Facilities and shall be replaced by reference documentation when that option becomes available.

Second Edition (December, 1969)

This edition, Form C28-6712-1, is a major revision of, and obsoletes Form C28-6712-0. Changes are designated in three ways:

1. A vertical line appears to the left of affected text where only part of the page has been changed.

2. A bullet (•) appears to the left of the page number where the complete page should be reviewed.

3. A bullet (•) appears to the left of the title of each figure that has been changed.

Changes are periodically made to the specifications herein; before using this publication in connection with the operation of IBM systems, refer to the latest IBM System/360 SRL Newsletter, Form N20-0360, for the editions that are applicable and current.

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

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Systems Publications, Department D58, PO Box 390, Poughkeepsie, N. Y. 12602

Preface

This publication provides installation managers and system programmers with the information required to plan for implementation of the System Management Facilities option. The reader should be familiar with the information presented in the following publications:

IBM System/360 Operating System:

Concepts and Facilities, Form C28-6535

System Programmer's Guide, Form C28-6550

The following publications are referred to throughout the text:

IBM System/360 Operating System:

Assembler (F) Programmer's Guide, Form C28-3756

PL/I (F) Programmer's Guide, Form C28-6594

PL/I Reference Manual, Form C28-8201

Sort/Merge, Form C28-6543

Supervisor and Data Management Services, Form C28-6646

System Generation, Form C28-6554

Utilities, Form C28-6586

This publication consists of seven sections:

1. An introduction, which describes the basic SMF functions and their relationship to the operating system.
2. A section describing SMF exits, records, and data management requirements. This section contains information needed to select a specific SMF implementation for an installation.
3. Information on planning, writing, and testing routines that may be supplied by the installation.
4. Procedures for incorporating SMF into an operating system.
5. Operational considerations, including IPL and data management procedures, and execution of installation report programs.
6. SMF record formats.
7. An explanation of SMF messages.

Contents

Illustrations

Figures

Tables

Section 1: Introduction

System Management Facilities (SMF) is an optional feature of the System/360 Operating System that can be selected at system generation in conjunction with the Multiprogramming with a Fixed number of Tasks (MFT) configuration or the Multiprogramming with a Variable number of Tasks (MVT) configuration.

SMF provides a base for installation-supplied routines which perform the following functions:

- System, job, and step accounting.
- Job process monitoring.
- Data set accounting.
- Direct access volume usage.

All these functions depend partly on IBM-supplied data collection routines and control program exits and partly on installation-supplied analysis and report routines.

- SMF collects system, job, and step related information that can be used by installation-written management information programs reporting system efficiency, performance, and usage.

- SMF provides control program exits that can be used by installation-written routines to monitor jobs at specific points as they are processed. These routines can enforce installation standards such as: identification, priority, resource allocation, maximum execution time, and the number of logical records written by a user program to the SYSOUT data set.

- SMF records the activity of all user data sets and the status of all non-temporary user data sets. This information can be used by installation-written routines that report the data sets used by each job or step.

- SMF records the status of all removable direct access volumes. This information can be used by installation-written routines that correct problems of volume deterioration (defective tracks) and space fragmentation.

Since the need for these functions varies widely from one installation to another, SMF provides a great deal of flexibility. SMF must be specified at system generation but its use can be modified at each initial program loading (IPL).

SMF Concepts

SMF provides two basic services:

- Data Collection.
- Control Program Exits.

Data Collection

SMF writes 20 types of records to a data set residing on either a tape unit or a direct access device. Information includes:

- System configuration.

- Job and step identification.

- CPU wait time.

- CPU and I/O device usage by job and step.

- Amount of storage assigned to each active reader, writer and problem program partition (MFT).

- Temporary and non-temporary data set usage by job and step.

- Temporary and non-temporary data set status.

- Status of removable direct access volumes.

An installation selects which records it will use by including SMF default parameters in a member (SMFDEFLT) of SYS1.PARMLIB prior to system generation. One of the parameters (OPI) allows the operator to override the other preset default parameters at IPL time. Additional records can be written to the SMF data set by installation-written routines using the SMFWTM macro instruction.

The SMF data set must be on a permanently resident direct access device or tape unit. If the SMF data set is on a direct access device, it must be periodically dumped (by the SMF dump program) to another data set (e.g., on tape) which will serve as input to the installation-supplied analysis and report routines. Other factors must be considered, however, before deciding whether to put the SMF data set on a tape unit or on a direct access device. These considerations are discussed in Section 2.

Each record in the SMF data set includes a header containing the date and time of recording, CPU model number, and an installation-defined system identification code. A record type code indicates the information contained in the rest of the record. The information contained in each record type is described in Section 2.

System information is recorded at IPL, whenever a VARY ONLINE or VARY OFFLINE is entered, or when job management brings a device online through allocation recovery. This information includes main storage size, I/O devices online, and (for MFT) the amount of storage assigned to each active reader, writer and problem program partition.

Job and step information is recorded at job initiation and at job and step termination. This includes identification fields and accounting information from the JOB and EXEC statements, start time, CPU time (the time a job or step actually uses the CPU), SYSIN and SYSOUT usage, and job or step termination status.

Data set information is recorded each time a data set opened by a user program is scratched, renamed, closed, or processed by EOV. Information includes data set names, number of volumes, volume serial numbers, and various control block fields.

Direct access volume information is recorded at IPL and HALT EOD and whenever the system requests that a volume be demounted. Information includes volume serial number, VTOC address, owner identification number, number of unallocated tracks, amount of unallocated space, and size of the largest extent.

Analysis can consist of simply listing the SMF data set, using the operating system's sort/merge program to order the data, or performing detailed analysis operations using installation-written analysis and report routines.

Control Program Exits

SMF provides exits in the control program that can be used by installation-supplied routines:

- From the reader/interpreter before each JCL statement is interpreted.
- From the reader/interpreter before a job is placed on the job queue.
- From the initiator/terminator when a job is selected for initiation.
- From the initiator/terminator when a step is selected for initiation.
- From the initiator/terminator when a step is terminated.
- From the initiator/terminator when a job is terminated.
- From the supervisor when a specified SYSOUT limit for a system output SYSOUT data set is exceeded.
- From the timer second level interruption handler (SLIH) when a specified CPU time limit for a job is exceeded.
- From the timer second level interruption handler (SLIH) when a specified CPU time limit for a step is exceeded.
- From the timer second level interruption handler (SLIH) when a specified job wait time limit is exceeded.

An installation can make use of any or all of these exits by providing installation-written exit routines and including them in system library SYS1.CI505 before system generation or in the link library SYS1.LINKLIB after system generation. Dummy routines are automatically provided for all unused exits.

The installation-written exit routines can cancel jobs, use the SMFWTM macro instruction to write installation-defined records to the SMF data set, or open and close other installation-defined data sets. Thorough debugging is especially important for these routines since they will become part of the control program where errors could cause repeated system failure.

System Requirements

System Management Facilities can be specified as an option at system generation only in conjunction with the Multiprogramming with a Fixed number of Tasks (MFT) configuration or the Multiprogramming with a Variable number of Tasks (MVT) configuration. SMF can be used with the Model 65 multiprocessing (M65MP) option of MVT and the Remote Job Entry (RJE) and Graphic Job Processor (GJP) options of MFT and MVT.

At IPL, an SMF initialization routine receives control and reads in the member (SMFDEFLT) of SYS1.PARMLIB containing SMF control parameters. The options specified in these parameters -- including such options as whether both the data collection routines and the control program exits will be active, the job wait time limit, and the definition of the SMF data sets -- will be typed out at the console if the operator is allowed to modify them for the current workday. (The default parameters can be permanently changed by replacing the member in SYS1.PARMLIB.) Before job processing begins, the SMF initialization routine records IPL and initial I/O configuration information and the amount of storage assigned to each active reader, writer and problem program partition (MFT only).

As the input stream is read in for processing, the reader/interpreter routine passes control to an exit routine just before each job control statement is interpreted. This routine may verify any fields in the statement, modify JCL, or reject jobs that do not meet installation standards. As each job is enqueued, the same exit routine receives control for further validity checking. Other exit routines receive control just before job and step initiation. These routines can decide whether to continue processing the job or step.

On job initiation, SMF creates a job commencement record identifying the job and the programmer, giving the reader start time and reader start date, and including accounting information. On job or step termination, SMF creates a record identifying the job or step, its execution time, and its use of system resources such as main storage and I/O devices.

Just before the record is written to the SMF data set, an installation-written accounting routine receives control. The accounting routine may modify the SMF records, add a record to its own accounting data set, or use the SMF I/O subroutines to add a record or records to the SMF data set. On return, the accounting routine indicates whether the job is to continue (if a step termination is being processed) and whether the SMF job or step termination record is to be written.

Data set information is recorded whenever a data set opened by a user program is scratched, renamed, closed, or processed by end-of-volume (EOV). Direct access volume information is recorded for direct access devices online at IPL, whenever a volume is demounted, and for direct access devices online at HALT EOD.

If the SMF data set is defined on a direct access device, records continue to be written to the primary SMF data set (SYS1.MANX) until the end of the allocated extent is reached. Then SMF opens the alternate SMF data set (SYS1.MANY) and continues recording. The operator should then enter a job into the system to copy SYS1.MANX to a dump data set using the SMF dump program.

Similar recording and copying operations continue throughout the workday, with SMF adding special records whenever a VARY command or allocation recovery changes the system configuration and, at the end of other intervals, to record system wait time. When the workday ends, the HALT EOD command from the operator causes the system wait time to be recorded and the SMF buffers to be emptied into the active SMF data set. The SMF dump program may then be used to copy the active SMF data set to the dump data set, which now contains a complete history of the day's processing. The dump data set serves as input to the installation-supplied analysis and report programs, which may be executed as ordinary problem programs under the operating system.

The analysis and report functions using SMF data are broad and flexible, limited only by the installation's needs and the skill and imagination of its programmers and systems analysts. A user exit routine from the sort/merge program may be used to list and total system usage by account number. A separate analysis program can process the SMF records in the order recorded, to detect excessive system wait time or inefficient I/O device utilization. This information could suggest changes in the job mixture or device allocation leading to improved system throughput.

Performance

Degradation of system throughput due to SMF will depend on such factors as the extent and blocksize of the SMF data sets, the types of devices to which the data sets are assigned, the types of SMF records written, and the execution time of exit routines.

Section 2: Choosing an Implementation

Planning for the SMF option consists of choosing which of the various features offered are suited to the purposes of the installation, then defining the SMF control parameters and writing the installation routines required to implement those features. The installation manager must decide what controls he wants to place on jobs submitted to the installation, what information he wants about the jobs, and in what format he wants the information. System programmers must then "tailor" the SMF option to provide the desired output most efficiently. This section discusses the decisions that must be made, and gives some of the factors that must be considered in those decisions. The following section includes the details necessary for implementing the desired features.

SMF Records

The SMF data collection routines format and write 20 types of records to the SMF data set. Some records describe system configuration and changes to system configuration after IPL. Other records describe the processing of jobs by the system, the status and activity of user data sets, and the status of removable direct access volumes.

The installation can suppress the writing of SMF records at IPL time and can select certain groups of SMF-formatted records through the use of SMF control parameters. Installation-formatted records can be created in the installation-written exit routines to supplement or replace the SMF-formatted records.

SMF Record Types

The SMF record types are described in the following paragraphs. (See Section 6 for the actual record formats.) The methods for suppressing the records are discussed following the record description.

Standard Record Header: Every SMF record begins with a standard record heading containing:

- A record type code.
- The date and time (timestamp) of writing.
- The system identification.

IPL Record (Type 0): The IPL record is created during system initialization (after IPL). It includes the amount of main storage available, and the SMF options in effect.

Wait Time Record (Type 1): SMF maintains a total of the time the CPU spends in the wait state. The record is written at the first step termination following a ten minute interval. The time value written is not cumulative; that is, each wait time record indicates the wait time since the previous wait time record was written.

Dump Header Record (Type 2): This record consists only of the standard record heading. It indicates the beginning of a "dump" of the SMF data set from a direct access device to tape.

Dump Trailer Record (Type 3): Like the dump header record, the dump trailer record consists only of the standard heading. It marks the end of an SMF dump to tape. Record types 2 and 3 are written directly onto the dump data set by the SMF dump program.

Step Termination Record (Type 4): The step termination record is created after the normal or abnormal completion of a job step. The step is identified by job log number (job name and reader start time for the job), step name, the number of the step within the job, the user identification field (which may be initialized by the installation to facilitate subsequent sorting of records), and the program name. If accounting numbers were specified in the EXEC statement, they are included.

The record provides operating information such as the time the step was started and completed, the CPU time, the amount of main storage requested and used, and the termination status. I/O activity is recorded for each data set used by this step; each entry lists the device class, type, and address, and the EXCP count for the data set. The EXCP count includes direct EXCPs, program controlled interruptions (PCIs), and channel end and abnormal end EXCP returns. Data sets are not identified by name, but are recorded in the order of the step DD statements. An installation exit routine can record this order as each statement is validated if a report on data set activity is needed. The number of records in SYSIN data sets for the step is also included.

Job Termination Record (Type 5): This record is created at normal or abnormal job termination. The job is identified by job log number, programmer name, the installation-supplied user identification field, input class, requested priority, and the accounting fields from the JOB statement. Operating information includes the start and stop time for processing of the job by the reader/interpreter, and the device type and class of the reader device. The number of records in SYSIN data sets for the job and the number of steps in the job are included. Job CPU time equals the sum of the step CPU times. A checkpoint/restart indicator and the job completion code are recorded, along with a termination code indicating which SMF installation exit routine, if any, canceled the job. A flag marks each SYSOUT and MSGCLASS used by the job.

Output Writer Record (Type 6): At least one output writer record is produced for each SYSOUT class used by a job. If two or more forms are used within a class, one output writer record is produced for each form. The record is created when processing of a SYSOUT class completes, or when the form changes.

The output writer is identified by class and form number. The job is identified by job log number, the installation user identification, and the time the job was read in. Output writer activity is recorded by a count of the number of logical records processed, the number of SYSOUT data sets within the class and form, writer start and end times, and a code indicating what I/O errors, if any, occurred.

Note: If installation-written system output writers are present, they must supply the output writer record. If they do not, the record is not produced.

Data Lost Record (Type 7): This record is created whenever no SMF data set is available for recording. It contains a count of SMF records not written, and the start and end times of the period during which no records were written. The record is the first written when a SMF data set again becomes available.

I/O Configuration Record (Type 8): The I/O configuration record is created during system initialization (after IPL). It consists of the standard heading, and an entry describing each device online at IPL. Devices are identified by device class, unit type, and device address.

VARY ONLINE Record (Type 9): This record is created during processing of the VARY ONLINE operator command. It identifies the system resource being added to the configuration.

Allocation Recovery Record (Type 10): This record is created during allocation recovery. It identifies the device brought online, or otherwise made available, by device class, unit type, and device address. The job requiring the allocation is identified by job name, reader start time, and the user identification field. No allocation recovery record is produced if the operator cancels the job instead of attempting recovery.

VARY OFFLINE Record (Type 11): This record is produced during processing of the VARY OFFLINE operator command. It identifies the system resource being removed from the configuration.

End-of-Day Record (Type 12): This record is produced during processing of the HALT operator command. It records the system wait time since the last wait time record (type 1) was written.

Dynamic Storage Configuration Record (Type 13): For MFT only. This record is written at IPL and after each DEFINE command. It shows the amount of storage assigned to each active reader, writer and problem program partition.

INPUT or RDBACK Data Set Activity Record (Type 14): This record is written whenever a user data set opened for INPUT or RDBACK is closed or processed by end-of-volume (EOV).[1] It contains the data set name, creation and expiration dates, device type, EXCP count, data set indicator, data set organization, record format, record length, number of volumes, volume serial numbers, and additional information that depends on whether the data set is on a tape unit or a direct access device and on the access method used.

OUTPUT, UPDAT, INOUT or OUTIN Data Set Activity Record (Type 15): This record is written whenever a user data set opened for OUTPUT, UPDAT, INOUT or OUTIN is closed or processed by end-of-volume (EOV).[1] It contains the same information as the type 14 record.

SCRATCH Data Set Status Record (Type 17): This record is written whenever a user data set is scratched. It contains the data set

[1] For type 14 and 15 records, a user data set is a data set that is defined by a DD statement and opened by a user program. Type 14 and 15 records may be written for system data sets if the data sets are defined by DD statements and opened by a user program.

name, number of volumes, and volume serial numbers.

RENAME Data Set Status Record (Type 18): This record is written whenever any data set is renamed. It includes the old data set name, the new data set name, number of volumes, and volume serial numbers.

Direct Access Volume Record (Type 19): This record is written for all direct access devices on-line at IPL and HALT EOD and whenever a direct access volume is demounted. It contains the volume serial number, VTOC address, owner identification number, device type, number of unused alternate tracks, number of unallocated cylinders and tracks, number of cylinders and tracks in the largest free extent, and number of unallocated extents.

Note: Synchronization of clocks is essential in a shared file environment.

Job Commencement Record (Type 20): This record is written each time a job (not a job step) is initiated. It contains the record type, time stamp (time and date), CPU identification, job log number (job name, entry time, and entry date), programmer's name, user identification, number of accounting fields on the JOB statement, and accounting fields.

SYSOUT Records

In addition to the records written to the SMF data set, SMF writes four record types to the SYSOUT data set. These records are assigned message numbers IEF373I, IEF374I, IEF375I and IEF376I, and indicate, respectively, the start and end times for each step and for each job. The text of these records is described in Section 7.

Selecting SMF Records

The types of records written to the SMF data set can be controlled by four SMFDEFLT keyword parameters (MAN, OPT, DSV, and REC) and also, to a limited extent, by the installation-written exit routines. The use of the MAN, OPT, DSV and REC parameters to select SMF records is summarized in Figure 2. The contents of SMFDEFLT and the parameter formats are described in Section 4.

The MAN Keyword: The MAN keyword allows all SMF record types or all IBM-defined SMF record types to be suppressed. If MAN=NONE, no records are written to the SMF data set regardless of the values specified in the OPT and DSV parameters. If MAN=ALL, all SMF records are written unless suppressed by the OPT or DSV parameters or by an installation-written exit routine. If MAN=USER, all IBM-defined records (type 0 - 127) are suppressed but all installation-defined records (type 128 - 255) are written. This option allows the installation to define its own records but still make use of the SMF data management routines. By overriding the MAN=USER parameter at IPL, the full set of SMF records can be recorded when needed.

The OPT Keyword: The OPT keyword allows records containing step information to be suppressed. If OPT=2, record types 0 - 13 are written so that system, job and step information is available. If OPT=1, record type 4 is suppressed so that system and job (but not step) information is available. This option allows the installation to reduce its SMF overhead if step information is not required. By overriding the OPT=1 parameter at IPL, step information (record type 4) is available when needed.

The DSV Keyword: The DSV keyword allows records containing data set and direct access volume information to be suppressed. (Data set information is not available unless step information has been specified -- OPT=2.) If DSV=0, record types 14 - 20 are suppressed and neither data set nor direct access volume information is available. If DSV=1, record types 14, 15, 17, 18 and 20 are suppressed so that direct access volume information is available but data set information is not available. If DSV=2, record type 19 is suppressed so that data set information is available but direct access volume information is not available. If DSV=3, both data set and direct access volume information are available.

The REC Keyword: The REC keyword allows record type 17 to be written for temporary data sets[1]. If REC=0, record type 17 is written for non-temporary data sets only; it is not written for temporary data sets. If REC=2, record type 17 is written for temporary as well as non-temporary data sets.

Exit Routine Control: The final control over SMF data set contents is the code returned to the system from the step and job termination exit routines. After

[1] The system determines that a data set is temporary if either of the following criteria is met:

- The DD statement for the data set contains DISP=(NEW,DELETE).
- The DD statement for a new data set does not include the DSNAME parameter, or contains a parameter of the form DSNAME=&name.

Keyword	Value	Meaning	Effect on SMF Records
MAN	ALL	All SMF records.	Record types 0-255 are to be written to the SMF data set.
	NONE	No SMF records.	The SMF data set is not used.
	USER	Only user-formatted SMF records.	Only record types 128-255 are to be written to the SMF data set.
OPT	1	System and job (but not step) information.	Record types 0-3 and 5-13 are created but record type 4 is suppressed.
	2	System, job, and step information.	Record types 0-13 are created.
DSV	0	No information for data sets or direct access volumes.	Record types 14-20 are suppressed.
	1	Direct access volume information.	Record type 19 is created but record types 14, 15, 17, 18, and 20 are suppressed.
	2	Data set information.	Record type 19 is suppressed but record types 14, 15, 17, 18 and 20 are created.
	3	Data set and direct access volume information.	Record types 14-20 are created.
REC	0	No information for temporary data sets.	Record type 17 is created for non-temporary data sets only.
	2	Temporary data set information.	Record type 17 is created for temporary as well as non-temporary data sets.

Notes:
1. OPT must equal 2 if DSV is 1, 2 or 3.
2. There is no record type 16.

• Figure 2. Use of SMFDEFLT Keyword Parameters to Select SMF Records

inspecting the step or job termination record, the routine specifies whether or not it is to be written to the SMF data set. This option allows installations to selectively write records according to installation-defined algorithms.

Exits and Control Routines

The installation-written routines added to the control program through the SMF option allow monitoring of each job at specific points from the time it is encountered in the input stream to termination. These routines are called from "exits" in the job scheduler and nucleus, and may be referred to as exit routines. Like execution time for any other part of the control program, the execution time of the exit routines is added to system overhead, and will degrade system throughput. The amount of the degradation depends on the length of the routines and the number of times each is performed during processing of a job, and other options present in the control program. The advantages of including exit routines must be weighed against the factors affecting system throughput when choosing which exits to activate. Note that it is possible (by the SMFDEFLT parameters) to suppress all exits or step-related exits at IPL time, permitting the system to operate without the exit routines.

The following paragraphs discuss the general purpose of each exit, the information available to the exit routine, and when the exit routine is called. Figure 3 summarizes this information. The names in parentheses are mandatory load module and entry point names that must be assigned. Section 3 describes the formats of the parameters passed to each exit routine. Section 4 describes how to add exit routines to the system.

Exit Routine	Parameters Passed		Entered	Return
	Standard			
Job Validation (IEFUJV)	Job Name Time Stamp System ID User ID Step Number Installation Data Communication Field	JCL Image JCL Type	Each JCL Card Job Enqueue	Continue or Cancel
Job Initiation (IEFUJI)		Programmer Name Priority Account Flds.	Job Initiation	Continue or Cancel
Step Initiation (IEFUSI)		Step Name Program Name Step Accounting Flds.	Step Initiation	Continue or Cancel
SYSOUT Limit (IEFUSO)		Address of DCB	SYSOUT Limit Exceeded	Continue with new limit or cancel
Time Limit (IEFUTL)		Type Code	Job CPU Time Limit Exceeded Step CPU Time Limit Exceeded Job Wait Time Limit Exceeded	Continue with new time limit or Cancel
Termination (IEFACTRT)		Programmer Name Job CPU Time Job Accounting Flds. Step CPU Time Step Accounting Flds Completion Code SMF Record	Step Termination Job Termination	Continue or Cancel; Write or Skip SMF record

Figure 3. Installation Exit Routine Characteristics

Common Parameters and Attributes: Each exit routine receives the following standard parameters:

- Job log number, which includes the job name and the time of day and date when JCL for the job entered the system (the "timestamp").
- System identification.
- Step number.

Two additional fields are provided, one for communication among the installation routines, and one for an installation job identification to be included on job and step termination records and SYSOUT writer records originated by the job.

All exit routines must be reenterable. See IBM System/360 Operating System: Concepts and Facilities, Form C28-6535 for a discussion of the reenterable attribute.

Job Validation (IEFUJV): The job management portion of the control program exits to this installation-written routine just before the interpretation of each job control statement encountered in the input stream (or cataloged procedure). One final entry is made just before the job is queued for scheduling.

In addition to the standard calling parameters, IEFUJV receives the address of the job control statement to be scanned next. Another parameter indicates whether the current entry is for a null, JOB, EXEC, DD, or PROC job control statement, or if it is the final entry before queueing the job. The job validation routine may test and modify any of the fields in the job control statements, and indicate, through a return code passed to job management, whether processing of this job is to continue, or be terminated. Statements are processed in the same order in which they appear on the SYSOUT listing; that is, a statement overriding a statement within a cataloged procedure will immediately precede the statement being overridden. Editing of the job control statements must not result in additional job control statements or continuation cards.

