CA SMF Director®

User Guide

12.6.00

This documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Common Services for z/OS
- CA MICS® Resource Management
- CA JARS® Resource Accounting
- CA Service Desk
- CA 7® Workload Automation

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA Technologies product documentation, complete our short customer survey, which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 4: SMFD Control Statement Reference 35

Chapter 5: SMFDLS Control Statement Reference 121

Chapter 1: Introduction

Even though IBM Systems Management Facilities Data (SMF) is important, depending upon your business' use of chargeback analysis and reporting, resource accounting and billing, and/or capacity planning, it is cumbersome to manage. It places a heavy burden on the Computer Operator, who traditionally has had to copy SMF data to tape, record its location, and protect the volume from inadvertently being overwritten. This manual procedure is obviously error prone. If an installation has manual tape management procedures, errors are even more liable to occur, causing loss of valuable SMF records. Failure to dump and clear SMF SYSx.MANx files or SMF logstreams in a timely fashion can degrade overall system performance and in some situations halt the operating system.

CA SMF Director is a software product for the management of data sets containing SMF data in a z/OS environment. When properly used it can be a cost effective tool for tracking and maintaining SMF data at your installation.

The following features are provided with this product:

- Automatic Unloading of SYSx.MANx Files
- Automatic Copying of SMF Data from Logstreams
- Maintaining Inventory of SMF Records by Date, Time, and Volume Serial
- Optionally segregate SMF data into separate history archives with separate retention periods based on the SMF record type
- Protecting Data Sets Containing SMF Records
- Printing and Extracting SMF Records by Date and Time
- Optionally segregating SMF data by day onto a single tape volume
- Providing Automatic Duplexing and Multiplexing of SMF Records
- Providing SMF Record Verification
- Providing Automatic Magnetic Tape Management
- Integrating with Other Products
- Online Query Interface

These features are described in detail on the following pages.

Features

Automatic Unloading of SYSx.MANx Files

This product can be customized to automatically unload (dump and clear) the SMF records from switched or full SYSx.MANx files to history files. The product determines which data set should be unloaded by inspecting the System Control Blocks as well as the actual SYSx.MANx files.

The history files can reside on DASD devices, magnetic tape volumes, or on Optical Library CDs.

CA SMF Director can honor explicit dump requests, and replace the need to use the IBM SMF dump program IFASMFDP.

Automatic Copying of SMF Data from Logstreams

This product can be configured to automatically copy SMF data from the logstreams that SMF uses to record data. The history files created will be identical to those created when SMF data is recorded to the MAN files.

Unlike MAN file dumping, where IFASMFDP is replaced, CA SMF Director uses the IBM logstream dumping utility, IFASMFDL, to access the data from the logstream for dumping. This utility is controlled by a driver program and three user exits that are provided.

Data is copied from SMF logstreams but not cleared from the logstreams. You can use standard logstream management techniques to have the data deleted from the logstreams after it has reached a certain age.

Maintaining Inventory of SMF Records by Date, Time, and Volume Serial

Upon successful completion of the dump process, CA SMF Director keeps track of the date and time of the SMF records placed on a history file, the volume serial of the device where the history data set resides, along with the date and time when the dump process was performed. This eliminates the need for manual recording/tracking of data sets containing SMF records. Optionally an entry for a history file can also be added to the Operating Systems Catalog.

If expiration dates/retention periods are desired for the history files, this product can be customized to automatically delete any entries referring to the expired history data set, and optionally uncatalog the data set from the system catalog.

Optional Segregation of SMF Records

With CA SMF Director it is possible to break up your SMF MAN file records into several historical archives called substreams. These substreams can be set up with different retention periods, archival methods, and other properties. This way, records can be kept for different lengths of time or on different media based on their record types.

Protecting Data Sets Containing SMF Records

Various control keywords are provided in CA SMF Director allowing the specification of the retention period or expiration date of a history file. Permanent retention of history files is also provided if desired by specifying a never-expire expiration date such as 99.366. Because the history files are created with an expiration date, it minimizes the possibility of destroying a history file containing SMF records.