An installation-written routine entered at this exit might perform any or all of the following:

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

Note that this routine, if present, is entered whenever a JCL statement is encountered in the job stream. Depending on the processing necessary, it may be more efficient to check JOB and EXEC statement accounting fields at the exits for job initiation and the first step initiation, respectively. The fields are passed as parameters to those routines, making a statement scan routine unnecessary. The user identification may also be assigned at either of those exits.

Job Initiation (IEFUJI): The initiator/terminator routine of the control program enters this installation-written routine whenever a job on the input queue is selected for initiation.

In addition to the standard parameters, IEFUJI is passed the programmer name field from the JOB statement, the requested job priority, and the accounting fields from the JOB statement. The accounting fields are placed in a formatted list for simple access. If the installation uses major and minor account numbers with several fields, this exit is easier to use for account number processing than IEFUJV because of the formatted list. The requested job priority can be changed by this exit routine.

A return code from this exit to the job initiation routine specifies whether the job is to be started or canceled.

Step Initiation (IEFUSI): The initiator/terminator enters the installation-written step initiation routine just before each step is started (prior to allocation). In addition to the standard parameters, the step initiation exit routine is passed the step name, program name, and step accounting fields from the EXEC statement. The accounting fields are listed in the same format as in the job initiation exit. A return code from this exit specifies whether the step is to be started or whether the job is to be canceled.

SYSOUT Limit (IEFUSO): The supervisor enters the SYSOUT limit routine whenever the number of logical records written to a SYSOUT data set exceeds the output limit specified by the OUTLIM parameter on the DD statement. The form for the new parameter is:

OUTLIM=number

where the number is the number of logical records plus the number of EXCPs issued by the OPEN and CLOSE macro instructions. (Refer to the publication Job Control Language, Form C28-6539.) If OUTLIM=number is not specified, or if OPT does not equal 2, output limiting is not done.

Parameters identifying the job and the SYSOUT data set are passed. The installation return code specifies whether the job is to be terminated or whether processing is to continue using a new SYSOUT limit.

Termination (IEFACTRT): The terminator enters the IEFACTRT routine on the normal or abnormal termination of each job and each job step.

In addition to the standard parameters, the termination exit receives the following information:

- Step name (from EXEC statement).
- Programmer's name (from JOB statement).
- Job CPU time.
- Job accounting fields (from JOB statement).
- Step CPU time.
- Step accounting fields (from EXEC statement).
- Flags and the step number.
- Job or step completion code.
- The SMF-formatted completion record that will be written to the SMF data set.

The parameters are passed in a list compatible with the parameter list supplied to installation accounting routines written for previous versions of the operating system. The interface for this exit is designed so that such previous accounting routines, which were also named IEFACTRT, will operate correctly under SMF, except that the routine is not entered at step initiation.

As with past versions of IEFACTRT, output may be directed to the console or SYSOUT. Under SMF, installation information may be added to the SMF data set, or written to an installation-supplied data set.

Return codes to the initiator/terminator specify whether the job is to be continued or terminated (for step entry only), and whether the SMF record is to be written or skipped.

Time Limit (IEFUTL): The timer second level interruption handler enters the time limit routine whenever one of the following time limits expires:

- The job execution time limit (from the JOB statement or reader procedure).
- The step execution time limit (from the EXEC statement or reader procedure).
- The job wait time limit (from SMFDEFLT).

If no time limit is specified in the JOB statement, the time limit value from the reader procedure is used. If no time limit is specified in the EXEC statement, the value from the reader procedure or the time remaining for the job (whichever is smaller) is used. Specifying TIME=1440 on the EXEC statement is invalid and the value is ignored. Specifying TIME=1440 on the JOB statement eliminates job timing.

The standard parameters identifying the job and step are passed and a code indicates which time limit has been exceeded. The installation routine return code specifies whether the job is to be terminated, or whether processing is to continue. If the job is to continue, the installation routine must supply an additional time limit value for the job. The expiration may be recorded in the SMF data set or with a message to the console or SYSOUT.

Data Management Considerations

The SMF data set may reside on a tape unit or 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 type of device is most efficient for a particular installation. Following are requirements for the SMF data set on each device type.

SYS1.MAN Resident on Tape

If the SMF data set is resident on tape only a primary data set (SYS1.MANX) is required. Specification of an alternate data set (SYS1.MANY) is unnecessary and therefore is ignored. The device must be online and ready during IPL, when it will be allocated to SMF. The device will not be available to other jobs in the system.

The data set may reside on as many as 20 tape volumes. The system IPL procedure must be repeated before the 20th volume is filled. At each IPL the volume count is reset to one and the tape is rewound; a new tape must be mounted before each IPL to avoid writing over previous SMF data.

When the SMF data set is resident on tape it may be used directly as input to installation report and analysis routines, without the intermediate use of the SMF dump program.

SYS1.MAN Resident on Direct Access

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

The devices used for the data sets are defined in SMFDEFLT, and will become permanently resident at IPL. The devices must be online and ready during IPL. Space for the SMF data sets must be allocated on appropriate volume types; the volumes must be mounted prior to IPL.

Switching between the primary and alternate data sets is automatic as each becomes filled. The SMF dump program must be used to transfer a full data set to tape. When a dump is completed, the status of the data set is reset to empty by the dump program.

SMF Buffer

SMF records are blocked internally in a buffer before they are written to the SMF data set. The buffer is logically divided into two equal parts to allow overlapping of blocking and writing. The buffer size is specified in the SMFDEFLT data set before or during IPL. Records will be spanned if the record length (LRECL) exceeds half the buffer size.

To avoid spanning of logical records across blocks, the buffer size should be twice the largest SMF record that may be generated. Minimum buffer size is 400 bytes. The size (in bytes) of the largest record produced by SMF routines is determined as follows:

SMF record size = 105 + (largest number of bytes of accounting information) + (the number of fields of accounting information) + 8 * (the number of devices allocated on all DD statements)

Note: The maximum length of records written to the SMF data set by exit routines depends on installation requirements.

Table 1 lists several buffer sizes and shows the corresponding number of "average" jobs whose records can be written on a single track of various direct access devices. For an "average" job (defined below) approximately 400 bytes of SMF data is recorded if all record types are included. An "average" job is defined as follows:

- 3 job steps.
- 4 DD statements per step.
- 10 bytes of accounting information in the JOB statement.
- 10 bytes of accounting information in the EXEC statement.
- 2 output writer forms.
- 2 output writer classes.
- 1 device for allocation recovery.
- 1 VARY ONLINE command.

Table 1. SMF Buffer Size and Utilization of Direct Access Space

Buffer Size	Physical Record Length	Jobs per Track of Various Devices			
		2301	2303	2311	2314
400	200	15.0	8.0	6.5	10.5
600	300	22.5	9.0	6.75	12.0
800	400	30.0	9.0	7.0	13.0
1000	500	30.75	10.0	7.5	13.75
1400	700	42.0	10.5	7.0	14.0
1800	900	42.75	9.0	6.75	13.5
2200	1100	44.0	11.0	8.25	13.75
2600	1300	45.5	9.75	6.5	13.0
3000	1500	45.0	11.25	7.5	15.0
4000	2000	45.0	10.0	5.0	15.0
6000	3000	45.0	7.5	7.5	15.0

Note: Figures have been adjusted to allow for inter-record gaps.

Section 3: Planning User Routines

This section contains information for planning and writing user routines. Parameter formats for exit routines and descriptions of sample assembler language exit routines are provided, along with procedures for testing exit routines. The SMFWTM macro instruction, which is used to write records to the SMF data set, is described. Guidance for writing report programs is provided in the form of sample JCL for executing the operating system sort/merge program, descriptions of sample sort/merge exit routines, and sample PL/I statements for a general report program.

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

- Exit routines that periodically monitor jobs; they may also write user records to the SMF data set or to any installation-defined data sets.

- Analysis and report routines that process and format information contained in the SMF and user data sets.

If further analysis of SMF data is desired, a more complex report program, or a separate analysis program, must be written. The use made of SMF data is determined solely by the requirements of the installation; consequently, no discussion of analysis procedures will be provided in this planning guide.

All sample programs discussed in this section will be included in SYS1.SAMPLIB. Procedures for obtaining listings and/or source decks of the routines from SAMPLIB are provided in this section.

SMFWTM Macro Instruction

You may use the SMFWTM macro instruction in exit routines to write a record to the SMF data set. Routines other than exit routines using the macro instruction must have a protection key of 0. Write the macro instruction as follows:

[symbol]	SMFWTM	{ record address }
		{ (r) }

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

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

The record to be written should include a standard SMF record header; see Section 6 for the header format. The record must be preceded by a record descriptor word (RDW); for a discussion of the RDW refer to the publication IBM System/360 Operating System: Supervisor and Data Management Services, Form C28-6646.

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

SMFWTM Return Codes: The SMFWTM macro instruction returns a code in register 15, indicating the disposition of the user record.

0 -- The record was written without error.

4 -- The record was truncated because it would not completely fit in an empty SMF data set.

8 -- The record was not written because the specified length was less than five bytes.

12 -- The record was not written because the routine was not authorized to write to the SMF data set. (The requesting routine had a nonzero protection key.)

16 -- The record was not written because: (a) the writing of records to the SMF data set is prohibited (i.e., MAN=NONE was specified in SMFDEFLT), or (b) the writing of records is allowed but the SMF data set was full. If (b), the data set must be dumped before additional SMF records can be written. (See "Section 5: Operational Considerations" for the procedure for executing the SMF dump program.)

Designing Exit Routines

The functions performed by your exit routines are determined solely by the requirements of your installation. Section 2 describes when each exit routine receives control and the information available to each exit. The following paragraphs describe exit routine restrictions, the formats of parameters passed to each exit, and the required return codes. Sample assembler language exit routines are provided in a member (SMFEXITS) of SYS1.SAMPLIB. Figure 4 illustrates the JCL required to retrieve a listing of these sample routines.

```
//PRINT      JOB 123456,SMITH
//          EXEC PGM=IEBPTPCH
//SYSPRINT   DD  SYSOUT=A
//SYSUT1     DD  DSNAME=SYS1.SAMPLIB,DISP=(OLD,KEEP),UNIT=2311,        X
//               VOLUME=SER=DLIB01
//SYSUT2     DD  SYSOUT=A
//SYSIN      DD  *
       PRINT    TYPORG=PO,MAXNAME=1,MAXFLDS=1
       MEMBER    NAME=SMFEXITS
       RECORD    FIELD=(80)
/*
```

Figure 4. Obtaining a Listing of Sample Installation Exit Routines

Exit Routine Facilities and Restrictions

Your exit routines must be written in reenterable code. They must save registers when they receive control and restore registers before returning to the control program. Register 13 will contain the address of the register save area, register 14 the return address, and register 15 the entry point address.

Your routines can communicate with each other via the user communication field and the user identification field passed to every exit routine. Any exit routine may obtain an additional work area by issuing a GETMAIN macro instruction specifying an appropriate subpool in system queue space as shown in Table 2. The address of the work area can be placed in the user communication field. (You must allocate storage for the work area at system generation.). The address of the work area can be placed in the user communication field.

Note: Communication areas obtained by exit IEFUJV will not be maintained if the system is restarted.

• Table 2. Characteristics of Subpools in System Queue Space

Subpool No.	Storage is Allocated:	Storage is De-allocated:
253	In system queue space (MVT) or high in the partition (MFT).	Automatically (by the System) when the task terminates (MVT) or when the partition is freed (MFT).
254	In system queue space (MVT) or high in the partition (MFT).	Automatically (by the system) when the task terminates (MVT) or when the partition is freed (MFT).
255	In system queue space (MVT or MFT).	Explicitly, by issuing a FREEMAIN macro instruction.

Note: Only routines having a protect key of zero can obtain storage in system queue space (MVT or MFT) or high in a partition (MFT).

If any of your exit routines cancels a job, a job termination record (type 5) will be written to the SMF data set (assuming the writing of records is permitted); this record indicates which routine canceled the job. If you require this information in the SYSOUT data set, the exit routine must write an appropriate record to SYSOUT before it returns control to the system.

Your routines can write to the SMF data set via the SMFWTM macro instruction. If you wish to use your own data sets, you must define them as follows: a data set used by routine IEFUJV requires a DD statement in the reader/interpreter cataloged procedure, a data set used by routines IEFUJI, IEFUSO, IEFUSI, IEFACTRT, and IEFUTL requires a DD statement in the initiator procedure.

Common Exit Parameters

When an exit routine receives control, register 1 points to a list of four-byte addresses. The first entry in the list is common to all exit routines; it points to a 36 byte parameter area having the format described in Table 3.

• Table 3. Common Exit Parameter Area

Displacement from Pointer	Length (in bytes)	Data Format	Description
0	8	EBCDIC	Job name
8	4	Binary	Reader start time (in hundredths of seconds) for the job.
12	4	Packed Decimal	Reader start date for the job. Format is 00YYDDDF. (F is a sign.)
16	4	EBCDIC	System identifier (2 bytes) and model number (2 bytes).
20	8	EBCDIC	User identification field.
28	4	Binary	Number of the step being processed, in high-order byte. The other 3 bytes are reserved.
32	4	Binary	User communication field.

IEFUJV -- Job Validation Exit

Exit routine IEFUJV receives control just before each JCL statement for a job is interpreted, and again just before the job is queued for scheduling. At entry to the routine register 1 points to a list of four-byte addresses in the following order:

1. The address of the common parameter area. (See Table 3.)

2. The address of an 80 character JCL statement image (in EBCDIC). Statements are presented to the exit in the order in which they are listed in the SYSOUT data set. If a cataloged procedure is being executed, the sequence of statements is JOB, EXEC PROC=..., EXEC PGM=..., followed by the other statements of the procedure. Override statements immediately precede the statement being overridden.

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

   0 - Null statement.

   1 - JOB statement.

2 - EXEC statement.

4 - DD statement.

8 - PROC statement (for symbolic parameter definition).

16 - No statement is being presented; this is the job enqueue entry to the exit routine.

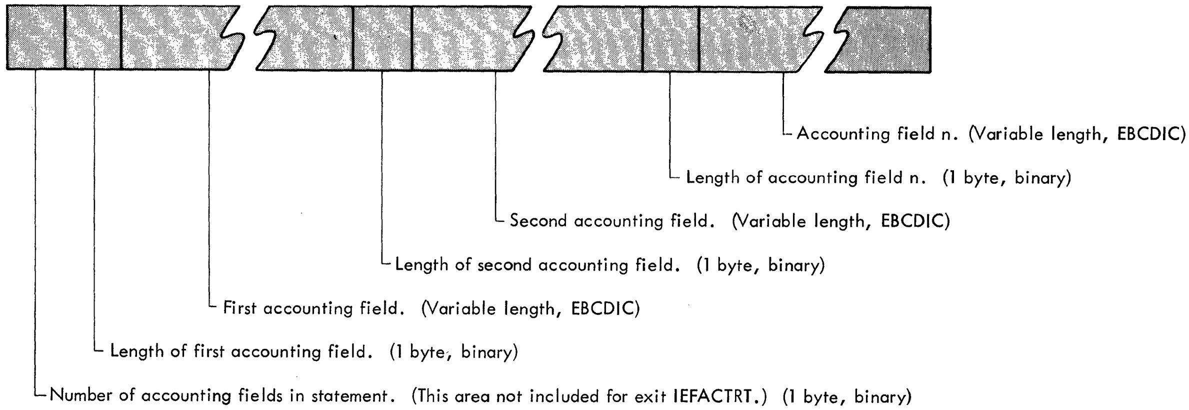
Before the IEFUJV exit routine returns to the control program, it must place a return code in register 15: 0 indicates that processing of the job should continue; binary 4 indicates that the job is to be canceled.

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

IEFUJI -- Job Initiation Exit

Exit routine IEFUJI receives control just before each job is initiated. At entry to the routine, register 1 points to a list of four-byte addresses in the following order:

1. The address of the common parameter area. (See Table 3.)
2. The address of a 20-byte area containing the programmer's name (in EBCDIC). This area is left-justified and, if necessary, padded with blanks.
3. The address of a one-byte area indicating the requested priority.
4. The address of an area containing accounting information from the JOB statement. The area has the format illustrated in Figure 5. If the JOB statement contains no accounting information the area is one byte of zeroes.


Accounting field n. (Variable length, EBCDIC)
Length of accounting field n. (1 byte, binary)
Second accounting field. (Variable length, EBCDIC)
Length of second accounting field. (1 byte, binary)
First accounting field. (Variable length, EBCDIC)
Length of first accounting field. (1 byte, binary)
Number of accounting fields in statement. (This area not included for exit IEFACTRT.) (1 byte, binary)

• Figure 5. Format of Accounting Field Parameter

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

Sample IEFUJI Routine: The sample IEFUJI exit routine provided in SYS1.SAMPLIB determines how long a job has been in the input job queue before it is initiated. This value and the job priority are written to the SMF data set.

IEFUSI -- Step Initiation Exit

Exit routine IEFUSI receives control just before each job step is initiated. (If OPT=1 was specified in the SMFDEFLT data set, this exit is not taken.) At entry to the routine, register 1 points to a list of four-byte addresses in the following order:

1. The address of the common parameter area. (See Table 3.)

2. The address of an eight-byte area containing the step name (in EBCDIC). This area is left-justified and, if necessary, padded with blanks.

3. The address of an eight-byte area containing the program name (in EBCDIC). This area is left-justified and, if necessary, padded with blanks.

4. The address of an area containing accounting information from the EXEC statement. This area has the format illustrated in Figure 4. If the EXEC statement contains no accounting information, the field is one byte of zeroes.

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

No sample IEFUSI routine is provided in SYS1.SAMPLIB.

IEFUSO -- SYSOUT Limit

Exit routine IEFUSO receives control when the number of logical records written to a SYSOUT data set exceeds the output limit specified by the OUTLIM parameter on the DD statement. The form for the new parameter is:

OUTLIM=number

where the number must include the number of EXCPs from the OPEN and CLOSE macro instructions. (Refer to the publication Job Control Language, Form C28-6539.) If OUTLIM=number is not specified, or if OPT does not equal 2, output limiting is not done.

At entry to IEFUSO, register 1 points to a list of 4-byte addresses in the following format:

1. The address of the common exit parameter area. (See Table 2.)

2. The address of the DCB for the data set.

Before the IEFUSO routine returns control to the control program it must place a return code in register 15: 0 indicates that the step is to be terminated; binary 4 indicates that the output limit is to be increased by a value placed in register 1.

Note: Unless the output limit has been increased, the exit to the installation routine will again be taken when the next record is written to this SYSOUT data set.

No sample IEFUSO routine is provided in SYS1. SAMPLIB.

IEFACTRT -- Termination Exit

Exit routine IEFACTRT receives control when execution of a job or job step is terminated. (If OPT = 1 was specified in the SMFDEFLT data set, this exit is taken only at job termination.) At entry to the routine register 1 points to a list of four-byte addresses in the following format:

1. The address of the common parameter area. (See Table 3.) At entry for job termination the parameter at displacement 28 contains the number of steps in the job.

2. The address of an eight-byte area containing the step name (in EBCDIC). This area is left-justified and, if necessary, padded with blanks. At job termination the address is zero.

3. The address of a 20-byte area containing the programmer's name (in EBCDIC). This area is left-justified and, if necessary, padded with blanks.

4. The address of a four-byte area containing, in the first three bytes, job CPU time in hundredths of seconds (a binary value). The last byte contains the number (binary) of accounting fields in the JOB statement.

5. The address of an area containing accounting information from the JOB statement. This area has the format illustrated in Figure 4, excluding the first field illustrated (the number of accounting fields). If the JOB statement contains no accounting information, the area is one byte of zeroes.

6. The address of a four-byte area containing, in the first three bytes, step CPU time in hundredths of second (a binary value). The last byte contains the number (binary) of accounting fields in the EXEC statement. At job termination the address is zero.

7. The address of an area containing accounting information from the EXEC statement. This area has the format illustrated in Figure 4, excluding the first field illustrated (the number of accounting fields). If the EXEC statement contains no accounting information, the area is one byte of zeroes. 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.

9. The address of a two-byte area containing the termination status (condition code) of the job or job step.

10. The address of an area containing the step termination record (type 4) or job termination record (type 5) to be written to the SMF data set.

At entry to the routine register 0 contains a binary code indicating the reason for entry: 12 = step termination, 16 = job termination.

If your IEFACTRT routine writes to the SYSPRINT device, the contents of register 12 must be the same as when the routine was entered, register 13 must contain the address of a 64 fullword area, and the technique illustrated in Figure 6 must be used.

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

```
Register 1  = 4 -- Do not write the termination record on the SMF
                   data set.
Register 1  ≠ 4 -- Write the termination record.
Register 15 = 4 -- Cancel the job.
Register 15 ≠ 4 -- Continue processing the job.
```

Sample IEFACTRT Routine: The sample IEFACTRT routine provided in SYS1.SAMPLIB changes the SMF job termination and step termination records to user records and attempts to write them to the SMF data set. If the data set is full, a message indicating lost SMF records is written to the console. At step termination the routine writes to the SMF data set a record containing all DD names used by the step. At job termination a record containing the job name, programmer's name, and account number is written to the SYSOUT device.

```
          .
          .
          .
          MVC    36(4,12),MSGADDR
          MVC    42(2,12),MSGLEN
          L      REG15,VIEFYS
          BALR   REG14,REG15
          .
          .
          .
MSGADDR   DC     A(MSG)
MSG       DC     C'message text'
MSGLEN    DC     H'xx' message length
VIEFYS    DC     V(IEFYS)
```

Figure 6. Writing to SYSPRINT from IEFACTRT During Testing

IEFUTL -- Time Limit Exit

Exit routine IEFUTL receives control whenever the job CPU, step CPU, or job wait time limits expire. At entry to the routine, register 1 points to the address of the common parameter area. (See Table 3.) Register 0 contains a binary code indicating which time limit has expired:

```
0 = job CPU time limit
4 = step CPU time limit
8 = job wait time limit
```

Before the IEFUTL routine returns to the control program it must place a return code in register 15: 0 indicates that processing is to continue (i.e., the job is to be canceled); binary 4 indicates that the job is to continue processing with an additional time allocation. The additional time (in timer units) must be placed in register 1. The number of timer units is determined via the following algorithm: units = milliseconds/26.04.

Sample IEFUTL Routine: The sample IEFUTL routine provided in SYS1.SAMPLIB causes a job to be terminated if the job CPU time limit or step CPU time limit has been exceeded. If the job wait time limit has been exceeded, the limit is extended twice; on the third entry for job wait time limit exceeded, the job is canceled. Each time the routine is entered for job wait time limit exceeded, it writes to the SMF data set a record describing the action taken.

Testing Exit Routines

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

Special Requirements for Testing Exit Routines

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

- Subpool 0 must be specified in GETMAIN macro instructions. (When testing is completed, subpool 253, 254 or 255 must be specified for the area used to communicate between exit routines as shown in Table 2.)

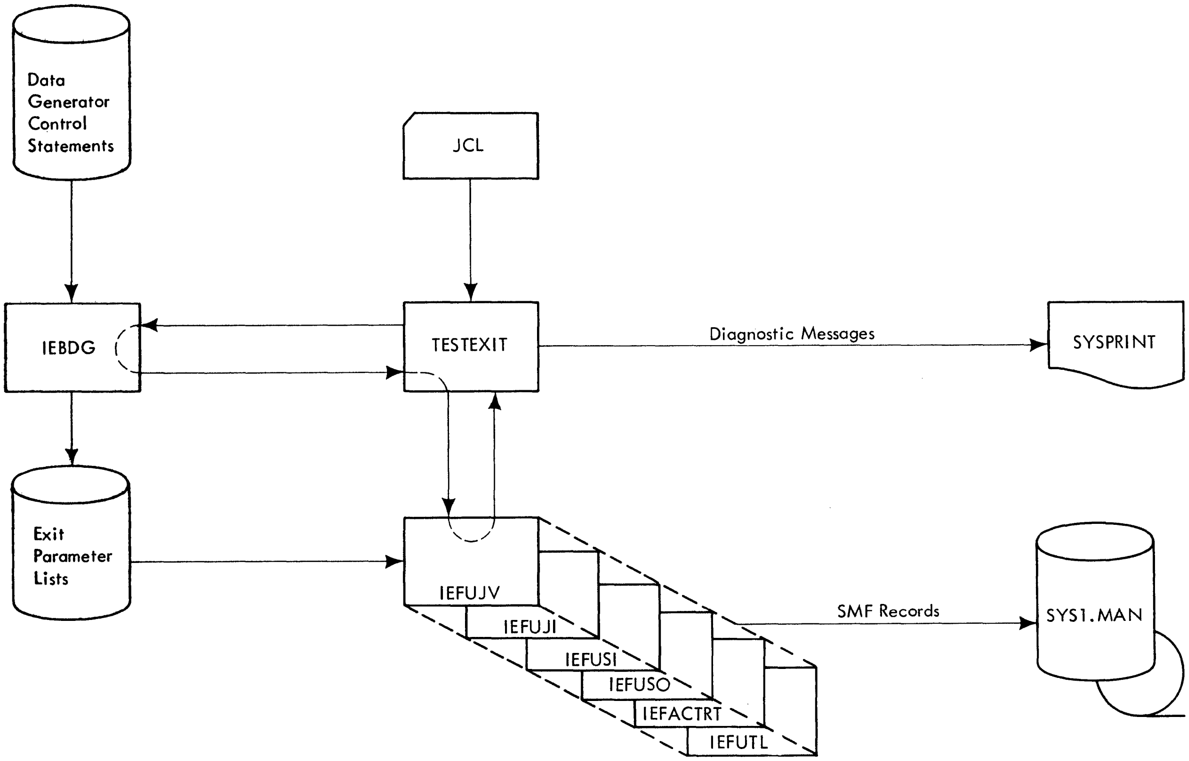
- If the SMFWTM macro instruction is used in any of your routines, you must provide a special macro definition in the routine. (See Figure 7.) When testing is completed the macro definition must be removed.

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

Figure 7. SMFWTM Macro Definition Required for Testing

The TESTEXIT Procedure

Included in the test procedure is an assembler language source program (also named TESTEXIT). This source program attaches the data generator utility program (IEBDG) to create sample parameter lists; TESTEXIT then calls each exit routine being tested, passing to it the appropriate parameter list. Figure 8 illustrates the I/O and control flow of the TESTEXIT routine.



• Figure 8. TESTEXIT Control and I/O Flow

Figure 9 illustrates the JCL included in the TESTEXIT procedure. Following is a summary of the operations performed by the procedure:

- TESTEXIT -- This job assembles the TESTEXIT routine (not illustrated in Figure 9) and link edits it with the exit routines being tested. The exit routines must reside in a partitioned data set (EXITLIB); Figure 10 illustrates the procedure for placing your exit routines in EXITLIB.

- DATAGEN -- This job creates (via the IEBUPDTE utility program) a partitioned data set (DGINPUT) containing control statements for the IEBDG utility program, which will be attached by the TESTEXIT program. The control statements are not illustrated in Figure 9.

- TESTING -- This job includes the execution of the TESTEXIT program.

```
//TESTEXIT      JOB  MSGLEVEL=1
//TEST          EXEC ASMFCL
//ASM.SYSIN     DD   *
    .
    .  (TESTEXIT Source Module)
    .
/*
//LKED.SYSLMOD DD   DSNAME=TESTLIB,VOLUME=SER=231100,UNIT=2311,        *
//                  SPACE=(TRK,(10,2,1)),DISP=(NEW,KEEP)
//LKED.EXITS    DD   DSNAME=EXITLIB,VOLUME=SER=231100,UNIT=2311,        *
//                   DISP=OLD
//LKED.SYSIN    DD   *
       INCLUDE EXITS(IEFUJV,IEFUJI,IEFUSI,IEFUSO,IEFACTRT,IEFUTL)
       NAME TESTEXIT
/*
//DATAGEN       JOB  MSGLEVEL=1
//              EXEC PGM=IEBUPDTE,PARM=NEW
//SYSUT2        DD   DSNAME=DGINPUT,UNIT=2311,DISP=(,KEEP),             *
//                   VOLUME=SER=231100,SPACE=(TRK,(20,5,1)),            *
//                   DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//SYSPRINT      DD   SYSOUT=A
//SYSIN         DD   DATA
./  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=ACT
    .
    .  (IEBDG Control Statements for IEFACTRT)
    .
./  ENDUP
/*
//TESTING       JOB  MSGLEVEL=1
//JOBLIB        DD   DSNAME=TESTLIB,VOLUME=SER=231100,UNIT=2311,        *
//                   DISP=(OLD,KEEP)
//              EXEC PGM=TESTEXIT,PARM='UJV=26,UJI=10,UTL=5,ACT=2'
//INUJV         DD   DSNAME=DGINPUT(UJV),DCB=(LRECL=80,                 *
//                   BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),UNIT=2311,   *
//                   VOLUME=SER=231100
//INUJI         DD   DSNAME=DGINPUT(UJI),DCB=(LRECL=80,                 *
//                   BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),UNIT=2311,   *
//                   VOLUME=SER=231100
//INUSI         DD   DSNAME=DGINPUT(USI),DCB=(LRECL=80,                 *
//                   BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),UNIT=2311,   *
//                   VOLUME=SER=231100
```

• Figure 9. TESTEXIT Procedure JCL (Part 1 of 2)

```
//INUSO        DD   DSNAME=DGINPUT(USO),DCB=(LRECL=80,                    *
//                  BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),UNIT=2311,      *
//                  VOLUME=SER=231100
//INUTL        DD   DSNAME=DGINPUT(UTL),DCB=(LRECL=80,                    *
//                  BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),UNIT=2311,      *
//                  VOLUME=SER=231100
//INACT        DD   DSNAME=DGINPUT(ACT),DCB=(LRECL=80,                    *
//                  BLKSIZE=400,RECFM=FB),DISP=(OLD,PASS),UNIT=2311,      *
//                  VOLUME=SER=231100
//OUTUJV       DD   DSNAME=UJV(OUT),UNIT=2311,DISP=(,PASS),               *
//                  SPACE=(TRK,(20,5,1)),VOLUME=SER=231100,               *
//                  DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//OUTUJI       DD   DSNAME=UJI(OUT),UNIT=2311,DISP=(,PASS),               *
//                  SPACE=(TRK,(20,5,1)),VOLUME=SER=231100,               *
//                  DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//OUTUSI       DD   DSNAME=USI(OUT),UNIT=2311,DISP=(,PASS),               *
//                  SPACE=(TRK,(20,5,1)),VOLUME=SER=231100,               *
//                  DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//OUTUSO       DD   DSNAME=USO(OUT),UNIT=2311,DISP=(,PASS),               *
//                  SPACE=(TRK,(20,5,1)),VOLUME=SER=231100,               *
//                  DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//OUTUTL       DD   DSNAME=UTL(OUT),UNIT=2311,DISP=(,PASS),               *
//                  SPACE=(TRK,(20,5,1)),VOLUME=SER=231100,               *
//                  DCB=(LRECL=80,BLKSIZE=400,RECFM=FB)
//OUTACT       DD   DSNAME=ACT(OUT),UNIT=2311,DISP=(,PASS),               *
//                  SPACE=(TRK,(20,5,1)),VOLUME=SER=231100,               *
//                  DCB=(LRECL=130,BLKSIZE=130,RECFM=FB)
//MANX         DD   UNIT=2311,VOLUME=SER=231100,SPACE=(TRK,(5,1)),        *
//                  DISP=(NEW,PASS),DCB=(BLKSIZE=144,LRECL=140)
//SYSPRINT     DD   SYSOUT=A,DCB=(BLKSIZE=136,LRECL=132)
//DGPRINT      DD   SYSOUT=A
//SYSABEND     DD   SYSOUT=A
/*
```

• Figure 9. TESTEXIT Procedure JCL (Part 2 of 2)

Using TESTEXIT

To use the TESTEXIT procedure you must do the following:

- Place your exit routines into a partitioned data set. (See Figure 10.)
- Obtain a punched deck of TESTEXIT. (See Figure 11.)
- Modify the procedure to meet your testing requirements.
- Execute the three jobs in the procedure.

The procedure provided in SYS1.SAMPLIB, without modification, will link edit the sample exit routines (also in SAMPLIB), generate sample parameter lists and test the sample exit routines. You should consider the following modifications in adapting the procedure to your testing requirements. (Refer also to Figure 9.)

Linkage Editor Modifications: The linkage editor step of the first job (TESTEXIT) in the procedure link edits the TESTEXIT program with the exit routines. You must substitute an INCLUDE control statement specifying the names of the exit routines you are testing.

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

Note that you must provide control statements in such an order that the records subsequently generated by the IEBDG utility will be grouped as complete parameter lists which conform in length and format to the exit parameters previously defined in this section. (The code passed to exits IEFACTRT and IEFUTL in register 0 must be included as a one-byte parameter at the end of the parameter lists for those exits.) For detailed information on the use of IEBDG control statements, refer to the publication IBM System/360 Operating System: Utilities, Form C28-6586.

```
//UPDTE         JOB  MSGLEVEL=1
//              EXEC PGM=IEBUPDTE,PARM=NEW
//SYSUT2        DD   DSNAME=EXITLIB,VOLUME=SER=231100,UNIT=2311,          *
//                   SPACE=(TRK,(20,5,1)),DCB=(LRECL=80,BLKSIZE=400,      *
//                   RECFM=FB)
//SYSPRINT      DD   SYSOUT=A
//SYSIN         DD   DATA
./  ADD         NAME=IEFUJV
    .
    .  (IEFUJV object deck)
    .
./  ADD         NAME=IEFUJI
    .
    .  (IEFUJI object deck)
    .
./  ADD         NAME=IEFUSI
    .
    .  (IEFUSI object deck)
    .
./  ADD         NAME=IEFUSO
    .
    .  (IEFUSO object deck)
    .
./  ADD         NAME=IEFUTL
    .
    .  (IEFUTL object deck)
    .
./  ADD         NAME=IEFACTRT
    .
    .  (IEFACTRT object deck)
    .
./  ENDUP
/*
```

• Figure 10. Entering Exit Routines into EXITLIB

```
//PUNCH    JOB   MSGLEVEL=1
//         EXEC  PGM=IEBPTPCH
//SYSPRINT DD    SYSOUT=A
//SYSUT1   DD    DSNAME=SYS1.SAMPLIB,DISP=(OLD,KEEP),                  *
//               UNIT=2311,VOLUME=SER=DLIB01
//SYSUT2   DD    UNIT=2540-2
//SYSIN    DD    *
      PUNCH TYPORG=PO,MAXNAME=1,MAXFLDS=1
      MEMBER  NAME=TESTEXIT
      RECORD  FIELD=(80)
/*
```

• Figure 11. Obtaining a Punched Deck of TESTEXIT

TESTEXIT Execution Modifications: The third job (TESTING) of the procedure includes execution of the TESTEXIT program. Values for the PARM parameter of the EXEC statement specify which exit routines are to be tested and the number of times each is to be tested. This parameter has the format PARM='xxx=nnn,...,xxx=nnn' where nnn is the number of times an exit routine is to be tested (maximum value 255), and xxx is one of the exit routine identifiers shown in Table 4.

Depending on the exit routines being tested, you may omit certain DD statements. Table 4 specifies which DD statements must be included. You must also include in the TESTEXIT JCL DD statements for any other data sets used by your exit routines.

• Table 4. Parameters and DD Statements for Executing TESTEXIT

If you are testing this exit routine	Use this identifier in the EXEC statement	And include these DD statements
IEFUJV	UJV	INUJV OUTUJV
IEFUJI	UJI	INUJI OUTUJI
IEFUSI	USI	INUSI OUTUSI
IEFUSO	USO	INUSO OUTUSO
IEFUTL	UTL	INUTL OUTUTL
IEFACTRT	ACT	INACT OUTACT
Any		MANX SYSPRINT DGPRINT SYSABEND

Report Programs

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

Sorting SMF Records

You may use the IBM System/360 Operating System Sort/Merge Program to sort SMF records. Your report format and your analysis requirements will determine the fields on which to sort and the sorting sequence. Sort exit E15 allows you to extract or delete selected records as the SMF dump data set is sorted, and sort exit E35 allows you to insert records into the final sorted output data set.

Sample Sort Exit Routines: Three sample sort exit routines are provided in SYS1.SAMPLIB. The routine for exit E15 extracts all system-oriented SMF records (i.e., records without a job log number) from the SMF dump dat set. Dump header and dump trailer records (SMF record types 2 and 3) are retained in a temporary data set (ddname HDRDATA), and all other system-oriented records are retained on another temporary data set (ddname SORDATA). Only job-oriented records (i.e., records having a job log number) will be sorted.

The first sample routine for exit E35 places in the sort output data set all records extracted by the E15 exit routine. The system-oriented records will precede the sorted job-oriented records in the output data set, and will be inserted in the following order: dump header records, dump trailer records, and all other system records.

The second sample routine for exit E35 also places in the sort output data set the system-oriented records extracted by the E15 exit routine. Dump header and dump trailer records will be inserted before all job-oriented records. All other system records will be inserted after the job records relating to the job being processed at the time the system records were written.

An example of the JCL required to execute the sort/merge program is also provided in SYS1.SAMPLIB. You may obtain a listing of the sample sort exit routines and sort JCL from SYS1.SAMPLIB by executing the control statements shown in Figure 12.

```
//PRINT  JOB   123456,SMITH
//        EXEC  PGM=IEBPTPCH
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   DSNAME=SYS1.SAMPLIB,DISP=(OLD,KEEP),                   *
//              UNIT=2311,VOLUME=SER=DLIB01
//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=SMFE35A
      RECORD  FIELD=(80)
      MEMBER  NAME=SMFE35B
      RECORD  FIELD=(80)
/*
```

• Figure 12. Obtaining a Listing of Sample Sort Exit Routines

Sort Procedure: If you plan to include the sample exit routines in your sort application, you must assemble and link edit them before executing the sort/merge program. Figure 13 illustrates this procedure, including one possible sort application. In this example, SMF records are to be sorted first on the job log number (major control field), and then on the date and time portions of the time stamp (minor control fields). Displacements of these fields (from the beginning of the record) are 19, 11, and 7 bytes, respectively.

For a description of the cataloged procedure ASMFCL (assemble and link edit) refer to the publication IBM System/360 Operating System: Assembler (F) Programmer's Guide, Form C26-3756. For a detailed discussion of the sort/merge program refer to the publication IBM System/360 Operating System: Sort/Merge, Form C28-6543.

```
//SMFSORT  JOB  MSGLEVEL=1
//STEP1  EXEC  ASMFCL                                                 1
//ASM.SYSIN DD *
       E15 SOURCE DECK
/*
//LKED.SYSLMOD DD DSNAME=SMF1.EXITS,UNIT=2311,                         2
//         DISP=(,PASS),SPACE=(TRK,(10,1)),VOLUME=SER=231100
//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,PASS)                   2
//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,DELETE)                    5
//SORTIN   DD  UNIT=2400,VOLUME=SER=SYSMAN,DISP=OLD,                  6
//         LABEL=(,NL),DCB=(RECFM=VBS,LRECL=200,BLKSIZE=204)
//SORTWK01 DD UNIT=2311,SPACE=(TRK,(100),,CONTIG)                     7
//SORTWK02 DD UNIT=2311,SPACE=(TRK,(100),,CONTIG)                     7
//SORTWK03 DD UNIT=2311,SPACE=(TRK,(100),,CONTIG)                     7
//SORTOUT  DD UNIT=2400,DSNAME=SMF1.SORTOUT,LABEL=(,NL),              8
//         DISP=(,KEEP),DCB=(RECFM=VBS,LRECL=200,BLKSIZE=204)
//SORDATA  DD  UNIT=SYSDA,SPACE=(CYL,(1,1)),                          9
//             DCB=(RECFM=VBS,LRECL=200,BLKSIZE=204)
//HDRDATA  DD  UNIT=SYSDA,SPACE=(TRK,(5,5)),                          9
//             DCB=(RECFM=VBS,LRECL=200,BLKSIZE=204)
//SYSIN  DD    *
 SORT  FIELDS=(19,16,A,11,4,A,7,4,A),FORMAT=BI,SIZE=E4000            10
 MODS  E15=(E15,700,EXITLIB,N),E35=(E35,700,EXITLIB,N)               10
 END
/*
```

1. EXEC statements for cataloged procedure ASMFCL (assemble and link edit).
2. The E15 and E35 sample exit routines will be link edited into data set SMF1.EXITS.
3. Linkage editor control statements specifying that E15 and E35 will be the load module names of the exit routines.
4. EXEC statement for the sort/merge program.
5. Data set SMF1.EXITS is specified as the library in which sort exit routines may be found.
6. Input to the sort program is the SMF dump data set, contained on a tape having a volume serial number of SYSMAN.
7. Three sort work units are defined as being direct access devices.
8. The sort output data set is to be written on tape.
9. Two data sets required by the sample sort exit routines are defined on direct access devices.
10. The sort/merge control statements define the sort control fields and exit routines to be used in this sort application.

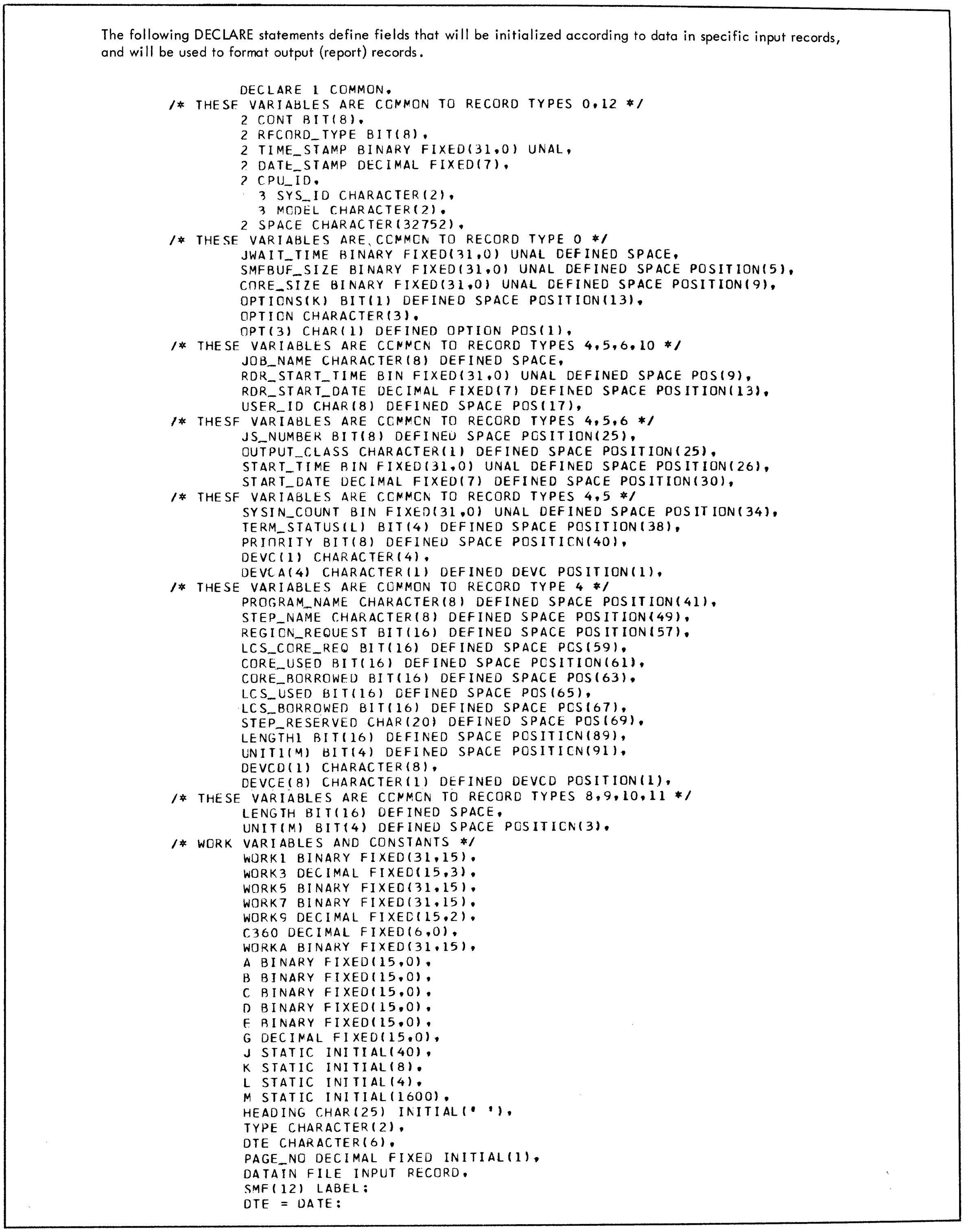
• Figure 13. Sample Sort Procedure

Designing a Report Program

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

A sample source report program (SMFPOST), written in PL/I, is provided in SYS1.SAMPLIB. This program processes SMF record types 0-12. Data from each input record is converted to an appropriate format and written as an output record. Figure 14 illustrates how SMFPOST uses DECLARE, assignment, and PUT EDIT statements to format SMF record types 0, 4, and 8. Figure 15 illustrates sample output from the SMFPOST program.

If you plan to modify the sample program or to write your own report program in PL/I, refer to the following publications for detailed information on PL/I: IBM System/360 Operating System: PL/I Reference Manual, Form C28-8201, and IBM System/360 Operating System: PL/I (F) Programmer's Guide, Form C28-6594.