Printing and Extracting SMF Records by Date and Time

Because CA SMF Director maintains a catalog of the SMF records and history files, you can extract or print SMF records by date, time and/or record type without regard to what file or device contains the desired SMF records. This product can be customized to dynamically allocate the desired history file data sets to perform the requests.

It is also possible to copy all or any part of the SMF data to one or more data sets for analysis and further processing, depending upon SMF ID, time period, and record type.

Optionally Segregating SMF Data by Day onto a Single Tape Volume

The DAYSTACK feature forces CA SMF Director to request a new tape volume when the end of the day is reached during a dump. The end of the day (and start of the next day) is defined by the SDAY option.

The DAYSTACK feature also determines whether a new dump goes on a new tape or on the same tape used in the previous dump. A new tape is used if the timestamp of the first record in the dump is after the SDAY boundary of the last record on the old tape.

Providing Automatic Duplexing and Multiplexing of SMF Records

CA SMF Director can automatically create a duplicate copy of a history file during the dump/unload process.

In addition to duplexing, CA SMF Director can also be directed to write a portion of the SMF records being dumped to other output files. These records can then be processed immediately, without an intervening step to extract them from the history files that were just dumped.

Data management information is also provided for the additional output files, called split files, in the form of an index, called the split file index. The index itself is managed and used, but a copy of it can be produced by CA SMF Director for use by other applications, if needed.

Providing SMF Record Verification

CA SMF Director checks the standard SMF record header information for validity, according to the definition contained in the IBM *System Management Facilities Manual*. Any records not meeting the proper requirements are printed for further investigation.

As part of its verification, this product checks to make sure all records in a particular MAN file contain the same SMF ID. If the dump process encounters SMF records containing more than one SID, it issues this error message: CAF1232E SMF DATA FROM MULTIPLE SYSTEMS WAS DETECTED. It then aborts the dump.

Providing Automatic Magnetic Tape Management

If the use of magnetic tape volumes is desired for SMF history files, CA SMF Director automatically determines which tape is eligible for use by the dump/unload process. This product can be customized to request non-specific tape volumes (scratch tapes) or specific tape volumes. A specific "pool" of volume serials can be defined to CA SMF Director from which it chooses a tape volume for the mount request. After one of these volumes expires (if allowed), it is set to be reused.

Note: A specific pool of tapes is not allowed for substream or logstream archival.

Regardless of whether CA SMF Director requests specific or non-specific tape volumes, this product fills up the tape volume as much as possible with multiple files, ensuring full use of a tape volume.

CA SMF Director can be easily integrated to be used with CA 1, CA TLMS, CA Vtape Virtual Tape System, or any other tape management facility.

Integrating with Other Products

This product was primarily designed to manage the storing of SMF data for long-term use on magnetic tape or DASD. Additionally, it can be used in conjunction with Storage Management products such as CA ASM2 or IBM's DFSMS. This is accomplished by using storage devices under management of the Storage Management product for the SMF history files. If this is done, the maintaining of the SMF data for long term use becomes the responsibility of the Storage Management product. For example, the primary pool can use DASD or optical disk storage devices managed by DFSMS. Storage management attributes defined to DFSMS for the SMF history files determine when these data sets are archived, backed up to another medium, or scratched. Long term storage of the SMF data is dependent upon the Management Class attributes in DFSMS, and not on the retention period or expiration date values defined to CA SMF Director.

This product can be used with other performance management and resource accounting products such as CA JARS or CA MICS.

Online Query Interface

CA SMF Director provides an ISPF interface to the CA SMF Director Control Data Set, SCDS, that lets you view most of the information in the SCDS. To use the online interface, see the chapter, Online Interface to SCDS.

Program Operation

The SMFD program is at the heart of CA SMF Director. As the system's intelligent dispatcher, it tests for selection criteria and calls the individual modules that actually perform the tasks of dumping the SMF data, taking inventory of data, and executing selective processing requests.