The following DECLARE statements define fields that will be initialized according to data in specific input records, and will be used to format output (report) records.

```
                DECLARE 1 COMMON,
        /* THESF VARIABLES ARE COMMON TO RECORD TYPES 0,12 */
                2 CONT BIT(8),
                2 RFCORD_TYPE BIT(8),
                2 TIME_STAMP BINARY FIXED(31,0) UNAL,
                ? DATE_STAMP DECIMAL FIXED(7),
                ? CPU_ID,
                  3 SYS_ID CHARACTER(2),
                  3 MODEL CHARACTER(2),
                2 SPACE CHARACTER(32752),
        /* THESE VARIABLES ARE COMMCN TO RECORD TYPE 0 */
                JWAIT_TIME BINARY FIXED(31,0) UNAL DEFINED SPACE,
                SMFBUF_SIZE BINARY FIXED(31,0) UNAL DEFINED SPACE POSITION(5),
                CORE_SIZE BINARY FIXED(31,0) UNAL DEFINED SPACE POSITION(9),
                OPTICNS(K) BIT(1) DEFINED SPACE POSITION(13),
                OPTICN CHARACTER(3),
                OPT(3) CHAR(1) DEFINED OPTION POS(1),
        /* THESE VARIABLES ARE CCMMCN TO RECORD TYPES 4,5,6,10 */
                JOB_NAME CHARACTER(8) DEFINED SPACE,
                RDR_START_TIME BIN FIXED(31,0) UNAL DEFINED SPACE POS(9),
                RDR_START_DATE DECIMAL FIXED(7) DEFINED SPACE POSITION(13),
                USER_ID CHAR(8) DEFINED SPACE POS(17),
        /* THESF VARIABLES ARE CCMMCN TO RECORD TYPES 4,5,6 */
                JS_NUMBER BIT(8) DEFINED SPACE PCSITION(25),
                OUTPUT_CLASS CHARACTER(1) DEFINED SPACE POSITION(25),
                START_TIME BIN FIXED(31,0) UNAL DEFINED SPACE POSITION(26),
                START_DATE DECIMAL FIXED(7) DEFINED SPACE POSITION(30),
        /* THESF VARIABLES ARE CCMMCN TO RECORD TYPES 4,5 */
                SYSIN_COUNT BIN FIXED(31,0) UNAL DEFINED SPACE POSITION(34),
                TERM_STATUS(L) BIT(4) DEFINED SPACE POSITION(38),
                PRIORITY BIT(8) DEFINED SPACE POSITICN(40),
                DEVC(1) CHARACTER(4),
                DEVCA(4) CHARACTER(1) DEFINED DEVC POSITION(1),
        /* THESE VARIABLES ARE COMMON TO RECORD TYPE 4 */
                PROGRAM_NAME CHARACTER(8) DEFINED SPACE POSITION(41),
                STEP_NAME CHARACTER(8) DEFINED SPACE POSITION(49),
                REGICN_REQUEST BIT(16) DEFINED SPACE POSITION(57),
                LCS_CORE_REQ BIT(16) DEFINED SPACE PCS(59),
                CORE_USED BIT(16) DEFINED SPACE PCSITION(61),
                CORE_BORROWED BIT(16) DEFINED SPACE POS(63),
                LCS_USED BIT(16) DEFINED SPACE POS(65),
                LCS_BORROWED BIT(16) DEFINED SPACE PCS(67),
                STEP_RESERVED CHAR(20) DEFINED SPACE POS(69),
                LENGTH1 BIT(16) DEFINED SPACE PCSITICN(89),
                UNIT1(M) BIT(4) DEFINED SPACE POSITICN(91),
                DEVCD(1) CHARACTER(8),
                DEVCE(8) CHARACTER(1) DEFINED DEVCD POSITION(1),
        /* THESE VARIABLES ARE CCMMCN TO RECORD TYPES 8,9,10,11 */
                LENGTH BIT(16) DEFINED SPACE,
                UNIT(M) BIT(4) DEFINED SPACE PCSITICN(3),
        /* WORK VARIABLES AND CONSTANTS */
                WORK1 BINARY FIXED(31,15),
                WORK3 DECIMAL FIXED(15,3),
                WORK5 BINARY FIXED(31,15),
                WORK7 BINARY FIXED(31,15),
                WORK9 DECIMAL FIXED(15,2),
                C360 DECIMAL FIXED(6,0),
                WORKA BINARY FIXED(31,15),
                A BINARY FIXED(15,0),
                B BINARY FIXED(15,0),
                C BINARY FIXED(15,0),
                D BINARY FIXED(15,0),
                E BINARY FIXED(15,0),
                G DECIMAL FIXED(15,0),
                J STATIC INITIAL(40),
                K STATIC INITIAL(8),
                L STATIC INITIAL(4),
                M STATIC INITIAL(1600),
                HEADING CHAR(25) INITIAL(' '),
                TYPE CHARACTER(2),
                DTE CHARACTER(6),
                PAGE_NO DECIMAL FIXED INITIAL(1),
                DATAIN FILE INPUT RECORD,
                SMF(12) LABEL:
                DTE = DATE:
```

• Figure 14. Sample PL/I Report Program (Part 1 of 3)

Following are the assignment statements, processing, and the PUT EDIT statement for SMF record type 0.

```
         WORK3 = TIMF_STAMP:
         WORK1 = WCRK3/C360:
         K = 8:
         M = 1:
         N = 1:
         CPTICN = ' ':
 SMFOPT: IF M¬>3 THEN IF CPTIONS(M) = '1'B
         THEN DC:
         OPT(N) = SLBSTR('JSE',M,1);
         M = M + 1:
         N = N + 1:
         GO TC SMFCPT:
         END:
         ELSE DC:
         M = M + 1:
         GO TC SMFCPT:
         END:
PUT EDIT (WORK1,'0','SYSTEM=',SYS_IC,'MCDEL=',MODEL,'JWT=',
         JWAIT_TIME,'SMFBUF=',SMFBUF_SIZE,'MAIN CORE(K)=',
         CORE_SIZE,'SMFOPT=',OPTION,'DATE=',DATE_STAMP)
         (COL(1),F(6,3),COL(9),A,CCL(11),A(7),COL(18),A(2),
         COL(21),A(6),COL(27),A(2),CCL(30),A(4),COL(34),F(3),
         COL(38),A(7),COL(45),F(5),CCL(51),A(13),COL(64),F(4),
         COL(69),A(7),COL(76),A,COL(81),A,COL(86),F(5));
```

Following are the assignment statements, processing, and the PUT EDIT statement for SMF record type 4.

```
SMF(4): WORK3 = TIME_STAMP;
        WORK1 = WORK3/C360:
        WORK3 = RDR_START_TIME:
        WORK5 = WCRK3/C360:
        WORK3 = START_TIME;
        WORK7 = WORK3/C360:
/* CHECK FOR JOB OR OUTPUT WRITER RECORD */
        IF RFCORD_TYPE = 5 THEN GC TC JOBTYPE;
        IF RECORD_TYPE = 6 THEN GO TC WTRTYPE;
/* CONVERT STEP P/P CPU TIME FROM 100THS/SECS TO SECONDS AND 100THS  */
        I = UNSPEC(LENGTH1):
        I = I + 90:
        WORK9 = UNSPEC(SUBSTR(SPACE,I,3));
        WORK9 = WCRK9/100:
        A = JS_NUMBER:
        C = REGION_REQUEST:
        D = CORE_USED:
        E = PRIORITY:
/* CONVERT STEP COMPLETICN CCDE TO PRINTABLE FORM */
        DO I = 1 TO 4;
        L = TERM_STATUS(I) + 1:
        DEVCA(I) = SUBSTR('0123456789ABCDEF',L,1):
        END:
/* PRINT OUT STEP TERMINATICN RECCRD VALUES */
        PUT EDIT (WORK1,'4',JCB_NAME,WORK5,A,WORK7,'-------',
                 SYSIN_COUNT,'-----',WCRK9,PROGRAM_NAME,C,D,'--',E,
                 DEVC(1),'---- ---- --')
                 (COL(1),F(6,3),COL(9),A,CCL(11),A(8),COL(19),F(6,3),
                 COL(26),F(2),COL(29),F(6,3),CCL(36),A(6),COL(43),
                 F(5),COL(49),A(5),COL(55),F(7,2),COL(63),A(8),
                 COL(72),F(3),COL(76),F(3),CCL(80),A(2),COL(83),F(3),
                 COL(87),A(4),COL(92),A(12));
```

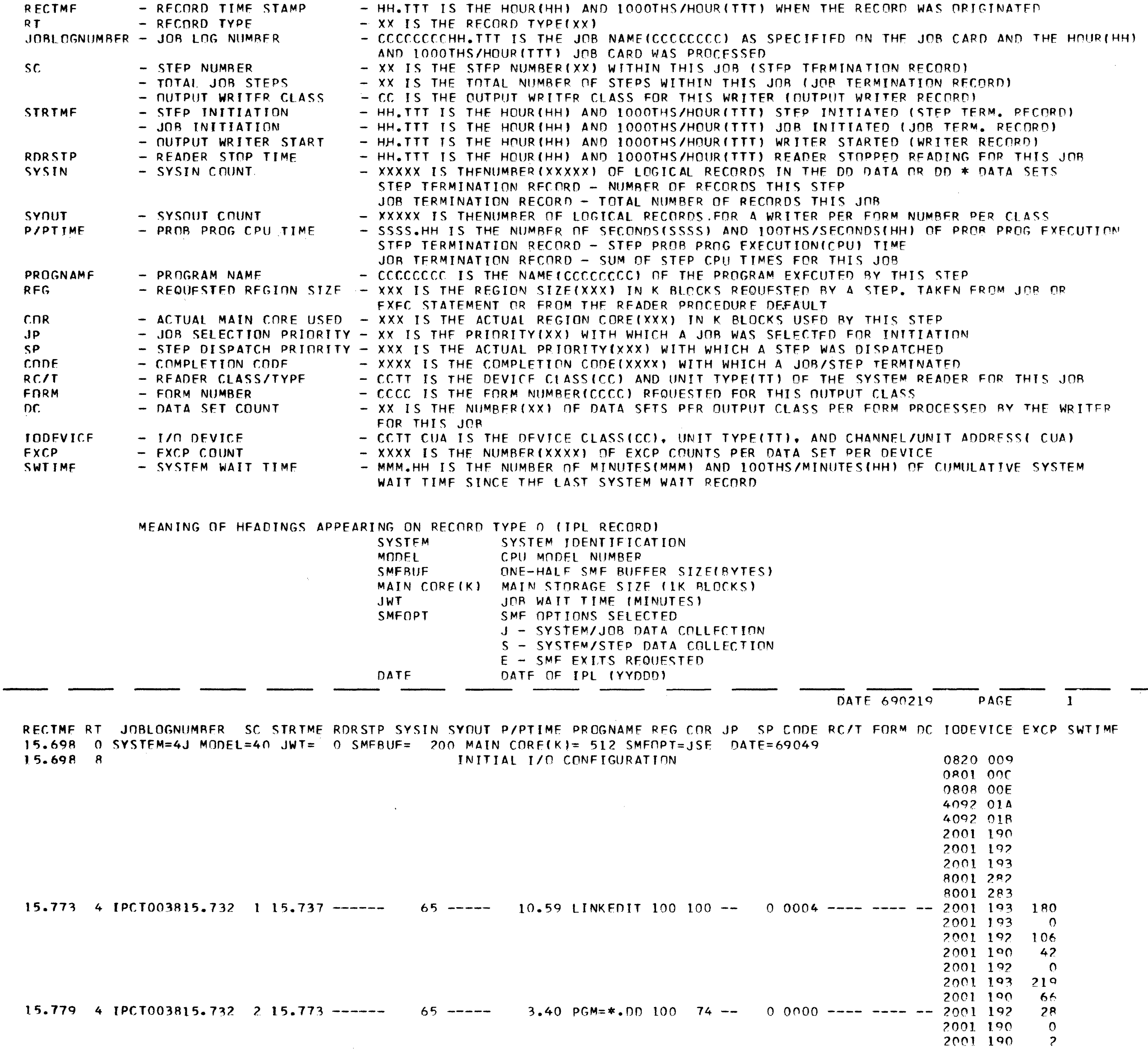
• Figure 14. Sample PL/I Report Program (Part 2 of 3)

Following are the assignment statements, processing, and the PUT EDIT statement for SMF record type 8.

```
SMF(8): HEADING = 'INITIAL I/O CONFIGURATICN'; TYPE = ' 8';
        GO TO DEVICE:
DEVICE: WORK3 = TIME_STAMP:
        WORK1 = WORK3/C360:
        PUT EDIT (WORK1,TYPE,HEADING)
                 (COL(1),F(6,3),COL(8),A,CCL(50),A);
AGAIN1: J = 1;
        K = 8;
        N = 1;
        B = C;
        M = (LENGTH - 2)*2:
AGAIN2: DO I = J TO K;
        L = UNIT(I) + 1;
        B = B + 1;
        DEVCE(B) = SUBSTR('0123456789ABCDEF',L,1);
       FND;
        DEVCE(5) = ' ';
        PUT EDIT (DEVCD(1)) (COL(105),A(8));
        J = J + 8;
        K = K + 8;
        N = N + 1;
        B = C;
        IF J<M THEN GC TC AGAIN2;
               ELSE HEADING = ' ': GO TO BRING;
```

The assignment statements assign values to fields previously defined in DECLARE statements. These values are used in the processing for each record type, and in the resulting output record. The PUT EDIT statements define the format for the output records, and cause them to be written on the SYSOUT device. (The standard system output file — SYSPRINT — is assumed.)

• Figure 14. Sample PL/I Report Program (Part 3 of 3)



```
RECTME        - RECORD TIME STAMP       - HH.TTT IS THE HOUR(HH) AND 1000THS/HOUR(TTT) WHEN THE RECORD WAS ORIGINATED
RT            - RECORD TYPE             - XX IS THE RECORD TYPE(XX)
JOBLOGNUMBER  - JOB LOG NUMBER          - CCCCCCCCHH.TTT IS THE JOB NAME(CCCCCCCC) AS SPECIFIED ON THE JOB CARD AND THE HOUR(HH)
                                          AND 1000THS/HOUR(TTT) JOB CARD WAS PROCESSED
SC            - STEP NUMBER             - XX IS THE STEP NUMBER(XX) WITHIN THIS JOB (STEP TERMINATION RECORD)
              - TOTAL JOB STEPS         - XX IS THE TOTAL NUMBER OF STEPS WITHIN THIS JOB (JOB TERMINATION RECORD)
              - OUTPUT WRITER CLASS     - CC IS THE OUTPUT WRITER CLASS FOR THIS WRITER (OUTPUT WRITER RECORD)
STRTME        - STEP INITIATION         - HH.TTT IS THE HOUR(HH) AND 1000THS/HOUR(TTT) STEP INITIATED (STEP TERM. RECORD)
              - JOB INITIATION          - HH.TTT IS THE HOUR(HH) AND 1000THS/HOUR(TTT) JOB INITIATED (JOB TERM. RECORD)
              - OUTPUT WRITER START     - HH.TTT IS THE HOUR(HH) AND 1000THS/HOUR(TTT) WRITER STARTED (WRITER RECORD)
RDRSTP        - READER STOP TIME        - HH.TTT IS THE HOUR(HH) AND 1000THS/HOUR(TTT) READER STOPPED READING FOR THIS JOB
SYSIN         - SYSIN COUNT             - XXXXX IS THENUMBER(XXXXX) OF LOGICAL RECORDS IN THE DD DATA OR DD * DATA SETS
                                          STEP TERMINATION RECORD - NUMBER OF RECORDS THIS STEP
                                          JOB TERMINATION RECORD - TOTAL NUMBER OF RECORDS THIS JOB
SYOUT         - SYSOUT COUNT            - XXXXX IS THENUMBER OF LOGICAL RECORDS.FOR A WRITER PER FORM NUMBER PER CLASS
P/PTIME       - PROB PROG CPU TIME      - SSSS.HH IS THE NUMBER OF SECONDS(SSSS) AND 100THS/SECONDS(HH) OF PROB PROG EXECUTION
                                          STEP TERMINATION RECORD - STEP PROB PROG EXECUTION(CPU) TIME
                                          JOB TERMINATION RECORD - SUM OF STEP CPU TIMES FOR THIS JOB
PROGNAME      - PROGRAM NAME            - CCCCCCCC IS THE NAME(CCCCCCCC) OF THE PROGRAM EXECUTED BY THIS STEP
REG           - REQUESTED REGION SIZE   - XXX IS THE REGION SIZE(XXX) IN K BLOCKS REQUESTED BY A STEP. TAKEN FROM JOB OR
                                          EXEC STATEMENT OR FROM THE READER PROCEDURE DEFAULT
COR           - ACTUAL MAIN CORE USED   - XXX IS THE ACTUAL REGION CORE(XXX) IN K BLOCKS USED BY THIS STEP
JP            - JOB SELECTION PRIORITY  - XX IS THE PRIORITY(XX) WITH WHICH A JOB WAS SELECTED FOR INITIATION
SP            - STEP DISPATCH PRIORITY  - XXX IS THE ACTUAL PRIORITY(XXX) WITH WHICH A STEP WAS DISPATCHED
CODE          - COMPLETION CODE         - XXXX IS THE COMPLETION CODE(XXXX) WITH WHICH A JOB/STEP TERMINATED
RC/T          - READER CLASS/TYPE       - CCTT IS THE DEVICE CLASS(CC) AND UNIT TYPE(TT) OF THE SYSTEM READER FOR THIS JOB
FORM          - FORM NUMBER             - CCCC IS THE FORM NUMBER(CCCC) REQUESTED FOR THIS OUTPUT CLASS
DC            - DATA SET COUNT          - XX IS THE NUMBER(XX) OF DATA SETS PER OUTPUT CLASS PER FORM PROCESSED BY THE WRITER
                                          FOR THIS JOB
IODEVICE      - I/O DEVICE              - CCTT CUA IS THE DEVICE CLASS(CC), UNIT TYPE(TT), AND CHANNEL/UNIT ADDRESS( CUA)
EXCP          - EXCP COUNT              - XXXX IS THE NUMBER(XXXX) OF EXCP COUNTS PER DATA SET PER DEVICE
SWTIME        - SYSTEM WAIT TIME        - MMM.HH IS THE NUMBER OF MINUTES(MMM) AND 100THS/MINUTES(HH) OF CUMULATIVE SYSTEM
                                          WAIT TIME SINCE THE LAST SYSTEM WAIT RECORD


              MEANING OF HEADINGS APPEARING ON RECORD TYPE 0 (IPL RECORD)
                                          SYSTEM        SYSTEM IDENTIFICATION
                                          MODEL         CPU MODEL NUMBER
                                          SMFBUF        ONE-HALF SMF BUFFER SIZE(BYTES)
                                          MAIN CORE(K)  MAIN STORAGE SIZE (1K BLOCKS)
                                          JWT           JOB WAIT TIME (MINUTES)
                                          SMFOPT        SMF OPTIONS SELECTED
                                                        J - SYSTEM/JOB DATA COLLECTION
                                                        S - SYSTEM/STEP DATA COLLECTION
                                                        E - SMF EXITS REQUESTED
                                          DATE          DATE OF IPL (YYDDD)
____  ____  ____  ____  ____  ____  ____  ____  ____  ____  ____  ____  ____  ____  ____  ____  ____  ____
                                                                                         DATE 690219     PAGE      1

RECTME RT  JOBLOGNUMBER  SC STRTME RDRSTP SYSIN SYOUT P/PTIME PROGNAME REG COR JP  SP CODE RC/T FORM DC IODEVICE EXCP SWTIME
15.698  0 SYSTEM=4J MODEL=40 JWT=  0 SMFBUF=  200 MAIN CORE(K)= 512 SMFOPT=JSE  DATE=69049
15.698  8                                     INITIAL I/O CONFIGURATION                             0820 009
                                                                                                    0801 00C
                                                                                                    0808 00E
                                                                                                    4092 01A
                                                                                                    4092 01B
                                                                                                    2001 190
                                                                                                    2001 192
                                                                                                    2001 193
                                                                                                    8001 282
                                                                                                    8001 283
15.773  4 IPCT003815.732  1 15.737 ------     65 -----    10.59 LINKEDIT 100 100 --    0 0004 ---- ---- -- 2001 193  180
                                                                                                    2001 193    0
                                                                                                    2001 192  106
                                                                                                    2001 190   42
                                                                                                    2001 192    0
                                                                                                    2001 193  219
                                                                                                    2001 190   66
15.779  4 IPCT003815.732  2 15.773 ------     65 -----     3.40 PGM=*.DD 100  74 --    0 0000 ---- ---- -- 2001 192   28
                                                                                                    2001 190    0
                                                                                                    2001 190    2
```

• Figure 15. Sample Output From PL/I Report Program

Associated with SMFPOST in SYS1.SAMPLIB is sample JCL for cataloged procedure PL1LFCL (compile and link edit). Figure 16 illustrates the JCL provided; if you plan to use SMFPOST, you must substitute JCL appropriate to your application.

You may obtain a punched source deck or a listing of SMFPOST by using the IEBPTPCH utility program. Figure 17 contains sample JCL for executing the utility program.

```
//A319P186 JOB PS24010101,MSGLEVEL=1,REGION=150K
//STEP1 EXEC PROC=PL1LFCLG,PARM.PL1L='SIZE=100000,L,E,A,X,O,NT,OP'
//PL1L.SYSPUNCH DD SYSOUT=B
//PL1L.SYSIN DD *
//GO.SYSPRINT DD SYSOUT=Q,DCB=(LRECL=132,BLKSIZE=3828)
//GO.DATAIN DD UNIT 2400,LABEL=(,NL),DCB=(RECFM=VBS,BLKSIZE=3625),    *
//       VOLUME=SER=(123456,AAAAAA,BBBBBB)
```

• Figure 16. Sample JCL for Procedure PL1LFCLG

Use the following procedure to obtain a source deck and listing of SMFPOST:

```
//PNCHPRNT JOB 123456, SMITH
//       EXEC  PGM=IEBPTPCH
//SYSPRINT  DD  SYSOUT=A
//SYSUT1    DD  DSNAME=SYS1.SAMPLIB,DISP=(OLD,KEEP),                  *
//              UNIT=2311,VOLUME=SER=DLIB01
//SYSUT2    DD  UNIT=2540-2
//SYSIN     DD  *
     PUNCH  TYPORG=PO,MAXNAME=1,MAXFLDS=1
     MEMBER NAME=SMFPOST
     RECORD FIELD=(80)
/*
//       EXEC  PGM=IEBPTPCH
//SYSPRINT  DD  SYSOUT=A
//SYSUT1    DD  DSNAME=SYS1.SAMPLIB,DISP=(OLD,KEEP),                  *
//              UNIT=2311,VOLUME=SER=DLIB01
//SYSUT2    DD  SYSOUT=A
//SYSIN     DD  *
     PRINT  TYPORG=PO,MAXNAME=1,MAXFLDS=1
     MEMBER NAME=SMFPOST
     RECORD FIELD=(80)
/*
```

• Figure 17. Obtaining a Source Deck and Listing of SMFPOST

Section 4: Incorporating SMF into the System

SMF is specified as a system generation option. The installation-supplied exit routines should be included in SYS1.CI505 prior to system generation; otherwise they must be link edited into the appropriate system load module. The member containing SMF options (SMFDEFLT) in SYS1.PARMLIB should be tailored to installation requirements before the first IPL of the generated system.

SYSGEN Procedure

Two system generation macro instructions are specifically related to SMF. Table 5 lists the macro instruction parameters you must specify to ensure that the SMF option is included in your system.

Table 5. Required SYSGEN Macro Instruction Parameters

Macro Instruction	Parameter required for SMF
SCHEDULR	ACCTRTN=SMF
SUPRVSOR	TIMER=JOBSTEP

The other parameters you will supply for these macro instructions depend on your installation requirements. The value of the JOBQLMT parameter in the SCHEDULR macro instruction must be increased to include two additional messages per job and per step. Additional system queue space must be specified in the CTRLPROG macro instruction using the QSPACE parameter for MVT or the SYSQUE parameter for MFT. For a complete discussion of the system generation procedure, refer to the publication IBM System/360 Operating System: System Generation, Form C28-6554.

Including Exit Routines in the System

You should incorporate your exit routines into system library SYS1.CI505 before you generate your system. Figure 18 illustrates the JCL required. If you do not require all exit routines in your system, simply omit the object deck and NAME control statement for those you do not need.

```
//EXITLNK  JOB 123456,SMITH
//          EXEC  PGM=IEWL,PARM=(DC,LET,LIST,RENT,NCAL,REFR)
//SYSPRINT DD   SYSOUT=A
//SYSLMOD  DD   DSNAME=SYS1.CI505,DISP=(OLD,KEEP)
//SYSUT1   DD   UNIT=SYSDA,DISP=(,DELETE),SPACE=(TRK,(20,5))
//SYSLIN   DD   *
       object deck for IEFUJV
   NAME  IEFUJV(R)
       object deck for IEFUSO
   NAME  IEFUSO(R)
       object deck for IEFUTL
   NAME  IEFUTL(R)
/*
//          EXEC  PGM=IEWL,PARM=(DC,LET,LIST,RENT,NCAL)
//SYSPRINT DD   SYSOUT=A
//SYSLMOD  DD   DSNAME=SYS1.CI505,DISP=(OLD,KEEP)
//SYSUT1   DD   UNIT=SYSDA,DISP=(,DELETE),SPACE=(TRK,(20,5))
//SYSLIN   DD   *
       object deck for IEFUJI
   NAME  IEFUJI(R)
       object deck for IEFUSI
   NAME  IEFUSI(R)
       object deck for IEFACTRT
   NAME  IEFACTRT(R)
/*
```

• Figure 18. Adding Exit Routines to SYS1.CI505 Prior to System Generation

If you wish to add or replace exit routines after system generation, you must link edit the routines into the appropriate load module as shown in Figure 19. Figure 20 shows the required JCL for MVT Systems. Similar JCL is required for MFT systems except that exit routines IEFUSO and IEFUTL must be link edited into the nucleus.

Object module for this exit routine:	Must be link edited into this load module in MVT:	Must be link edited into this load module in MFT:
IEFUJV	Scheduler load module IEFUJV.	Scheduler load module IEFUJV.
IEFACTRT	Scheduler load module IEFSD061.	Scheduler load module IEFSD510.
IEFUJI	Scheduler load module IEFSD061.	Scheduler load module IEFSD512.
IEFUSI	Scheduler load module IEFSD061.	Scheduler load module IEFSD512.
IEFUSO	Scheduler load module IEFSD263.	Nucleus load module IEANUC01.
IEFUTL	Scheduler load module IEFSD263.	Nucleus load module IEANUC01.

• Figure 19. Required Load Module Assignments for Exit Routines

```
//EXITLNK JOB 123456,SMITH
//          EXEC  PGM=IEWL,PARM=(DC,LET,LIST,RENT,NCAL,REFR)
//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)
     .
     .  (IEFUSO and IEFUTL object decks)
     .
  ENTRY    IEFSD263
  INCLUDE  SYSLMOD(IEFSD263)
  NAME     IEFSD263(R)
//          EXEC  PGM=IEWL,PARM=(DC,LET,LIST,NCAL)
//SYSPRINT DD   SYSOUT=A
//SYSLMOD  DD   DSNAME=SYS1.LINKLIB,DISP=(OLD,KEEP)
//SYSUT1   DD   UNIT=SYSDA,DISP=(,DELETE),SPACE=(TRK,(20,5))
//SYSLIN   DD   *
     .
     .  (IEFUJI,IEFUSI,IEFACTRT object decks)
     .
  ENTRY    IEFSD061
  INCLUDE  SYSLMOD(IEFSD061)
  ALIAS    IEFSD065,IEFSD104,IEFW42SD,IEFV4221
  NAME     IEFSD061(R)
/*
```

• Figure 20. Adding Exit Routines to SYS1.LINKLIB After System Generation in MVT

When adding exit routines after MVT system generation as shown in Figure 20, the region size in the reader/interpreter must be increased to allow for exit routine IEFUJV; the region size specified in the initiator procedure must be increased to allow for exit routines IEFUJI, IEFUSI, and IEFACTRT; the region size allocated to the link pack area must be increased to allow for exit routines IEFUSO and IEFUTL. Refer to the publication IBM System/360 Operating System: System Programmer's Guide, Form C28-6550, for information on specifying these region sizes.

When adding exit routines after system generation in either MVT or MFT, refer to your system generation listing for exact load module names and aliases.

Specifying SMFDEFLT Parameters

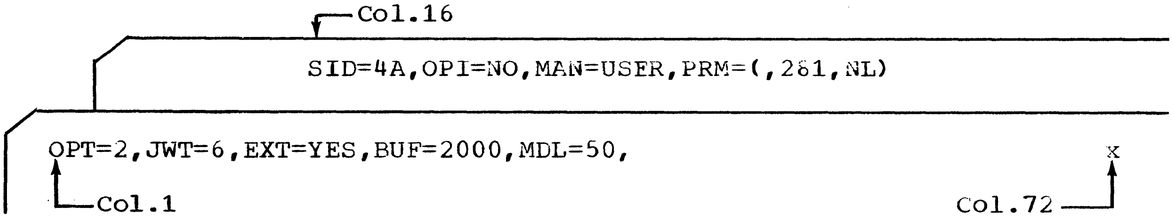
SMFDEFLT contains the options and parameters which control SMF operations. You must add SMFDEFLT as a member in SYS1.PARMLIB before the first IPL of your newly generated system.

SMFDEFLT Contents and Format

SMFDEFLT consists of a series of parameters contained in 80-character card-image records. (See Figure 21.) Each parameter has the format:

$$\text{keyword} = \begin{Bmatrix} \text{value} \\ (\text{value}_1,\text{value}_2,\ldots\text{value}_n) \end{Bmatrix}$$

Consecutive parameters are separated by commas. No imbedded blanks are permitted. Parameters may be included in any order.



```
               SID=4A,OPI=NO,MAN=USER,PRM=(,281,NL)
OPT=2,JWT=6,EXT=YES,BUF=2000,MDL=50,                                   x
```

- Figure 21. Sample SMFDEFLT Data Set

You must conform to the following restrictions in creating SMFDEFIT:

- Do not place parameters in columns 72-80; these columns are ignored.
- Indicate continuation of records by placing a nonblank character in column 72. (All records in the member except the last will include a continuation character.)
- Do not continue parameters from one record to the next. Each set of keyword and value must be complete within a single record.
- Start your data in column 1 on the first record, and in column 16 on all continuation records.

Following is a definition of the purpose and format of the SMFDEFLT parameters.

<u>System/Job/Step Information</u>: This optional parameter specifies the type of system, job, and step information to be collected by SMF.

$$\text{OPT} = \begin{Bmatrix} 1 \\ \underline{2} \end{Bmatrix}$$

1 - Only system and job information.

2 - System, job, and step information. If the parameter is omitted, 2 is the default value.

Data Set/Direct Access Volume Information: This optional parameter specifies the type of data set and direct access volume information to be collected by SMF.

$$\text{DSV} = \left\{\begin{matrix}\underline{0}\\1\\2\\3\end{matrix}\right\}$$

0 - Neither data set nor direct access volume information. If the parameter is omitted, 0 is the default value.

1 - Only direct access volume information.

2 - Only data set information.

3 - Both data set and direct access volume information.

Temporary Data Set Information: This optional parameter specifies whether record types 17 and 18 will be written for temporary data sets[1].

$$\text{REC} = \left\{\begin{matrix}\underline{0}\\2\end{matrix}\right\}$$

0 - Record type 17 will be written for non-temporary data sets only; it will not be written for temporary data sets. If the parameter is omitted, 0 is the default value.

2 - Record type 17 will be written for temporary as well as non-temporary data sets.

Exit Routines: This optional parameter specifies whether the SMF exits are to be taken.

$$\text{EXT} = \left\{\begin{matrix}\text{NO}\\\underline{\text{YES}}\end{matrix}\right\}$$

NO - exits will not be taken.

YES - exits will be taken. If the parameter is omitted, YES is the default value.

If EXT = YES is specified, the exits actually taken will depend on the data collection parameter. If OPT=2 is specified, all exits will be taken; if OPT=1 is specified, step initiation and step termination exits will not be taken.

[1] The system determines that a data set is temporary if either of the following criteria is met:

- The DD statement for the data set contains DISP=(NEW,DELETE).
- The DD statement for a new data set does not include the DSNAME parameter, or contains a parameter of the form DSNAME=&name.

Job Wait Time: This required parameter specifies the number of minutes a job is allowed to remain continuously in the wait state. When the specified limit has been reached, the time limit exit (IEFUTL) is entered (if an exit routine has been provided).

JWT = n

n - is a one-to-three digit decimal number specifying (in minutes) the job wait time limit. 0 must not be specified as the limit.

SMF Buffer Size: This optional parameter specifies the size of the SMF buffer. Buffer size requirements are previously discussed in Section 2 of this publication under the topic "Data Management Considerations." If SMF or user records are not to be recorded, this parameter is not required.

BUF = n

n - is a three-to-five digit decimal number defining the size (in bytes) of the SMF buffer. Minimum buffer size is 400 bytes; maximum size is 65,534 bytes. If the value is not a multiple of 4, it will be rounded to the next lower multiple of 4.

System Identification: This required parameter identifies the system on which SMF is active.

SID = xx

xx - is two alphameric characters identifying the system.

CPU Model Number: This required parameter identifies the model number of the system's CPU.

MDL = nn

nn - is a two-digit decimal number defining the CPU model number.

Operator Intervention: This optional parameter specifies whether the operator will be permitted to modify SMFDEFLT parameters from the console during IPL. This parameter is ignored if entered from the console.

$$\text{OPI} = \left\{\begin{matrix}\text{YES}\\ \underline{\text{NO}}\end{matrix}\right\}$$

YES - The operator is allowed to modify parameters.

NO - The operator is not allowed to modify parameters. If the parameter is omitted, NO is the default value.

SMF Records: This optional parameter specifies the type of records to be written to the SMF data set.

$$\text{MAN} = \left\{\begin{matrix}\text{NONE}\\ \text{USER}\\ \underline{\text{ALL}}\end{matrix}\right\}$$

NONE - No records will be written to the SMF data set.

USER - Only user records (from installation exit routines) will be written to the SMF data set.

ALL - Both SMF and user records will be written to the SMF data set. If the parameter is omitted, ALL is the default value.

<u>Primary SMF Data Set</u>: This optional parameter specifies the primary SMF data set (SYS1.MANX). If no SMF or user records are to be written, this parameter is not required.

PRM = (vol.ser.no.,dev.addr.,$\left\{\begin{matrix}\text{NL}\\ \text{NSL}\\ \underline{\text{SL}}\end{matrix}\right\}$)

vol.ser.no. - is one to six alphameric characters specifying the serial number of the volume on which the data set resides.

dev.addr. - is three alphameric characters specifying the address of the device on which the volume is mounted.

NL - indicates that the data set is on a tape with no label.

NSL - indicates that the data set is on a tape with a nonstandard label.

SL - indicates that the data set is on a tape with a standard label.

The absence of one of the values must be indicated by a comma only if another value is included following the omitted value. If the data set is defined on tape, the device address must be specified; if no tape label value is given, SL is assumed. If a 7-track tape is used, the byte converter feature must be available; recording will be at 800 BPI. If a dual density tape is used, recording will be at 1600 BPI. If the data set is defined on a direct access device, either the volume serial number or the device address, or both may be specified.

<u>Alternate SMF Data Set</u>: This optional parameter specifies the alternate SMF data set (SYS1.MANY). This data set must be defined on the same type of direct access device as the primary data set (SYS1.MANX). If no SMF or user records are to be written, or if the alternate data set is not used, this parameter is not required. If the primary data set is defined on tape, this parameter is ignored.

ALT = (vol.ser.no.,dev.addr.)

vol.ser.no. - is one to six alphameric characters specifying the serial number of the volume on which the data set resides.

dev.addr. - is three alphameric characters specifying the address of the device on which the volume is mounted.

Entering SMFDEFLT Into SYS1.PARMLIB

When you have determined the parameters for SMFDEFLT and have it punched into a card deck, you should add it as a member of system library SYS1.PARMLIB by executing the utility program IEBUPDTE. Figure 22 illustrates the JCL required to execute the utility program. If your parameters change, you may replace the entire SMFDEFLT member with a new version by again executing IEBUPDTE, substituting a ./ REPL control statement for the ./ ADD control statement. The REPL statement will have the same operand as the ADD statement. For further information on the IEBUPDTE program, refer to the publication IBM System/360 Operating System: Utilities.

```
//ENTER   JOB    123456,SMITH
//        EXEC   PGM=IEBUPDTE,PARM=MOD
//SYSPRINT  DD SYSOUT=A
//SYSUT1   DD DSAME=SYS1.PARMLIB,DISP=(OLD,KEEP)
//SYSUT2   DD DSAME=SYS1.PARMLIB,DISP=(OLD,KEEP)
//SYSIN    DD DATA
./    ADD  LIST=ALL,NAME=SMFDEFLT,LEVEL=01,SOURCE=0
        .
        .
        .
     SMFDEFLT data set
        .
        .
        .
/*
```

• Figure 22. Entering SMFDEFLT Into SYS1.PARMLIB

If operator intervention is allowed (OPI=YES), you may change SMFDEFLT parameter values from the operator's console during IPL. You may supply missing required parameters in the same manner. Appropriate messages will be issued to notify the operator when intervention is required or permitted. Messages are described in Section 7.

Note: If you do not add SMFDEFLT as a member of SYS1.PARMLIB, the parameters must be entered from the operator's console during IPL. This procedure, however, is quite time-consuming and must be repeated during each IPL of the system.

Section 5: Operational Considerations

Once SMF has been generated as part of a system, the system operator must be aware of several operational procedures such as requirements for initial program loading (IPL) and for dumping the SMF data set. The operator must also be aware of any special procedures required by installation-written report and analysis programs.

IPL Procedures

SMF initialization is the final step of the system IPL procedure. The SMF initialization program checks for the existence and validity of the SMFDEFLT member and for the availability of the devices on which data sets SYS1.MANX and SYS1.MANY are defined.

SMFDEFLT Verification: If SMFDEFLT has not been entered into SYS1.PARMLIB, the initialization program writes message IEE352A to the console, allowing you to enter SMFDEFLT parameters from the console. If the verification program encounters an I/O error while reading SMFDEFLT, it writes message IEE353A to the console, allowing you to either enter SMFDEFLT parameters from the console, or to repeat the entire IPL procedure.

If the initialization program finds SMFDEFLT and reads it without error, each parameter is checked for accuracy. If any parameters are incorrectly specified, messages IEE355I and IEE356A are issued, allowing you to enter the correct parameter from the console. If all parameters are correctly specified and the parameter OPI=YES (operator intervention allowed) is included in SMFDEFLT, messages IEE354I and IEE357A are issued, allowing you to examine all the parameters and make modifications as required.

SMF Data Set Verification: When the initialization program has completed verification of SMFDEFLT, it checks for the existence of data sets SYS1.MANX and SYS1.MANY and for the availability of the devices on which they are defined. Table 6 lists the conditions that the initialization routine checks, and the messages issued. If any error condition is found, message IEE351I is also issued, indicating that no recording of SMF records will be allowed until the condition is corrected and the IPL procedure is repeated. Section 7 contains an explanation of all SMF messages.

• Table 6. Verification of SMF Data Sets

If this condition is found	This message is written to the console
Data set not defined in SMFDEFLT	IEE350I
Specified device is not available	IEE363I
Data set not allocated on specified direct access device	IEE358I

If no errors are found, the initialization routine determines which data set (SYS1.MANX or SYS1.MANY) should receive SMF records. SMF data will be maintained in chronological sequence if both SYS1.MANX and SYS1.MANY are defined on the same type of device and have the same space attributes, and if the data sets are dumped when called for by the control program.

When data set verification has been completed without error, the initialization routine writes the IPL and I/O configuration records (SMF record types 0 and 8) to the specified data set, and gives control to the system so that processing of input streams may begin.

Data Management Procedures

Procedures for handling the SMF data set depend on the type of device on which the data set is resident. If the data set is resident on tape, you must mount a new tape prior to each IPL of the system. If the data set is resident on direct access, you must allocate space on each specified device before IPL. If possible, a device and channel other than those specified for SYSRES should be used. Figure 23 illustrates sample DD statements for allocating space to the SMF data set.

Note: Specification of a secondary space allocation is ignored.

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

• Figure 23. Allocating Space for SMF Data Sets on Direct Access Devices

SMF Dump Program

When the SMF data set is resident on a direct access device, you must use the SMF dump program to transfer filled data sets to another data set (on tape) for analysis. When either the primary (SYS1.MANX) or alternate (SYS1.MANY) data sets are filled, message IEE362A is written to the console, requesting a dump. Figure 24 illustrates the JCL required to execute the dump.

```
//DUMPX        JOB     201,,MSGLEVEL=1,PRTY=12

//STEP1        EXEC    PGM=IFASMFDP

                                      {X}
//DUMPIN       DD      DSNAME=SYS1.MAN{Y},DISP=OLD

//DUMPOUT      DD      UNIT=TAPE,DISP=(NEW,KEEP)

//SYSPRINT     DD      SYSOUT=A
```

• Figure 24. Executing the SMF Dump Program

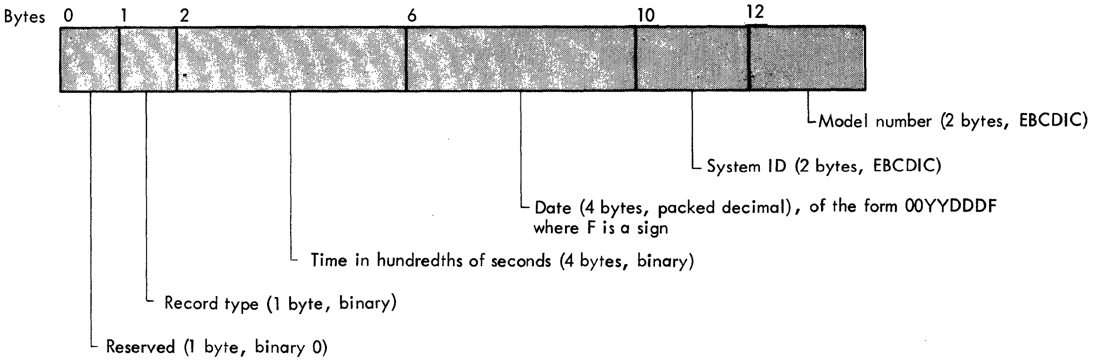
You may enter jobs specifying execution of the dump program into the system and hold them on the job queue until a dump is required. You may then release the appropriate job to dump the specified data set. Another method for executing the dump is to simply start a reader to an input stream containing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

Section 6: SMF Record Formats

This section describes the formats of all records written to the SMF data set by SMF routines.

Standard SMF Record Header

Each record written to the SMF data set (by SMF routines or by installation-written routines) should contain the following standard SMF record header (length 14 bytes):

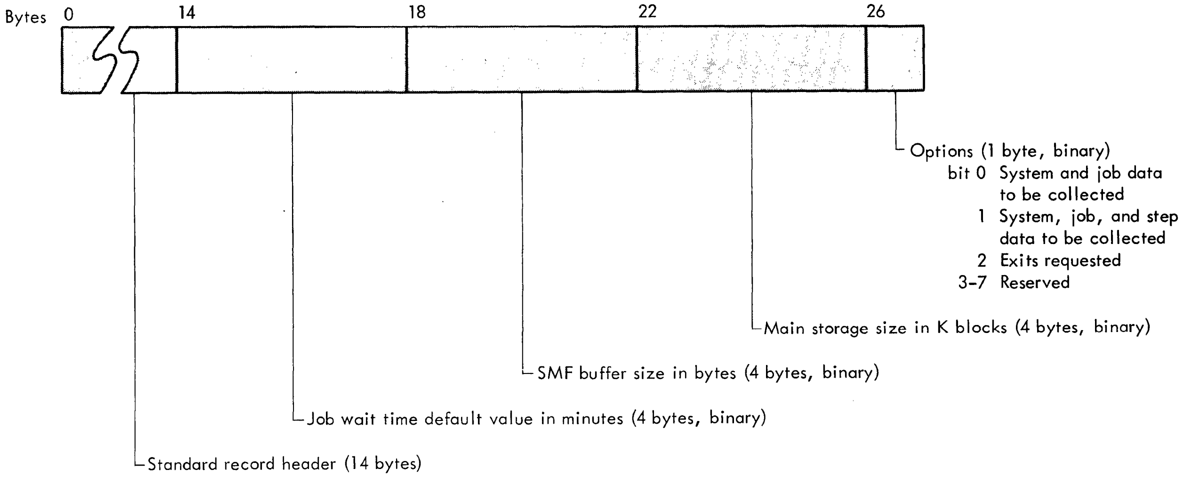


Bytes 0 1 2 6 10 12
Model number (2 bytes, EBCDIC)
System ID (2 bytes, EBCDIC)
Date (4 bytes, packed decimal), of the form 00YYDDDF where F is a sign
Time in hundredths of seconds (4 bytes, binary)
Record type (1 byte, binary)
Reserved (1 byte, binary 0)

Note: When the SMFWTM macro instruction is used to write a record to the SMF data set, a record descriptor word (4 bytes) must be added to the beginning of the record header; the address passed to the macro instruction must point to the beginning of the record descriptor word. For a discussion of the record descriptor word, refer to the publication IBM System/360 Operating System: Supervisor and Data Management Services.

Record Type 0 (IPL Record)

This record is written after every IPL of the system. Length is 27 bytes.



Options (1 byte, binary)
bit 0 System and job data to be collected
1 System, job, and step data to be collected
2 Exits requested
3-7 Reserved

Main storage size in K blocks (4 bytes, binary)

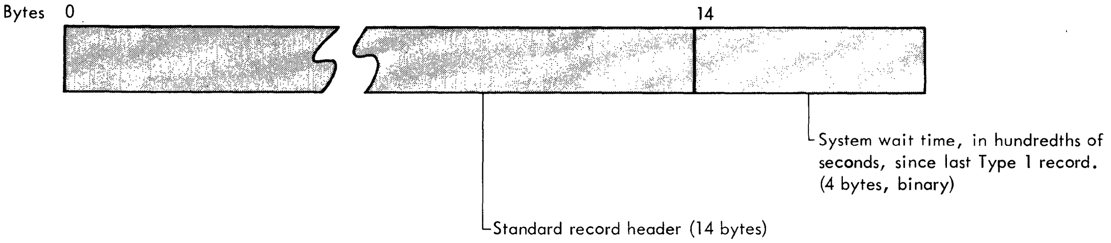
SMF buffer size in bytes (4 bytes, binary)

Job wait time default value in minutes (4 bytes, binary)

Standard record header (14 bytes)

Record Type 1 (Wait Time Record)

This record is written every ten minutes while the system is active. (Written at the first step termination after the ten minute interval.) Length is 18 bytes.

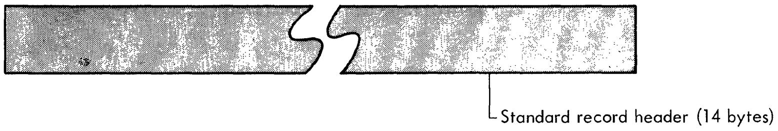


System wait time, in hundredths of seconds, since last Type 1 record. (4 bytes, binary)

Standard record header (14 bytes)

Record Type 2 (Dump Header Record)

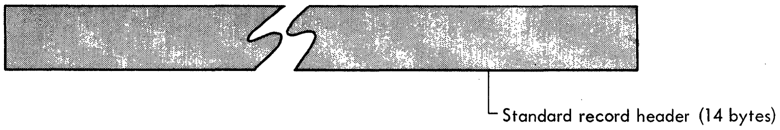
This record is written by the SMF dump program at the beginning of a dump data set. Length is 14 bytes.



Standard record header (14 bytes)

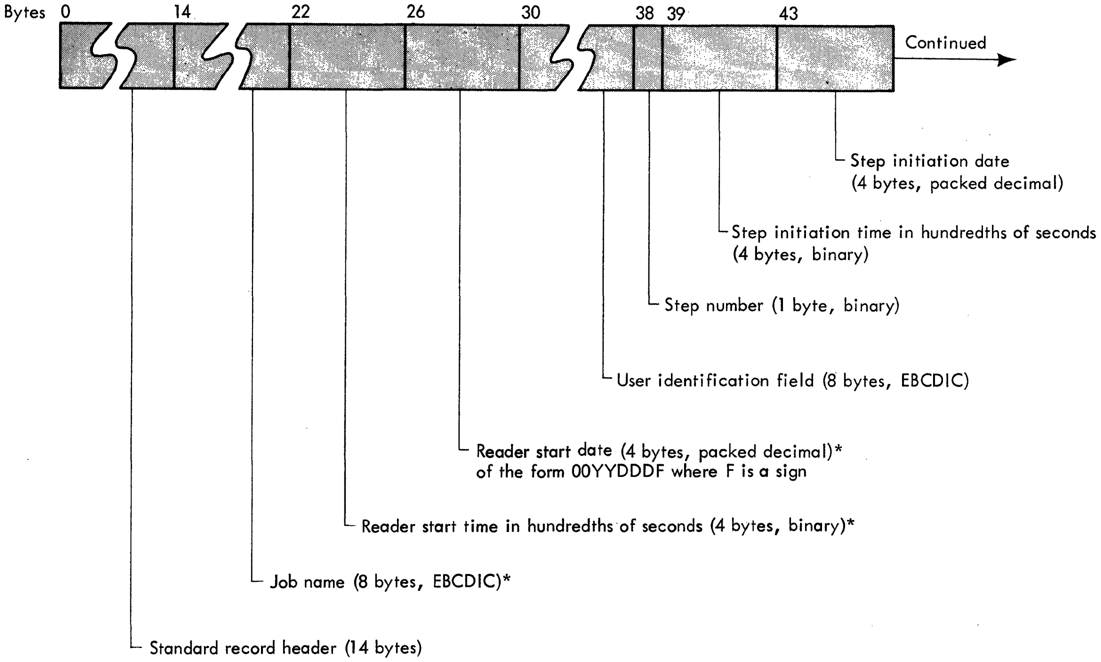
Record Type 3 (Dump Trailer Record)

This record is written by the SMF dump program at the end of a dump data set. Length is 14 bytes.