When dump processing begins, CA SMF Director allocates the history files based on default allocation parameters defined in the SCDS, as well as parameters coded in the history DD statements, if present. If writing to tape, it calls for the volume to which the last dump was written. It then empties the SMF data set to a history file and updates its history index in the SCDS.

The SCDS is a DASD data set that contains all the index entries for dumped SMF files, including cross-references to their location. This product uses tapes until they are filled. A history tape may contain the contents of several SMF dumps. With these index entries you can access specific records for extraction and printing.

Once the DUMP process is complete, you can access the SMF records for extraction or printing by:

- Date and time
- SYSID
- Record type

CA SMF Director locates the index entry that satisfies the EXTRACT or PRINT request, and allocates the corresponding history file. The history file is read copying the SMF records to the specified output location.

If SMF logstream recording is being used with z/OS 1.9 or above, CA SMF Director also provides an interface program for dumping, SMFDLS. This program will interface with the IBM Utility IFASMFDL via user exits and then activate CA SMF Director dynamically to manage the data being dumped to the history files.

Chapter 2: Customizing CA SMF Director

Before you customize CA SMF Director, you must plan the following:

- How often SMF data is unloaded from SYSx.MANx files or from SMF logstreams.
- Where the SMF data is kept (DASD or TAPE).
- How long the SMF data is kept.
- What the naming convention is for the SMF data.
- Whether a duplicate copy of the SMF data is to be maintained, and for how long (duplexing).

Setting up the Files

Configuration information is stored on a direct access data set known as the SCDS file (CA SMF Director Control Data Set). This is also the repository for the inventory of SMF data as well as the optional pool of magnetic tape volumes to be used by this product. For further details on how to define a configuration, see Customization Considerations in Chapter 1 of the *Systems Programmer Guide*.

This product can be configured to manage the SMF data within one or two history file pools. They are the primary and alternate pools.

Primary Pool

The primary pool is the main repository for the SMF records. Files created in the primary pool can reside on magnetic tape or DASD. If DASD is used, you may use DFHSM or another Storage Management product such as CA ASM2 to manage the data sets within this pool, allowing archival to tape after a predetermined period of time. When an extract or print request is issued, DFHSM restores the data set to its original DASD volume, allowing the request to be satisfied.

For more information on a tape pool of volume-specific tapes, see the DUMPTAPES control statement. For more information on a tape pool of non-specific tapes, see the SCRATCH operand on the OPTIONS, DUMPOPTIONS, and STREAMOPTIONS control statements.

Alternate Pool

The alternate pool is used when duplexing is specified. CA SMF Director keeps a second entry for this history file created on the SCDS for this pool. You can specify whether this product is to extract or print SMF records using the primary or the alternate copy of the history file containing the desired SMF records. Files created in the alternate pool can reside on magnetic tape or DASD.

Note: We recommend that you use duplexing to ensure that data is preserved in case something happens to the primary pool.

Configuring CA SMF Director

You customize CA SMF Director by defining configuration options. These customized configuration options are maintained in the SCDS. Three levels of granularity are available when configuring this product:

- Complex-wide configuration options. These options define specifications to CA SMF Director, regardless of the z/OS system within the computing complex on which it executes. These are specified on the OPTIONS control statement.
- System configuration options. These options control how this product will create and maintain the SMF history files. Every system that generates SMF data that this product manages must also be defined to this product. Options that are specific to a particular system are specified on the DUMPOPTIONS control statement.
- Stream configuration options. These options allow overrides for a particular stream of SMF records, whether they come from an SMF logstream or from a CA SMF Director substream. With this level, records can be managed more effectively based on business needs. Options that are specific to only one stream are specified on the STREAMOPTIONS control statement.

The DUMPOPTIONS control statement is used to indicate which logstreams or substreams are to have their data managed by CA SMF Director. The other operands on the DUMPOPTIONS statement set the default SMF record management parameters. The STREAMOPTIONS statement can be used to override most of the DUMPOPTIONS as needed for a specific stream that is being used by a system for SMF records.