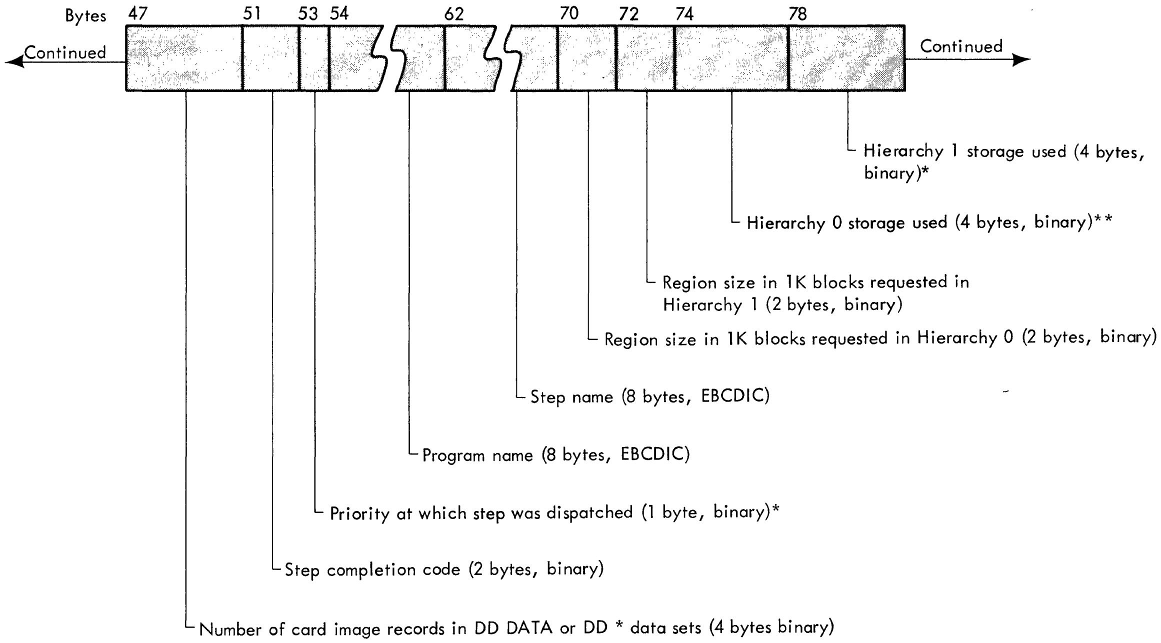
Record Type 4 (Step Termination Record) (Part 1 of 3)

This record is written at the normal or abnormal termination of a job step. Length is variable.


Bytes 0
14
22
26
30
38 39
43
Continued
Step initiation date (4 bytes, packed decimal)
Step initiation time in hundredths of seconds (4 bytes, binary)
Step number (1 byte, binary)
User identification field (8 bytes, EBCDIC)
Reader start date (4 bytes, packed decimal)* of the form 00YYDDDF where F is a sign
Reader start time in hundredths of seconds (4 bytes, binary)*
Job name (8 bytes, EBCDIC)*
Standard record header (14 bytes)

* The job name, reader start time, and reader start date fields constitute the job log number.

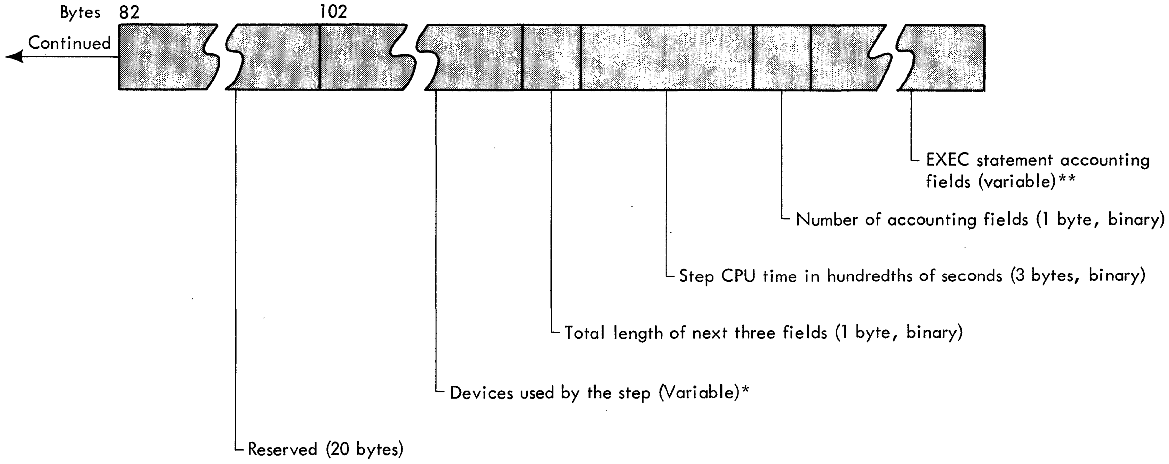
RECORD TYPE 4 (STEP TERMINATION RECORD) (PART 2 OF 3)


Bytes
47
51
53
54
62
70
72
74
78
Continued
Continued
Hierarchy 1 storage used (4 bytes, binary)*
Hierarchy 0 storage used (4 bytes, binary)**
Region size in 1K blocks requested in Hierarchy 1 (2 bytes, binary)
Region size in 1K blocks requested in Hierarchy 0 (2 bytes, binary)
Step name (8 bytes, EBCDIC)
Program name (8 bytes, EBCDIC)
Priority at which step was dispatched (1 byte, binary)*
Step completion code (2 bytes, binary)
Number of card image records in DD DATA or DD * data sets (4 bytes binary)

* Actual priority is computed from user-assigned priority (0-13) using the following algorithm: Actual priority=251-(15-requested priority)*16.

** Bytes 0 and 1 indicate the actual storage used (in 1K blocks) within the specified region. Bytes 2 and 3 indicate additional storage (in 1K blocks) allocated to the region. (Bytes 2 and 3 are only used in systems supporting the rollout/rollin feature.)

RECORD TYPE 4 (STEP TERMINATION RECORD) (PART 3 OF 3)

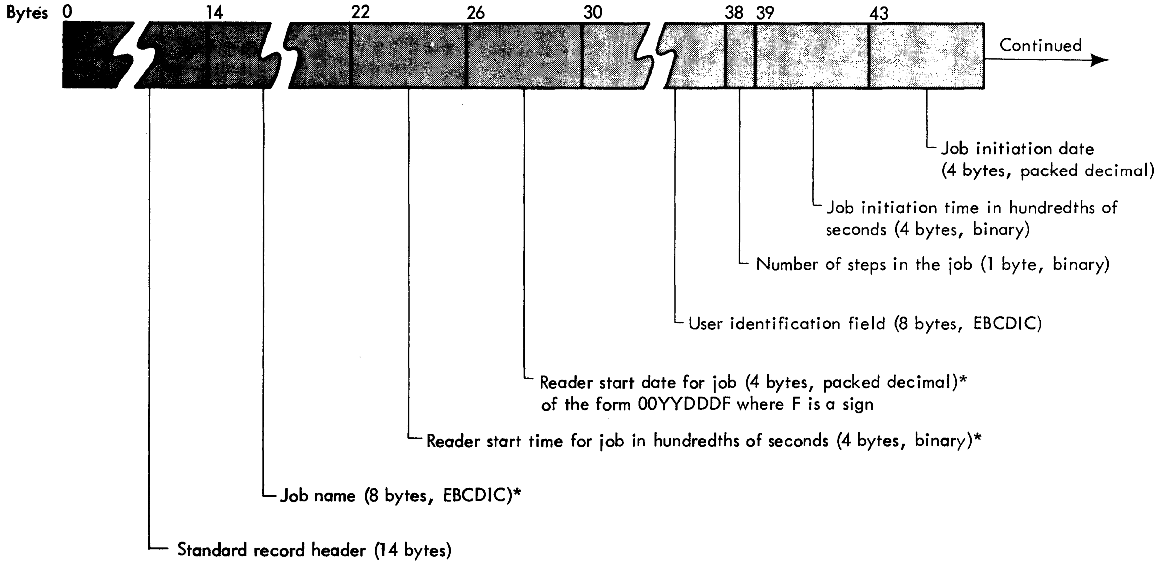

Bytes 82
102
Continued
EXEC statement accounting fields (variable)**
Number of accounting fields (1 byte, binary)
Step CPU time in hundredths of seconds (3 bytes, binary)
Total length of next three fields (1 byte, binary)
Devices used by the step (Variable)*
Reserved (20 bytes)

* Bytes 0 and 1 contain the length of the field. For each device assigned to each data set there is an eight byte entry having the following format:
Byte 0 -- Device class.
Byte 1 -- Unit type.
Bytes 2,3 -- Channel and unit address.
Bytes 4-7 -- Count of EXCPs issued for the device and data set.
For a DD DUMMY data set the entry is set to 0.

** Each entry for an accounting field contains the length of the field (1 byte, binary), followed by the field (EBCDIC). An omitted field is represented by a length indicator of 0.

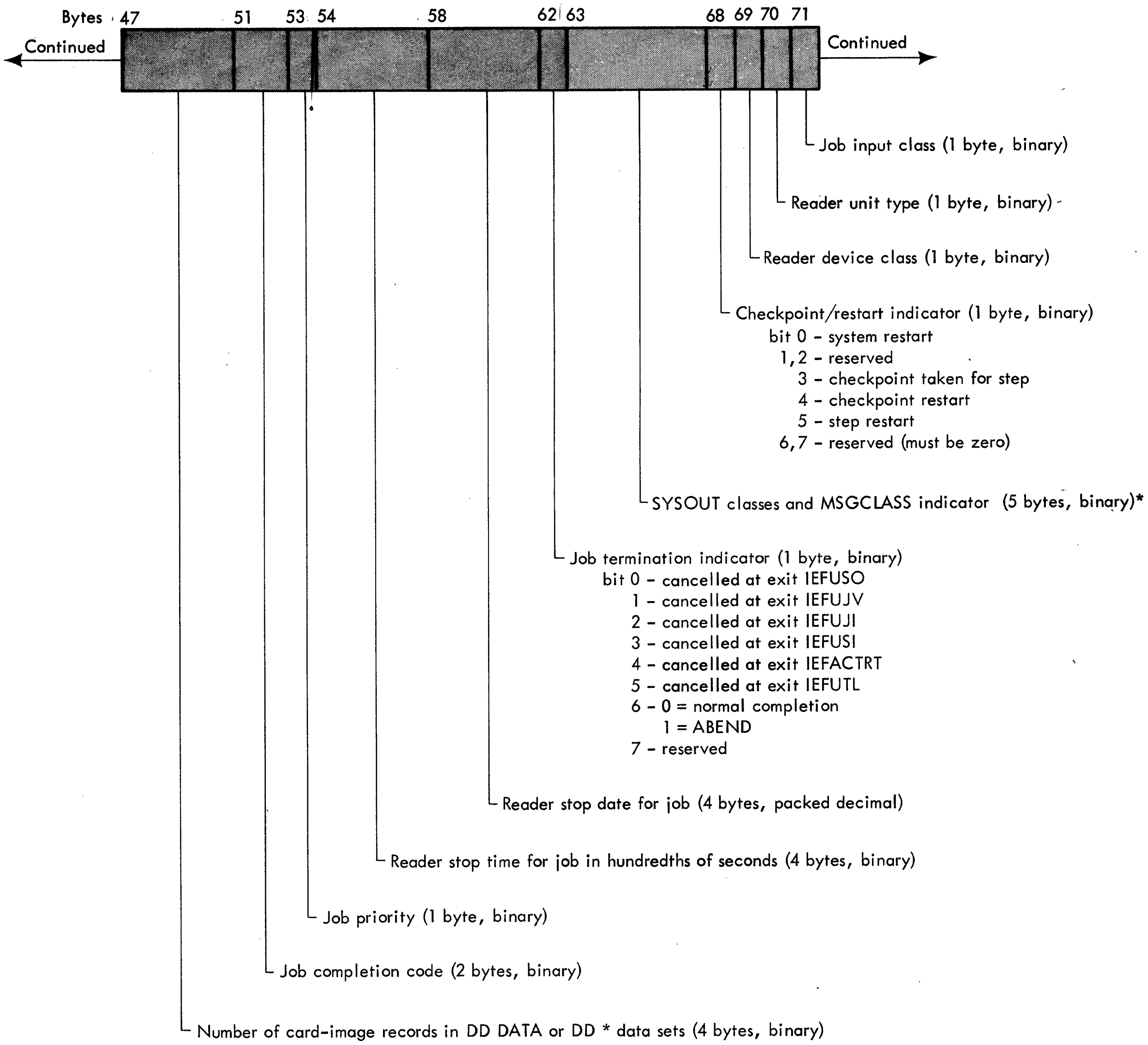
Record Type 5 (Job Termination Record) (Part 1 of 3)

This record is written at normal or abnormal termination of a job. Length is variable.


Bytes 0
14
22
26
30
38 39
43
Continued
Job initiation date (4 bytes, packed decimal)
Job initiation time in hundredths of seconds (4 bytes, binary)
Number of steps in the job (1 byte, binary)
User identification field (8 bytes, EBCDIC)
Reader start date for job (4 bytes, packed decimal)* of the form 00YYDDDF where F is a sign
Reader start time for job in hundredths of seconds (4 bytes, binary)*
Job name (8 bytes, EBCDIC)*
Standard record header (14 bytes)

* The job name, reader start time, and reader start date fields constitute the job log number.

RECORD TYPE 5 (JOB TERMINATION RECORD) (PART 2 OF 3)



Job input class (1 byte, binary)

Reader unit type (1 byte, binary)

Reader device class (1 byte, binary)

Checkpoint/restart indicator (1 byte, binary)
bit 0 - system restart
1,2 - reserved
3 - checkpoint taken for step
4 - checkpoint restart
5 - step restart
6,7 - reserved (must be zero)

SYSOUT classes and MSGCLASS indicator (5 bytes, binary)*

Job termination indicator (1 byte, binary)
bit 0 - cancelled at exit IEFUSO
1 - cancelled at exit IEFUJV
2 - cancelled at exit IEFUJI
3 - cancelled at exit IEFUSI
4 - cancelled at exit IEFACTRT
5 - cancelled at exit IEFUTL
6 - 0 = normal completion
1 = ABEND
7 - reserved

Reader stop date for job (4 bytes, packed decimal)

Reader stop time for job in hundredths of seconds (4 bytes, binary)

Job priority (1 byte, binary)

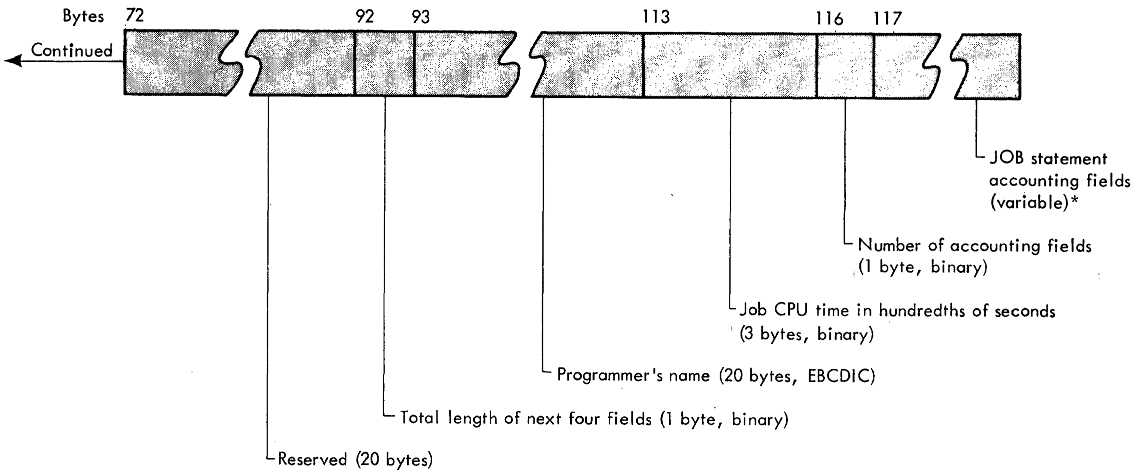
Job completion code (2 bytes, binary)

Number of card-image records in DD DATA or DD * data sets (4 bytes, binary)

* Each bit of the indicator represents the following classes:

Byte	0	1	2	3	4
Bit	01234567	01234567	01234567	01234567	0123
Class	ABCDEFGH	IJKLMNOP	QRSTUVWX	YZ012345	6789

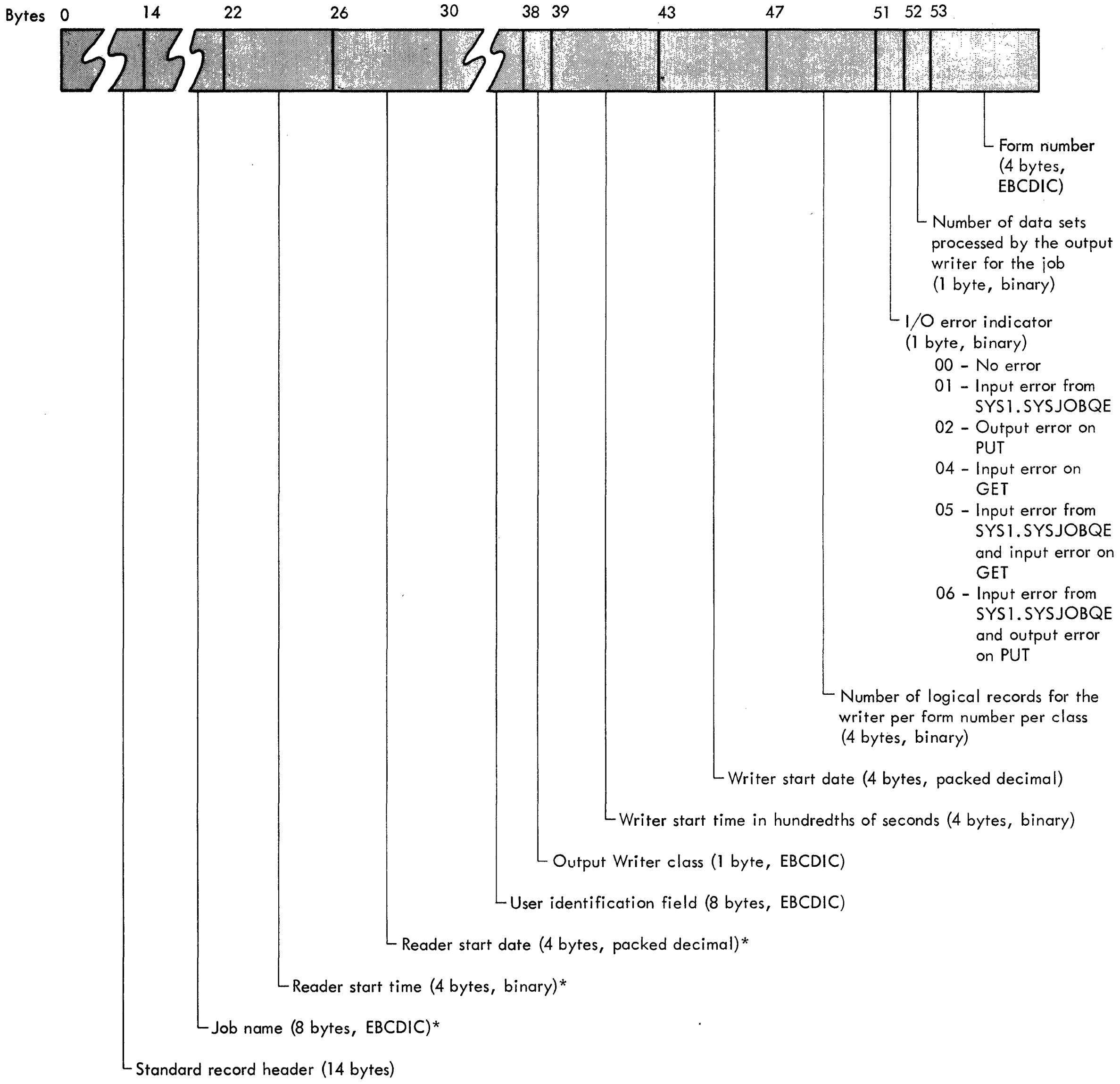
RECORD TYPE 5 (JOB TERMINATION RECORD) (PART 3 OF 3)


Bytes
72
92
93
113
116
117
Continued
JOB statement accounting fields (variable)*
Number of accounting fields (1 byte, binary)
Job CPU time in hundredths of seconds (3 bytes, binary)
Programmer's name (20 bytes, EBCDIC)
Total length of next four fields (1 byte, binary)
Reserved (20 bytes)

* Each entry for an accounting field contains the length of the field (1 byte, binary), followed by the field (EBCDIC). An omitted field is represented by a length indicator of 0.

Record Type 6 (Output Writer Record)

This record is written when processing of a SYSOUT class is completed for a job, and when the form changes within a class. Length is 57 bytes.



Form number (4 bytes, EBCDIC)

Number of data sets processed by the output writer for the job (1 byte, binary)

I/O error indicator (1 byte, binary)

- 00 - No error
- 01 - Input error from SYS1.SYSJOBQE
- 02 - Output error on PUT
- 04 - Input error on GET
- 05 - Input error from SYS1.SYSJOBQE and input error on GET
- 06 - Input error from SYS1.SYSJOBQE and output error on PUT

Number of logical records for the writer per form number per class (4 bytes, binary)

Writer start date (4 bytes, packed decimal)

Writer start time in hundredths of seconds (4 bytes, binary)

Output Writer class (1 byte, EBCDIC)

User identification field (8 bytes, EBCDIC)

Reader start date (4 bytes, packed decimal)*

Reader start time (4 bytes, binary)*

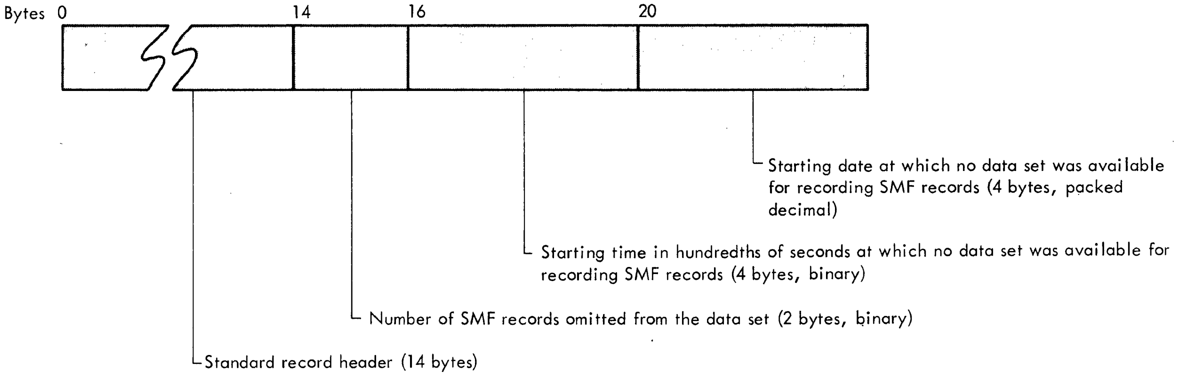
Job name (8 bytes, EBCDIC)*

Standard record header (14 bytes)

* The job name, reader start time, and reader start date fields constitute the job log number.

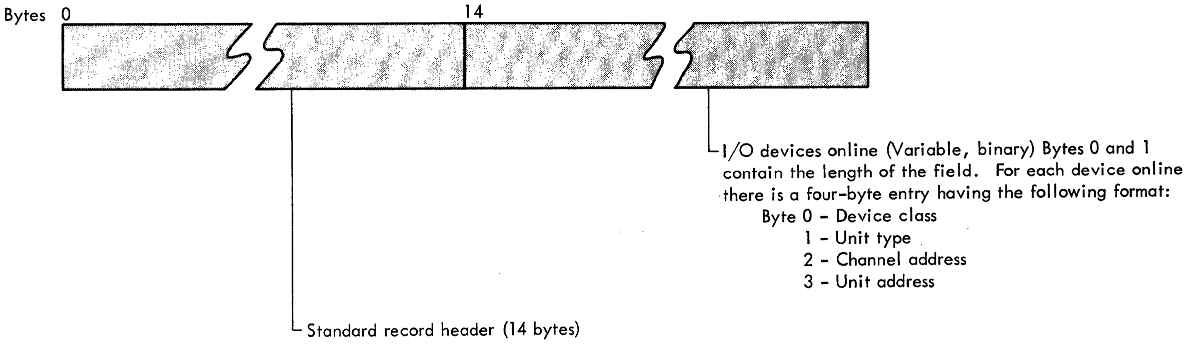
Record Type 7 (Data Lost)

This record is written when an SMF data set becomes available following the unavailability of an SMF data set. Length is 24 bytes.


Bytes 0
14
16
20
Starting date at which no data set was available for recording SMF records (4 bytes, packed decimal)
Starting time in hundredths of seconds at which no data set was available for recording SMF records (4 bytes, binary)
Number of SMF records omitted from the data set (2 bytes, binary)
Standard record header (14 bytes)

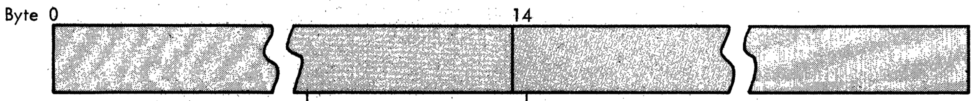
Record Type 8 (I/O Configuration)

This record is written after completion of IPL, following the SET DATE command. Length is variable.


Bytes 0
14
I/O devices online (Variable, binary) Bytes 0 and 1 contain the length of the field. For each device online there is a four-byte entry having the following format:
Byte 0 - Device class
1 - Unit type
2 - Channel address
3 - Unit address
Standard record header (14 bytes)

Record Type 9 (VARY ONLINE Record)

This record is written when a VARY ONLINE command is processed. Length is variable.



CPU, channel, device, or storage added (variable, binary). Bytes 0 and 1 contain the length of the field. For each element added there is a four byte entry with the following format:

Vary CPU

Bytes 0 and 1 - 2065 (hexadecimal)
2 - 00 (hexadecimal)
3 - 01 or 02 (hexadecimal) indicating CPU A or CPU B

Vary Channel

Bytes 0 and 1 - 2860 or 2870 (hexadecimal) indicating which channel (multiplexor or selector)
2 - 00 to 06 (hexadecimal) indicating which channel number
3 - 01 or 02 (hexadecimal) indicating CPU A or CPU B

Vary Device

Byte 0 - device class
1 - unit type
2 - channel address
3 - unit address

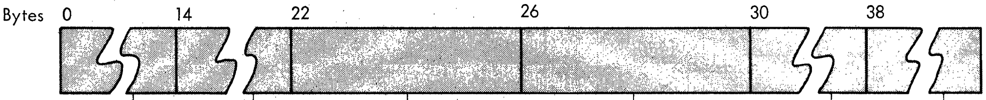
Vary Storage (Separate entries for Box 1 through Box 8)

Bytes 0 and 1 - 2365 (hexadecimal)
2 - address index (a hexadecimal number between 0 and 80 which when converted to decimal and multiplied by 2048 will give the starting address of the storage relative to the beginning of the box)
3 - size index (a hexadecimal number between 0 and 80 which when converted to decimal and multiplied by 2048 will give the number of bytes of storage varied online)

Standard record header (14 bytes)

Record Type 10 (Allocation Recovery Record)

This record is written after successful allocation. Length is variable.



Devices being made available (Variable, binary). Bytes 0 and 1 contain the length of the field. For each device there is a four-byte entry having the following for format:

Byte 0 - Device class
1 - Unit type
2 - Channel address
3 - Unit address

User identification field (8 bytes, EBCDIC)

Reader start date (4 bytes, packed decimal)*

Reader start time in hundredths of seconds (4 bytes, binary)*

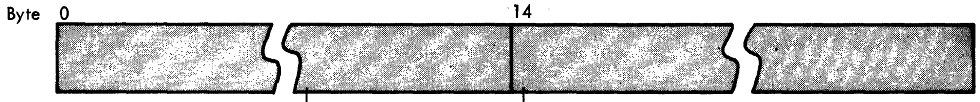
Job name (8 bytes, EBCDIC)*

Standard record header (14 bytes)

* The job name, reader start time, and reader start date fields constitute the job log number. Note: If allocation recovery is for a system task, the job name field will contain blanks and the reader start time and reader start data fields will contain binary zeroes.

Record Type 11 (VARY OFFLINE Record)

This record is written when a VARY OFFLINE command is processed. Length is variable.



CPU, channel, device, or storage removed (variable, binary). Bytes 0 and 1 contain the length of the field. For each element removed there is a four type byte entry with the following format:

Vary CPU

Bytes 0 and 1 - 2065 (hexadecimal)
2 - 00 (hexadecimal)
3 - 01 or 02 (hexadecimal) indicating CPU A or CPU B

Vary Channel

Bytes 0 and 1 - 2860 or 2870 (hexadecimal) indicating which channel (multiplexor or selector)
2 - 00 to 06 (hexadecimal) indicating which channel number
3 - 01 or 02 (hexadecimal) indicating CPU A or CPU B

Vary Device

Byte 0 - device class
1 - unit type
2 - channel address
3 - unit address

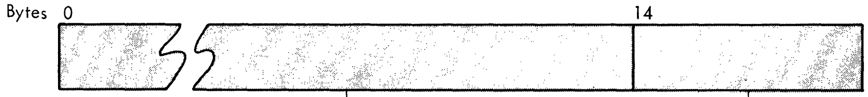
Vary Storage (Separate entries for Box 1 through Box 8)

Bytes 0 and 1 - 2365 (hexadecimal)
2 - address index (a hexadecimal number between 0 and 80 which when converted to decimal and multiplied by 2048 will give the starting address of the storage relative to the beginning of the box)
3 - size index (a hexadecimal number between 0 and 80 which when converted to decimal and multiplied by 2048 will give the number of bytes of storage varied offline)

Standard record header (14 bytes)

Record Type 12 (End-of-Day Record)

This record is written when the HALT command is processed. Length is 18 bytes.

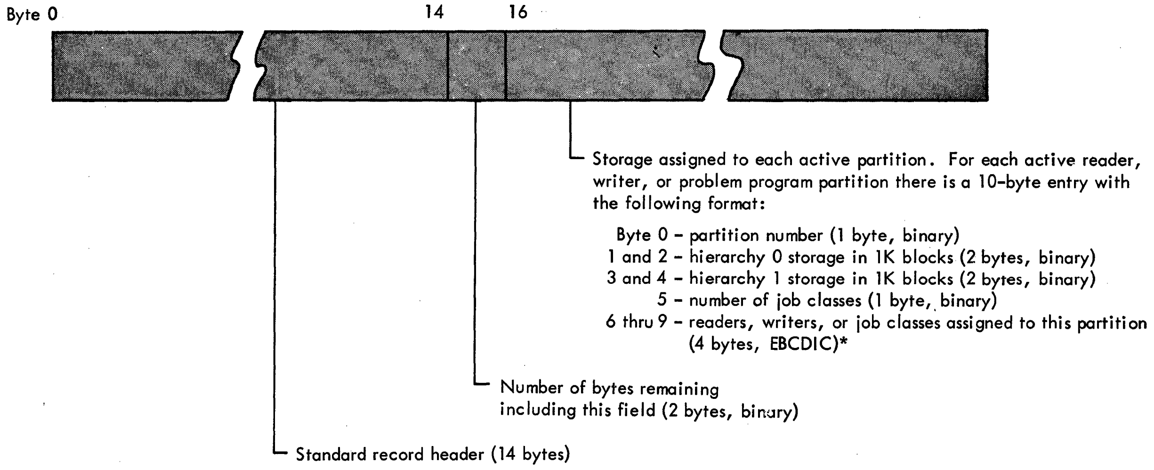


System wait time, in hundredths of seconds, since last Type 1 record (4 bytes, binary)

Standard record header (14 bytes)

Record Type 13 (Dynamic Storage Configuration)

(MFT only.) This record is written at IPL and after each DEFINE command. It shows the amount of storage assigned to each active reader, writer and problem program partition. Length is variable.



Storage assigned to each active partition. For each active reader, writer, or problem program partition there is a 10-byte entry with the following format:

Byte 0 - partition number (1 byte, binary)
1 and 2 - hierarchy 0 storage in 1K blocks (2 bytes, binary)
3 and 4 - hierarchy 1 storage in 1K blocks (2 bytes, binary)
5 - number of job classes (1 byte, binary)
6 thru 9 - readers, writers, or job classes assigned to this partition (4 bytes, EBCDIC)*

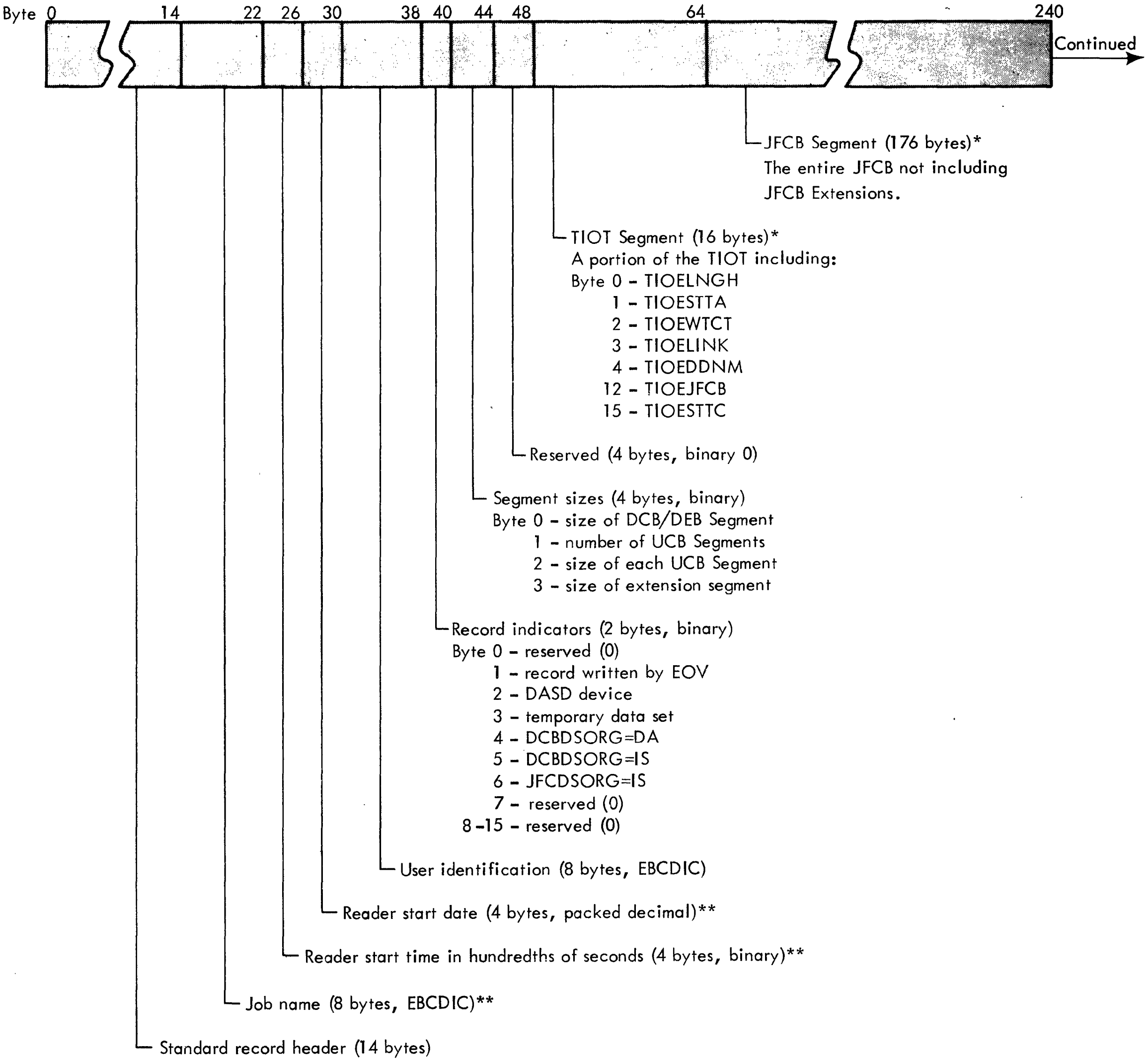
Number of bytes remaining including this field (2 bytes, binary)

Standard record header (14 bytes)

*If a reader or writer is assigned to the partition, byte 5 (number of job classes) will contain 1 and bytes 6-9 will contain ƀRDR or ƀWTR. If one job class is assigned to the partition, byte 5 will contain 1 and bytes 6-9 will contain the job class letter (A-O) right justified and padded to the left with blanks. For example: ƀƀƀA or ƀƀƀB. If more than one job class is assigned to the partition, byte 5 will contain the number of job classes (up to a maximum of 3) and bytes 6-9 will contain the job class letters in their specified order, right justified and padded to the left with blanks. For example: ƀƀAB or ƀABC.

Record Type 14 (INPUT or RDBACK Data Set Activity Record) (Part 1 of 2)

This record is written whenever a user data set opened for INPUT or RDBACK is closed or processed by EOV. Length varies from 288 to 6412 bytes, depending upon the number of volumes for the data set.



JFCB Segment (176 bytes)*
The entire JFCB not including
JFCB Extensions.

TIOT Segment (16 bytes)*
A portion of the TIOT including:
Byte 0 - TIOELNGH
1 - TIOESTTA
2 - TIOEWTCT
3 - TIOELINK
4 - TIOEDDNM
12 - TIOEJFCB
15 - TIOESTTC

Reserved (4 bytes, binary 0)

Segment sizes (4 bytes, binary)
Byte 0 - size of DCB/DEB Segment
1 - number of UCB Segments
2 - size of each UCB Segment
3 - size of extension segment

Record indicators (2 bytes, binary)
Byte 0 - reserved (0)
1 - record written by EOV
2 - DASD device
3 - temporary data set
4 - DCBDSORG=DA
5 - DCBDSORG=IS
6 - JFCDSORG=IS
7 - reserved (0)
8 -15 - reserved (0)

User identification (8 bytes, EBCDIC)

Reader start date (4 bytes, packed decimal)**

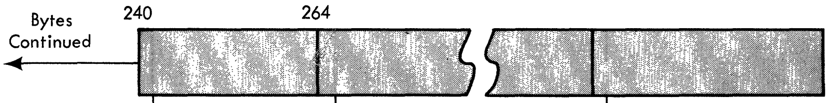
Reader start time in hundredths of seconds (4 bytes, binary)**

Job name (8 bytes, EBCDIC)**

Standard record header (14 bytes)

* For further information about the contents of the JFCB and TIOT, refer to the IBM publication IBM System/360 Operating System: System Control Blocks, Form C28-6628.

** The job name, reader start time, and reader start date fields constitute the job log number.



DCB/DEB Segment (24 bytes)*
Byte 0 - DCBDSORG
2 - DCBRECFM
3 - DCBMACRF
5 - DCBOFLGS
6 - DCBOPTCD
7 - Reserved (0)
8 - DEBOFLGS
9 - DEBOPATB
10 - DEBVOLSQ

tape extension
12 - DCBBLKCT
16 - Data Set Serial Number
22 - Reserved (0)

dasd extension
12 - Relative track (TTR) of the last record read or written
16 - Number of tracks released by the DADSM routine
20 - Number of extents released by the DADSM routine
21 - Reserved (0)

UCB Segment (24 bytes for each DCB in the data set).*
Byte 0 - UCBCHA
1 - UCBUA
2 - SRTEVOLI/DCELVOLI
8 - UCBTYP
12 - SRTESTAB/DCELSTAB
13 - Number of extents
14 - Reserved (0)
16 - EXCP Count

tape extension
20 - SRTEFSCT
22 - SRTEFSEQ

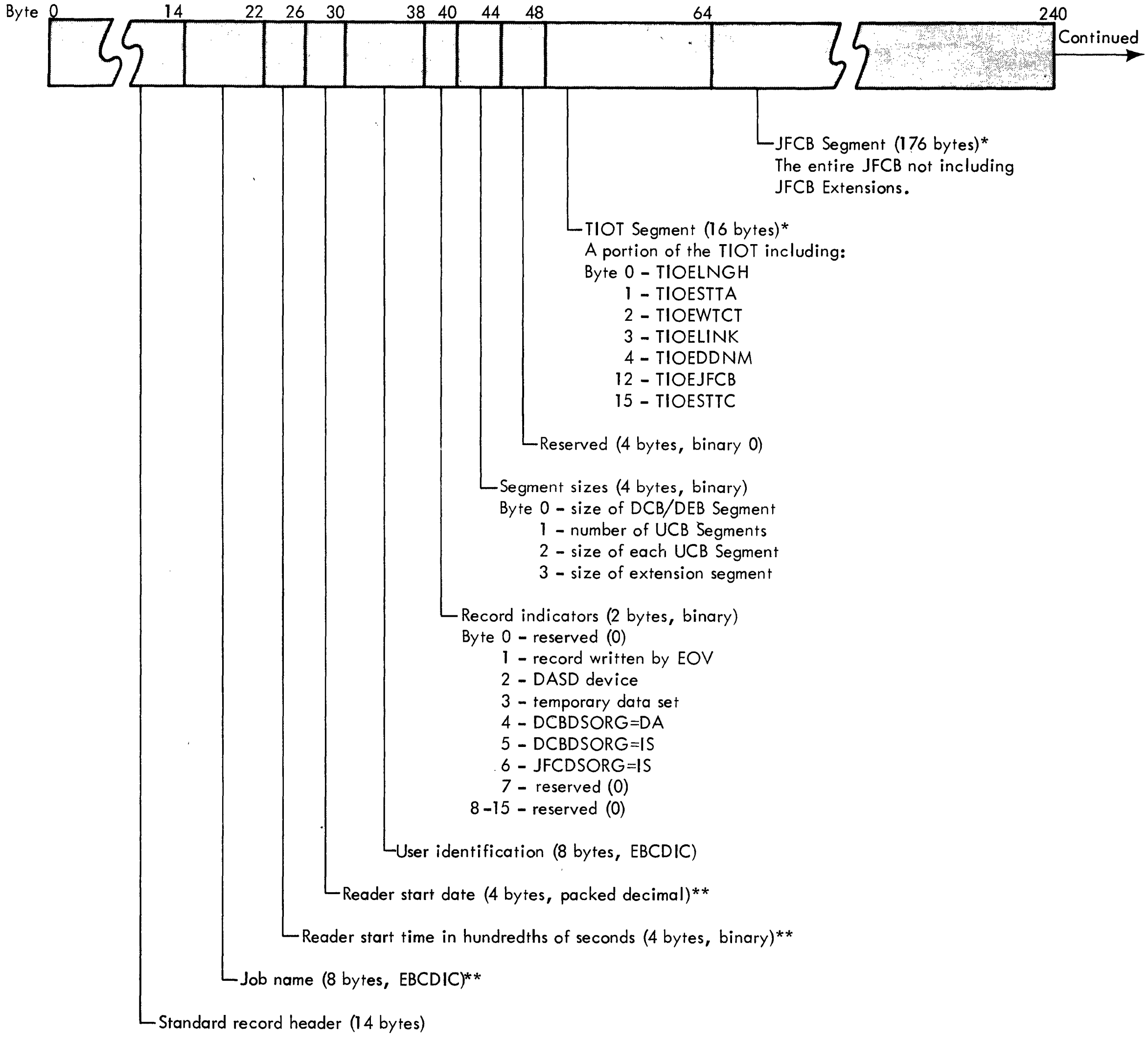
dasd extension
20 - Number of tracks allocated

ISAM Extension (28 bytes)*
For DCBDSORG=IS.
Byte 0 - Reserved (0)
2 - DCBMAC
3 - DCBNLEV
4 - DCBRORG3
8 - DCBNREC
12 - DCBRORG2
14 - DCBNOREC
16 - DCBRORG1
18 - Reserved (0)
19 - DEBNIEE
20 - DEBNPEE
21 - DEBNOEE
22 - Number of cylinders in Independent Index Area
24 - Number of cylinders in Prime Area
26 - Number of cylinders in Independent OVFL Area

*For further information about the contents of the DCB, DEB, and UCB, refer to the IBM publication IBM System/360 Operating System: System Control Blocks, Form C28-6628.

Record Type 15 (OUTPUT, UPDAT, INOUT or OUTIN Data Set Activity Record) (Part 1 of 2)

This record is written whenever a user data set opened for OUTPUT, UPDAT, INOUT, or OUTIN is closed or processed by EOV. Length varies from 288 to 6412 bytes, depending upon the number of volumes for the data set.



JFCB Segment (176 bytes)*
The entire JFCB not including
JFCB Extensions.

TIOT Segment (16 bytes)*
A portion of the TIOT including:
Byte 0 - TIOELNGH
1 - TIOESTTA
2 - TIOEWTCT
3 - TIOELINK
4 - TIOEDDNM
12 - TIOEJFCB
15 - TIOESTTC

Reserved (4 bytes, binary 0)

Segment sizes (4 bytes, binary)
Byte 0 - size of DCB/DEB Segment
1 - number of UCB Segments
2 - size of each UCB Segment
3 - size of extension segment

Record indicators (2 bytes, binary)
Byte 0 - reserved (0)
1 - record written by EOV
2 - DASD device
3 - temporary data set
4 - DCBDSORG=DA
5 - DCBDSORG=IS
6 - JFCDSORG=IS
7 - reserved (0)
8-15 - reserved (0)

User identification (8 bytes, EBCDIC)

Reader start date (4 bytes, packed decimal)**

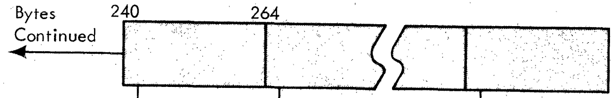
Reader start time in hundredths of seconds (4 bytes, binary)**

Job name (8 bytes, EBCDIC)**

Standard record header (14 bytes)

* For further information about the contents of the JFCB and TIOT, refer to the IBM publication IBM System/360 Operating System: System Control Blocks, Form C28-6628.

** The job name, reader start time, and reader start date fields constitute the job log number.



ISAM Extension (28 bytes)*
For DCBDSORG=IS.
Byte 0 – Reserved (0)
2 – DCBMAC
3 – DCBNLEV
4 – DCBRORG3
8 – DCBNREC
12 – DCBRORG2
14 – DCBNOREC
16 – DCBRORG1
18 – Reserved (0)
19 – DEBNIEE
20 – DEBNPEE
21 – DEBNOEE
22 – Number of cylinders in Independent Index Area
24 – Number of cylinders in Prime Area
26 – Number of cylinders in Independent OVFL Area

UCB Segment (24 bytes for each DCB in the data set).*
Byte 0 – UCBCHA
1 – UCBUA
2 – SRTEVOLI/DCELVOLI
8 – UCBTYP
12 – SRTESTAB/DCELSTAB
13 – Number of extents
14 – Reserved (0)
16 – EXCP Count

tape extension
20 – SRTEFSCT
22 – SRTEFSEQ

dasd extension
20 – Number of tracks allocated

DCB/DEB Segment (24 bytes)*
Byte 0 – DCBDSORG
2 – DCBRECFM
3 – DCBMACRF
5 – DCBOFLGS
6 – DCBOPTCD
7 – Reserved (0)
8 – DEBOFLGS
9 – DEBOPATB
10 – DEBVOLSQ

tape extension
12 – DCBBLKCT
16 – Data Set Serial Number
22 – Reserved (0)

dasd extension
12 – Relative track (TTR) of the last record read or written
16 – Number of tracks released by the DADSM routine
20 – Number of extents released by the DADSM routine
21 – Reserved (0)

* For further information about the contents of the DCB, DEB, and UCB, refer to the IBM publication IBM System/360 Operating System: System Control Blocks, Form C28-6628.

Main Storage Requirements

SMF requires additional main storage space for the resident nucleus, the system queue area, and (if exit routines are included) the region or partition size for system tasks.

Resident Nucleus: SMF adds approximately 1500 bytes to the size of the nucleus. In MFT, a separate 4000 byte partition contains the SMF resident writer routines.

System Queue Space: Space in the system queue area is required for the following control blocks:

- Timing Control Table (TCT). One TCT is created for each active step. Size for each step is determined by the following formula:

  TCT=92+12*(maximum number of DDs per step)+8*(number of devices allocated because of each DD statement)

- SMF Control Table. A permanent table occupying 120 bytes.

- SMF Buffer. The length is specified as an option in SMFDEFLT. Minimum buffer length is 400 bytes. For a discussion of the factors to consider in determining optimum buffer length, see the topic "Data Management Considerations" in Section 2.

- The area used for communication between exit routines. The way in which this area is obtained is discussed in Section 3 under the topic "Exit Routine Facilities and Restrictions".

Exit Routines: If installation-written exit routines are to be included, the region sizes specified for the reader/interpreter procedure, the initiator procedure, and the link pack area must be increased to accommodate the exit routines. The procedures for including exit routines in the system are described in Section 4.