Granular Options

CA SMF Director allows for SMF records to be managed based on business needs. Records can be retained for various periods of time based on the originating system, the logstream (if logstream SMF recording is in use), and the record type. The following retention attributes can be managed at any level (complex, system, and stream):

- Whether or not the archive should be duplexed
- Whether or not the history files in the archive should be cataloged
- The retention period or expiration date of the history files (if any)
- The data set prefix of the history files within the archive
- The tape recording technique for history files written to tape
- Whether or not there should be a forced end-of-volume if the date changes in the SMF records
- The time of day that should be considered to be the start of a new day
- The unit or SMS storage class where the history files should be allocated
- The maximum number of files that should be written to a single tape volume

Several additional options are available on both a system basis and a complex-wide basis:

- Whether or not CA SMF Director should use scratch tapes for tape history files. (Using scratch tapes is required for any systems with logstream or substream archives.)
- Whether or not CA SMF Director should dynamically allocate history files. (Dynamic allocation is required for any systems with logstream or substream archives.)

Complex-Wide Configuration Options

On the OPTIONS control statement you may specify:

- A site name for your complex and/or the maximum number of lines per page for CA SMF Director SYSOUT.
- Optionally, this product can be set up to open requests in CA Service Desk in the unlikely event of an error during the dump process.

System Configuration Options

CA SMF Director requires that each z/OS system that generates SMF data be defined. You may specify a comment describing each system along with the options that control the creation of SMF history files, for example, naming all of the logstreams if SMF is recording data to logstreams, or naming all of the substreams if data is going to be archived in separate streams on a single system.

Defining a Configuration to CA SMF Director

A CA SMF Director configuration definition can be produced through the execution of the batch program. The following example shows how a configuration is defined to this product. See member CASFCTL1 in the CASFJCL library, loaded during the installation process.

```
//CASFUTIL JOB ... MSGLEVEL=(1,1),CLASS=A
//STEP1  EXEC PGM=SMFD,REGION=512K
//SCDS    DD DISP=SHR,DSN=CORP.SMFD.SCDS
//SYSPRINT DD SYSOUT=*
//SYSXDIAG DD SYSOUT=*
//SYSIN    DD *
   COMPILE CONFIGURATION STORE.
     OPTIONS AUTODEL DYNAM MAXLINES(56) SDAY(0600).              (2)
      BEGIN CONFIG(050) SID(XAD1) NAME('CORPORATE-NAME').        (1)
           DUMPOPTIONS COPIES(2) CATLG SCRATCH NOVERIFY
             RETPD(730) PREFIX(SYS2.SMFDATA)                     (3)
             ARETPD(90) APREFIX(SYS3.SMFDATA).
      END CONFIG(050).
   COMPILE CONFIGURATION STORE.
      BEGIN CONFIG(052) SID(H001)  NAME('CORP TEXT SYSTEM-1').
           DUMPOPTIONS COPIES(1) CATLG SCRATCH NOVERIFY          (4)
           LSNAME(IFASMF.ACCT IFASMF.RMFDATA IFASMF.TCPIP        (5)
                IFASMF.CICS IFASMF.DB2 IFASMF.OTHER)
           RETPD(90) PREFIX(SYS2.H001.SMFDATA).
      END CONFIG(052).
   COMPILE CONFIGURATION STORE.
      BEGIN CONFIG(053) SID(H002)  NAME('CORP TEXT SYSTEM-2').
           DUMPOPTIONS COPIES(1) CATLG SCRATCH NOVERIFY          (4)
           LSNAME(IFASMF.ACCT IFASMF.RMFDATA IFASMF.TCPIP        (5)
                IFASMF.CICS IFASMF.DB2 IFASMF.OTHER)
           RETPD(90) PREFIX(SYS2.H002.SMFDATA).
      END CONFIG(053).
/*
//
```

Note:

1. In this example, configurations for three z/OS systems with the SMF IDs of XAD1, H001 and H002 are defined.
2. The OPTIONS control statement allows the setting of product options, such as the maximum number of lines per page, and what time of day is considered to be the beginning of a logical work day.
3. The XAD1 configuration retains history file data sets with the high level qualifiers of 'SYS2.SMFDATA' for 2 years, and 90 days with the high level qualifier of "SYS3.SMFDATA".
4. Since the configurations specify SCRATCH, any available scratch tape volume will be used to dump SMF data. However, once a tape volume is started, future dumps will go to that tape volume until it is full, or until the maximum number of dumps on a single volume is reached.
5. If CA SMF Director is going to archive and manage SMF data that is being recorded to logstreams, these logstreams must be named in the LSNAME operand in the DUMPOPTIONS control statement. If a logstream is being shared between two or more systems, it must be named in all of the configurations for all of the systems that are recording data to that logstream.
6. If CA SMF Director is going to archive and manage SMF data that is being recorded to substreams, these substreams must be named in the SUBNAME operand in the DUMPOPTIONS control statement.

Defining Service Desk Support

When defining configurations, CA SMF Director can be configured to create Service Desk requests in the unlikely event of an error during the dump process. Events are split into two categories:

High Priority Requests

High priority Service Desk requests are opened when there is a problem that could result in a loss of SMF data if the dump process does not complete:

- Allocation error on history file. CA SMF Director can dynamically allocate history files where the SMF data will be dumped during the dump process. If the file does not get allocated, the dump process will terminate before any data is dumped.
- I/O error on history file. If one of the output history files encounters an error while being written to, the dump process will terminate before all of the data in the input SMF data set is dumped.

Low Priority Requests

Low priority Service Desk requests are issued if something goes wrong during the Dump Process, but there is no risk of the loss of SMF data because the history files were written to completion as intended:

- Allocation and deallocation errors on split files.
- I/O error on split file.
- Error cataloging history file. The data was successfully written to the history file, but the attempt to CATALOG the data set was unsuccessful. The Service Desk request will indicate the catalog operation that is needed to catalog the history file.
- Overlap error in SMF data written to history file.

To enable CA SMF Director to open Service Desk requests, you must code the SERVICEDESK operand on the OPTIONS statement when compiling configurations. If the indication is made and an error occurs during the Dump Process, which is defined as a Service Desk event, CA SMF Director will make one and only one attempt to open a Service Desk request.   To allow for greater flexibility in the creation of Service Desk requests, additional operands provided on the OPTIONS control statement can indicate which instance of Service Desk is to be used (SDURL), what user ID and password work for that instance (SDUSERID, SDPASSWORD), and what Service Desk severity levels should be used for high priority and low priority requests.

In order for the Dump Process to open Service Desk requests successfully, you must do the following:

- The CA Common Services SOAP Client Service (CAISDI/soap) address space must be installed and configured. For information on configuring the SOAP Client Service, see the *CA Common Services for z/OS Service Desk Integration Guide.*
- Service Desk must be installed at your site. See the Service Desk documentation for installation and setup procedures for Service Desk.

Unloading SMF Data Sets

Once the desired configuration has been defined, this product is ready to manage your SMF data. If you have existing files that contain SMF records, but were not managed by this product, you can copy the existing data sets to CA SMF Director history files. During this process, this product creates an inventory of what data resides on the output history file and stores it on the SCDS file. You can use CASFJCL target library member CASFDMP4 to perform this function.

The CA Common Services for z/OS program CAIRIM is used by CA SMF Director to enable or disable the automatic dump feature (refer to the *Installation Guide* for more details).

CA SMF Director can be used to unload (dump and clear) the SYSx.MANx z/OS operating system SMF data sets at the operator's discretion, at specific times by a job scheduling product such as CA SCHEDULER, or automatically when an SMF data set is switched or becomes full. Procedure CASFDUMP has been provided in the CASFPROC library to perform this function. It can be modified to meet your site requirements. In addition, the procedure may be modified to indicate one or more files, called split files, that will be written to as indicated by SPLIT control statements (See Chapter 4 for more details on the SPLIT control statement.)