Auxiliary Storage Requirements

The incorporation of SMF into a system requires additional auxiliary storage for system libraries and the SMF data sets.

SMF Data Sets: The SMF data set may be defined on a direct access storage device, and periodically dumped to a dump data set (e.g., on magnetic tape), or it may be defined on a tape drive, in which case no dump is necessary. If the data set is defined on a direct access device, space must be allocated for the primary SMF data set (SYS1.MANX), and for an alternate SMF data set (SYS1.MANY). The amount of space required for SMF data sets on direct access devices is a function of the types of jobs being processed and the amount of information required in the SMF records. A guide for determining appropriate extents is included in Section 2, and the method for specifying these extents is in Section 5. If the SMF data set is defined on magnetic tape, it may be written on up to 20 volumes before the IPL procedure must be repeated; no alternate is required.

System Libraries: SMF requires direct access device space for expansion of the following system libraries:

- SYS1.LINKLIB -- five tracks.
- SYS1.SVCLIB -- three tracks.
- SYS1.PARMLIB -- one track.
- SYS1.MACLIB -- one track.
- SYS1.NUCLEUS -- one track.

Adding SMF to the Operating System

To add SMF to your operating system you must:

- Include SMF in the system generation statements.

- Define your use of SMF in SMFDEFLT or from the operator's console.

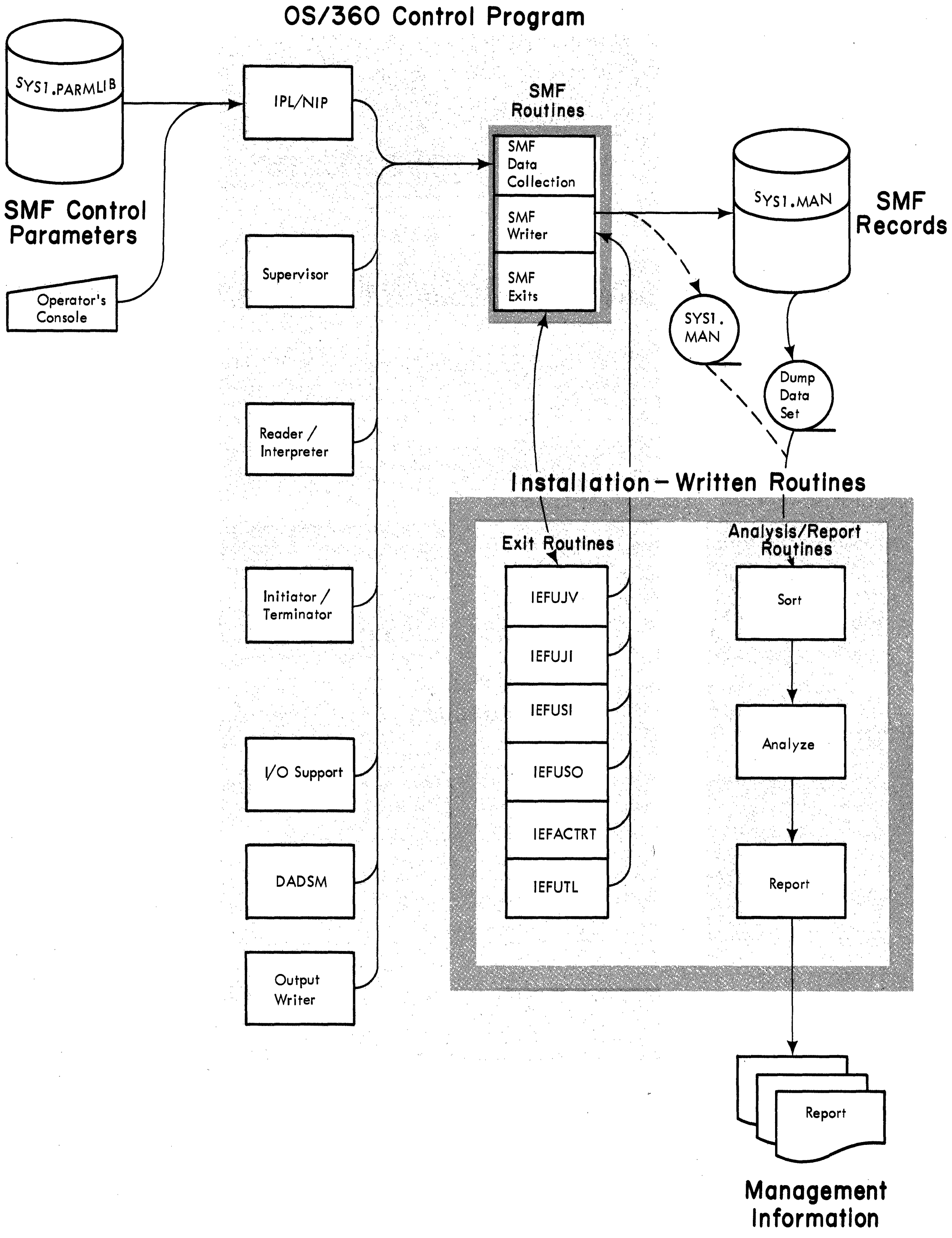
- Allocate storage for the SMF data set (if required).

- Add your exit routines (if any) to the control program.

Procedures for adding SMF to the system are described in Section 4.

SMF Operation

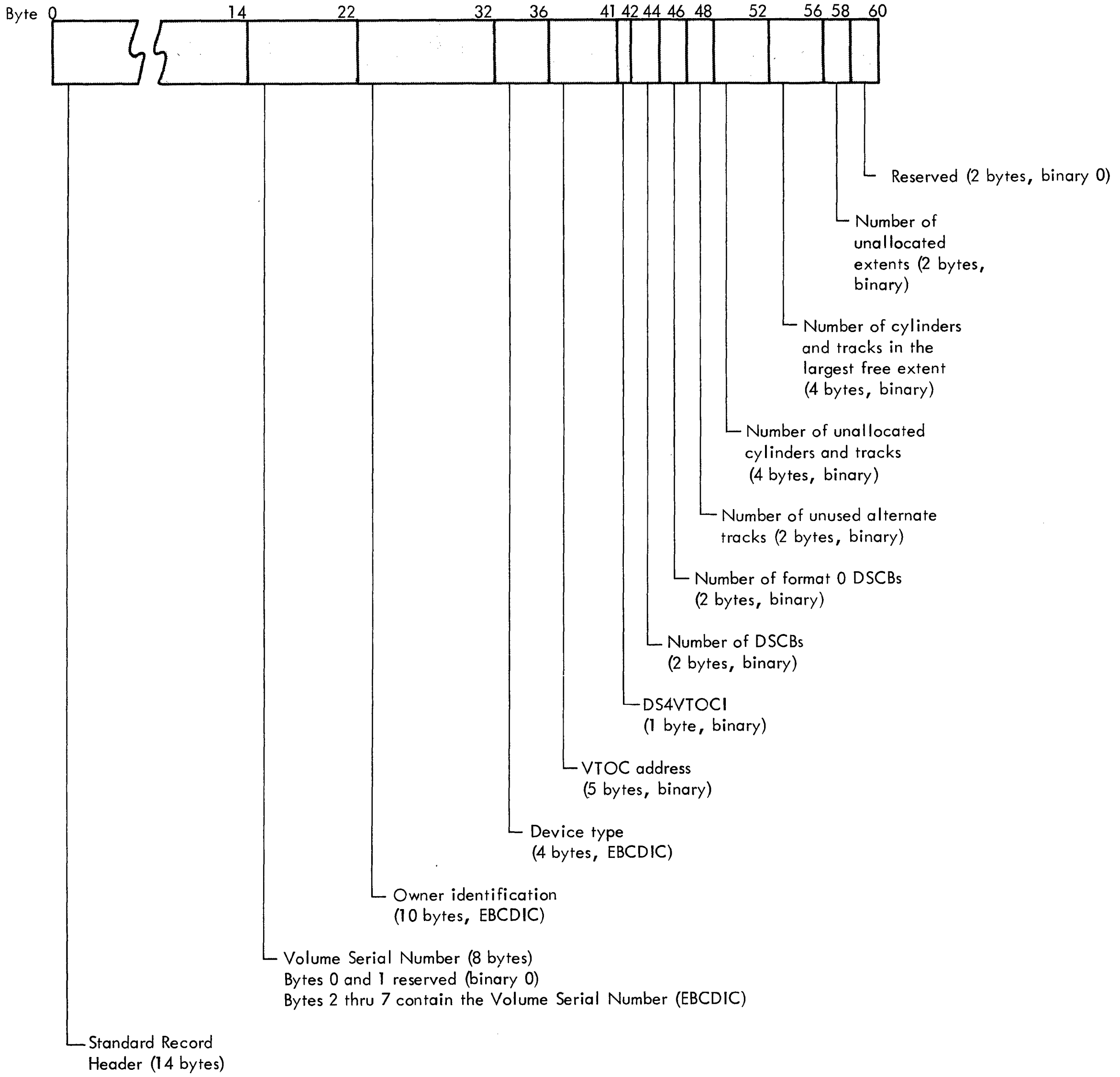
Figure 1 shows the System/360 Operating System incorporating SMF. The following paragraphs assume that installation-written exit routines are supplied for all SMF exits, that all SMF-formatted records are written to the SMF data set, and that installation-written analysis and report routines are supplied. In any real application, of course, the installation supplies only those exit routines which are needed, and specifies whether all records, no records, or a specific combination of records are to be written to the SMF data set.



• Figure 1. SMF in the Operating System

Record Type 19 (Direct Access Volume Record)

This record is written at IPL and HALT EOD and whenever a user volume is demounted. Length is 60 bytes.



Reserved (2 bytes, binary 0)

Number of unallocated extents (2 bytes, binary)

Number of cylinders and tracks in the largest free extent (4 bytes, binary)

Number of unallocated cylinders and tracks (4 bytes, binary)

Number of unused alternate tracks (2 bytes, binary)

Number of format 0 DSCBs (2 bytes, binary)

Number of DSCBs (2 bytes, binary)

DS4VTOCI (1 byte, binary)

VTOC address (5 bytes, binary)

Device type (4 bytes, EBCDIC)

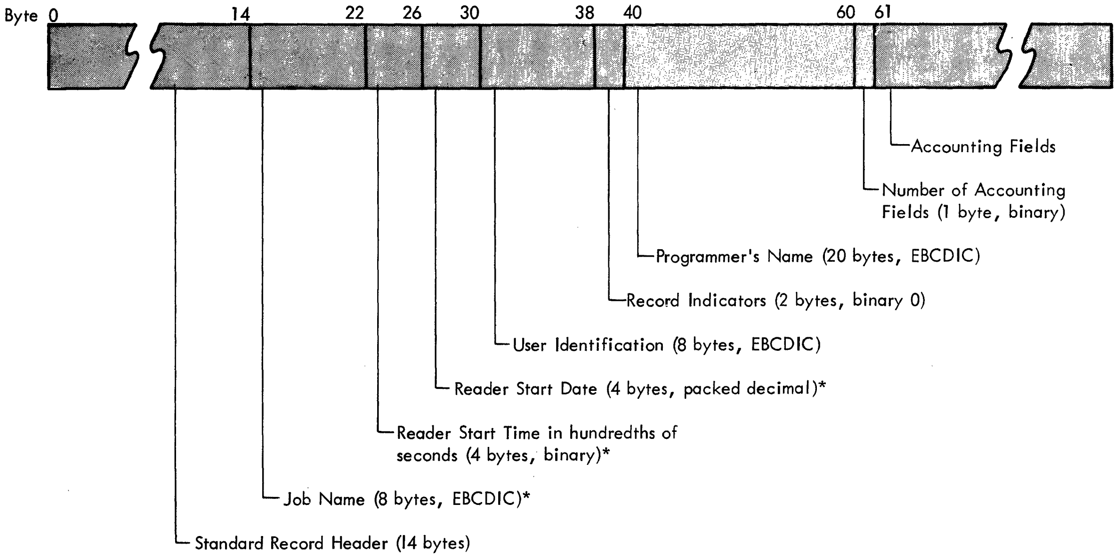
Owner identification (10 bytes, EBCDIC)

Volume Serial Number (8 bytes)
Bytes 0 and 1 reserved (binary 0)
Bytes 2 thru 7 contain the Volume Serial Number (EBCDIC)

Standard Record Header (14 bytes)

Record Type 20 (Job Commencement Record)

This record is written at job initiation when data set accounting and/or direct access volume information is specified. Length is variable.


Byte 0
14
22
26
30
38
40
60
61
Accounting Fields
Number of Accounting Fields (1 byte, binary)
Programmer's Name (20 bytes, EBCDIC)
Record Indicators (2 bytes, binary 0)
User Identification (8 bytes, EBCDIC)
Reader Start Date (4 bytes, packed decimal)*
Reader Start Time in hundredths of seconds (4 bytes, binary)*
Job Name (8 bytes, EBCDIC)*
Standard Record Header (14 bytes)

* The job name, reader start time, and reader start date fields constitute the job log number.

Section 7: SMF Messages

SMF writes the following messages to the operator's console or the SYSOUT writer. Messages having numbers with an "I" suffix are informational; i.e., the system will continue to operate, without waiting for a response. Messages with numbers having an "A" suffix indicate that some operator action is required before operation of the system will continue.

IEE350I SMF SYS1.MAN$\left\{\begin{matrix}X\\Y\end{matrix}\right\}$NOT DEFINED

Explanation: The SYS1.MANX or SYS1.MANY data set, as indicated in the message text, is required but was not defined in the SMFDEFLT data set or by the operator from the keyboard.

System Action: System operation continues, but no records are written in the SYS1.MAN data set.

Operator Response: Inform the system programmer to define the SYS1.MAN data set in the SMFDEFLT data set.

IEE351I SMF SYS1.MAN RECORDING NOT BEING USED

Explanation: During SMF initialization, either space was not allocated for the SYS1.MAN data set or the data control block for the SYS1.MAN data set could not be opened. Message IEE350I, IEE358I, or IEE363I precedes this message, indicating the cause of the error.

System Action: System operation continues, but no records are written in the SYS1.MAN data set.

Operator Response: Inform the system programmer to properly define the SMFDEFLT data set, as indicated in the response to the preceding message.

IEE352A SMF MEMBER MISSING - REPLY WITH SMF VALUES

Explanation: During SMF initialization, the SMEDEFLT data set was not found in the SYS1.PARMLIB data set.

System Action: SMF initialization will not continue until the operator has responded to this message.

Operator Response: Enter REPLY xx, 'keyword=value,keyword=value', listing all the required SMFDEFLT parameters as defined by the system programmer at your installation. Inform the system programmer to define the SMFDEFLT data set in the SYS1.PARMLIB data set, thereby eliminating the need to redefine the SMFDEFLT parameters from the keyboard at every SMF initialization.

IEE353A I/O ERROR ON SMFDEFLT READ - REPLY WITH SMF VALUES OR RE-IPL

Explanation: During SMF initialization, an uncorrectable input/output error occurred while reading or searching for the SMFDEFLT data set in the SYS1.PARMLIB data set.

System Action: SMF initialization will not continue until the operator has responded to this message.

Operator Response: Restart the system. If the error persists, enter REPLY xx, 'keyword=value,keyword=value', listing all the required SMFDEFLT parameters as defined by the system programmer at your installation. If the error still persists, call a customer engineer.

IEE354I SMF PARAMETERS

Explanation: This message is issued if OPI=YES was specified in the SMFDEFLT data set. All the parameters of the SMFDEFLT data set are listed, one parameter to a line, following this message; each parameter is listed in the format keyword=value. Message IEE357A follows, permitting changes to be made.

System Action: None.

Operator Response: None.

IEE355I SMF PARAMETER ERRORS

IEE355I $\left\{\begin{matrix}\text{xxxx UNRECOGNIZABLE KEYWORD/FORMAT}\\ \\ \text{keyword=value INVALID VALUE SPECIFIED}\\ \\ \text{keyword - KEYWORD NOT SPECIFIED}\end{matrix}\right\}$

Explanation: During SMF initialization, a parameter was specified incorrectly either in the SMFDEFLT data set or in a reply from the console. The specific error is listed in the second line of the message:

xxxx UNRECOGNIZABLE KEYWORD/FORMAT
An unrecognizable keyword was detected; xxxx specifies up to 25 characters of the keyword in error.

keyword=value INVALID VALUE SPECIFIED
The keyword specified is valid, but the value specified is invalid.

keyword - KEYWORD NOT SPECIFIED
The keyword specified is required, but the keyword was not found in the SMFDEFLT data set.

Message IEE356A follows, permitting changes to be made.

System Action: None.

Operator Response: None.

IEE356A REPLY WITH SMF VALUES

Explanation: This message follows message IEE355I, and permits corrections to be made to the parameters in error.

System Action: SMF initialization will not continue until the operator has responded to this message.

Operator Response: Enter REPLY xx, 'keyword=value,keyword=value', correcting the parameters in error indicated in message IEE355I (as defined by the system programmer at your installation). Inform the system programmer to correct the errors in the specification of the SMFDEFLT data set.

IEE357A REPLY WITH SMF VALUES OR U

Explanation: This message follows message IEE354I and permits changes to be made to parameters listed.

System Action: SMF initialization will not continue until the operator has responded to this message.

Operator Response: If any parameters are to be changed, enter REPLY xx,'keyword=value,keyword=value', specifying the changes desired (as defined by the system programmer at your installation). If parameters are not to be changed, enter REPLY xx, 'U'.

IEE358I SMF SYS1.MAN$\begin{Bmatrix}X\\Y\end{Bmatrix}$NOT FOUND ON utn

Explanation: The SYS1.MANX or SYS1.MANY data set, as indicated in the message text, was specified for the device whose unit address is utn. However, no space was allocated for the data set on that device.

System Action: System operation continues, but no records are written in the SYS1.MAN data set.

Operator Response: Inform the system programmer either to allocate space for the data set on the indicated device or to redefine the data set in the SMFDEFLT data set.

IEE360I SMF NOW RECORDING ON SYS1.MAN$\begin{Bmatrix}X\\Y\end{Bmatrix}$ON utn TIME=hh.mm.ss

Explanation: Records are being written in the SYS1.MANX or SYS1.MANY data set, as indicated in the message text, on the device whose unit address is utn. The message also indicates the time of day, where hh specifies the hour (00-23), mm specifies the minute (00-59), and ss specifies the second (00-59).

System Action: None.

Operator Response: None.

IEE361I SMF DATA LOST -- SYS1.MANX/Y NOT AVAILABLE TIME=hh.mm.ss

Explanation: The SYS1.MANX and SYS1.MANY data sets are both full. Therefore, no more records can be written. The message also indicates the time of day, where hh specifies the hour (00-23), mm specifies the minute (00-59), and ss specifies the second (00-59).

System Action: System operation continues, but no records are written in the SYS1.MAN data sets. Until a data set becomes available, a record is kept of the number of records lost and of the starting and ending times of the period during which no records were written. When a data set becomes available, an SMF data lost record (type 7) is written.

Operator Response: Initiate execution of the IFASMFDP program. If a dump program is currently executing, ensure that it completes as soon as possible.

```
IEE362A SMF ENTER DUMP FOR SYS1.MAN{X}ON
        utn                        {Y}
```

Explanation: The SYS1.MANX or SYS1.MANY data set, as indicated in the message text, is full or end-of-day was specified. The data set is on the device whose unit address is utn.

System Action: None.

Operator Response: Initiate execution of the IFASMFDP program for the indicated data set.

```
IEE363I SMF{ser}{DEVICE CAPACITY TOO SMALL}
           {utn}{FOR BUFFER               }
                {DEVICE NOT IN SYSTEM     }
                {INCORRECT DEVICE TYPE    }
                {NOT DIRECT ACCESS        }
                {OFFLINE                  }
                {VOLUME IS NON-SHAREABLE  }
```

Explanation: Space for the SYS1.MAN data set defined in the SMFDEFLT data set cannot be allocated for one of the following reasons:

DEVICE CAPACITY TOO SMALL FOR BUFFER
The specified device cannot contain the largest possible SYS1.MAN record.

DEVICE NOT IN SYSTEM
No unit control block exists for the specified device.

INCORRECT DEVICE TYPE
The wrong device type was specified for the SYS1.MAN data set.

NOT DIRECT ACCESS
A device other than a direct access device was specified for the SYS1.MANY data set.

OFFLINE
The unit control block for the device is marked offline.

VOLUME IS NON-SHAREABLE
A device was specified for the SYS1.MAN data set that could not be allocated to SMF.

In the message text, ser is the serial number of the volume and utn is the unit address of the device containing the SYS1.MAN data set.

System Action: System operation continues, but no records are written in the SYS1.MAN data set.

Operator Response: Inform the system programmer to properly define the device in the SMFDEFLT data set or to make the specified device available.

IEE364I SMF I/O ERROR ON utn

Explanation: A permanent input/output error occurred while writing in the SYS1.MAN data set on the device whose unit address is utn.

System Action: For direct access: if an alternate data set is available, records will be written on that data set; if no alternate data set is available, no records will be written.

For tape: the data control block for the data is closed; however, the system will request another tape volume to be mounted, and the data control block will be subsequently reopened.

Operator Response: Inform the system programmer to define a different data set in the SMFDEFLT data set. If the error persists, call a customer engineer.

IEF373I STEP/sss START/yyddd.hhmm

Explanation: At step termination for SMF, this message indicates the time and date that step sss was started.

In the message text, yy specifies the year, ddd specifies the day of the year (001-366), hh specifies the hour (00-23), and mm specifies the minute (00-59).

Programmer Response: None.

IEF374I STEP/sss STOP/yyddd.hhmm
CPU xxxxMIN xx.xxSEC MAIN xxxxK
LCS xxxxK

Explanation: At step termination for SMF, this message indicates the time and date that step sss was terminated, the step problem program CPU time, and the total storage used.

In the message text, yy specifies the year, ddd specifies the day of the year (001-366), hh specifies the hour (00-23), and mm specifies the minute (00-59). For the CPU time, xxxxMIN specifies the minute and xx.xxSEC specifies the second (in seconds and hundredths of a second). Also, the MAIN xxxxK specifies the processor storage (hierarchy 0) and the LCS xxxxK specifies the IBM 2361 core storage (hierarchy 1).

Programmer Response: None.

IEF375I JOB/jjj START/yyddd.hhmm

Explanation: At job termination for SMF, this message indicates the time and date that job jjj was started.

In the message text, yy specifies the year, ddd specifies the day of the year (001-366), hh specifies the hour (00-23), and mm specifies the minute (00-59).

Programmer Response: None.

IEF376I JOB/jjj STOP/yyddd.hhmm
CPU xxxxMIN xx.xxSEC

Explanation: At job termination for SMF, this message indicates the time and date that job jjj was terminated and the job problem program CPU time.

In the message text, yy specifies the year, ddd specifies the day of the year (001-366), hh specifies the hour (00-23), and mm specifies the minute (00-59). For the CPU time, xxxxMIN specifies the minute and xx.xxSEC specifies the second (in seconds and hundredths of a second).

Programmer Response: None.

Index

IBM®

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

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

READER'S COMMENT FORM

IBM System/360 Operating System
Planning for System Management Facilities

Form C28-6712-1

- Is the material:

	Yes	No
Easy to read?	□	□
Well organized?	□	□
Complete?	□	□
Well illustrated?	□	□
Accurate?	□	□
Suitable for its intended audience?	□	□

- How did you use this publication?
  □ As an introduction to the subject
  □ For additional knowledge
  Other ....

- Please check the items that describe your position:
  □ Customer personnel
  □ IBM personnel
  □ Manager
  □ Systems Analyst
  □ Operator
  □ Programmer
  □ Customer Engineer
  □ Instructor
  □ Sales Representative
  □ Systems Engineer
  □ Trainee
  Other ....

- Please check specific criticism(s), give page number(s), and explain below:
  □ Clarification on page(s) ....
  □ Addition on page(s) ....
  □ Deletion on page(s) ....
  □ Error on page(s) ....

Explanation:

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

YOUR COMMENTS, PLEASE . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Note: Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Cut Along Line

Fold Fold

FIRST CLASS
PERMIT NO. 81
POUGHKEEPSIE, N.Y.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY ...

IBM Corporation
P.O. Box 390
Poughkeepsie, N.Y. 12602

Attention: Programming Systems Publications
Department D58

Fold Fold

IBM®

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

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]