CASFJCL target library member CASFDMP1 illustrates how the CASFDUMP procedure can be invoked to start the dump process using a batch job. An operator can also start the dump process manually by entering the z/OS START command:

```
S CASFDUMP,ALTDS=DUMMY
```

In the unlikely event that CA SMF Director is unable to unload the SMF data from the SYSx.MANx files, the IBM utility IFASMFDP can be used to dump and clear the SMF data sets, and later the output file created by IFASMFDP can be copied or manually added by CA SMF Director (see the next two sections for further details).

Unloading SMF Logstreams

Versions of z/OS from 1.9 and above can also write SMF data to system logger logstreams instead of to the traditional SYSx.MANx files. This change in operation will still work with CA SMF Director, and setting up to process logstream data is similar to setting up for the SYSx.MANx files. The major difference is that the logstream must be provided to the logstream dump driver program SMFDLS in the input control statements or via the JCL.

We recommend that you modify and use the CASFDUML procedure that has been included in the CASFPROC target library if your logstream dumping needs are similar for all SMF logstreams. The procedure can be invoked with a variable that is the name of the logstream to be processed.

The CA Common Services for z/OS program CAIRIM is used by CA SMF Director to enable or disable the automatic dump feature (see the *Installation Guide* for more details). The intercept for IEFU29L can be used if desired, or dumps can be started on a regular basis using scheduling software such as CA 7.

Samples of control statements and JCL for SMFDLS are provided for setting up dump processes. The following members are available in the target libraries listed below:

Item Name	Target Library	Description
CASFDUML	CASFPROC	Automated dumping procedure invoked by the CA SMF Director IEFU29L intercept if installed. This procedure can also be started via a system command or within a batch job.
SMFDDUMP	CASFPARM	Standard control statements for the CA SMF Director dump process.
SMFDIN	CASFPARM	Standard control statements for the CA SMF Director logstream dump interface program.
SMFLSTBL	CASFSRC	Sample user exit that can be used to tailor the commands that are issued to the system to automatically start the dumping of logstream data by CA SMF Director.

Typically, dump time range information will not be needed for SMFDLS, which will automatically determine where to resume the dump for a given logstream and SYSID. Once the dump has completed, information will be kept to determine where to begin the next dump.

Placing Existing SMF Data under CA SMF Director Control

Placing a Single System Archive under Control

You can place previously dumped SMF data under CA SMF Director control manually (such as when logging tapes at a remote site) by using the following JCL and control statements:

```
//CASFCTL1 JOB (ACCTINFO),PGMR,CLASS=A,MSGCLASS=A
//*
//* THIS SAMPLE JCL ILLUSTRATES HOW CA SMF Director CONTROL
//* STATEMENTS CAN BE SPECIFIED.
//*
//* NOTE: REFER TO YOUR USER GUIDE
//*       FOR FURTHER INFORMATION ABOUT:
//*
//*        . CUSTOMIZING CA SMF Director
//*        . SELECTIVE PROCESSING OF SMF DATA
//*        . COMMAND REFERENCE
//*
```

```
//STEP1   EXEC PGM=SMFD,REGION=2048K
//STEPLIB  DD  DISP=SHR,DSN=CASFLOAD
//SCDS     DD  DISP=SHR,DSN=CORP.SMFD.SCDS
//SYSPRINT DD  SYSOUT=*
//SYSXDIAG DD  SYSOUT=*
//SYSIN    DD  *
*
* -----------------------------------------------------------------
* MANUALLY ADD SMF DATA INDEX INFORMATION TO THE CONTROL DATA SET
* FOR THE FOLLOWING DATA SETS FROM SYSID H002:
* 1. SYS4.SEPT.D093099 - 09/30/yyyy 16:30:58 TO 09/30/yyyy 16:59:58
* 2. SYS4.JUNE.D060100 - 06/01/yyyy 08:00:25 TO 06/02/yyyy 23:58:10
* -----------------------------------------------------------------
*
ADDX SID(H002) LOW(093099 163058) HIGH(093099 165958)
     VOLUME(122318) FILESEQ(1)  DUMPD(060300 013000)
           DSNAME(SYS4.SEPT.D093099) UNIT(3490).
ADDX SID(H002) LOW(060100 080025) HIGH(060200 235810)
     VOLUME(122316) FILESEQ(1)  DUMPD(060300 013000)
           DSNAME(SYS4.JUNE.D060100) UNIT(3490).
/*
//
```

CASFJCL target library member CASFCTL1 can be used as a prototype to create the above job stream. For a complete description of the ADDX control statement, see Chapter 4 of this guide.

To put existing SMF data under CA SMF Director control by copying the SMF records, CASFJCL member CASFDMP4 can be used as a prototype. The provided job stream shown below instructs CA SMF Director to locate the SMF data in data set CORP.YR1992.SMFDATA using the ddname SMFDATA, and copy it to the primary history file. During the copy process CA SMF Director automatically creates an index entry in the SCDS for the SMF records copied, thus producing an inventory of the SMF data.

```
//CASFDMP4   JOB...
//STEP 1  EXEC  PGM=SMFD,REGION=4096K
//STEPLIB  DD   DISP=SHR,DSN=CASFLOAD
//SCDS     DD   DISP=SHR,DSN=CORP.SMFD.SCDS
//SYSPRINT DD   SYSOUT=A
//SYSXDIAG DD   SYSOUT=A
//SMFDATA  DD   DISP=SHR,DSN=CORP.YR1992.SMFDATA
//SYSIN    DD   *
  SOURCE DUMP(SMFDATA).
  DUMP.
/*
//
```

For complete descriptions of the SOURCE and DUMP control statements, see Chapter 4 of this guide.

Placing a Multiple System Archive under Control

The architecture of the CA SMF Director historical archive requires that the SMF data be managed by a single SMF ID. For sites that have archives of SMF data with multiple SMF IDs, you must separate the data by SMF ID before it can be added to the CA SMF Director archive.

The most efficient way to do this is to set up a multiple-step job that reads the archived data with IFASMFDP. IFASMFDP can then be directed to create one temporary output data set for each SMF ID that you want to migrate. Then, in subsequent steps, CA SMF Director can be invoked for each system in the manner described above.

As an example, suppose one of your archives contains SMF data from two systems with the SMF IDs SYSP and SYST. The JCL to migrate this data would look like this:

```
//         JOB
//STEP001  EXEC PGM=IFASMFDP,REGION=0M
//SYSPRINT DD   SYSOUT=*
//SMFIN    DD   DSN=your.archive.data.set,DISP=SHR
//SYSPOUT  DD   DSN=&&SYSP,DISP=(NEW,PASS),STORCLAS=TEMP,
//              SPACE=(CYL,(200,100),RLSE),
//              DCB=*.SMFIN
  INDD(SMFIN,OPTIONS(DUMP))
  OUTDD(SYSPOUT)
  SID(SYSP)
/*
//STEP002  EXEC PGM=SORT,REGION=6M
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//STEPLIB  DD  DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01 DD  SPACE=(CYL,(300,50)),UNIT=SYSDA
//SORTWK02 DD  SPACE=(CYL,(300,50)),UNIT=SYSDA
//SORTWK03 DD  SPACE=(CYL,(300,50)),UNIT=SYSDA
//SORTWK04 DD  SPACE=(CYL,(300,50)),UNIT=SYSDA
//SORTIN  DD  DSN=&&SYSP,DISP=SHR
//SORTOUT DD  DSN=&&SYSPS,DISP=(NEW,PASS),STORCLAS=TEMP,
//             DISP=(NEW,CATLG,DELETE),
//             SPACE=(CYL,(200,100),RLSE),
//             DCB=*.SORTIN
//SYSIN   DD   *
 SORT FIELDS=(11,4,PD,A,7,4,BI,A)
/*
//STEP003  EXEC PGM=IFASMFDP,REGION=0M
//SYSPRINT DD   SYSOUT=*
//SMFIN    DD   DSN=your.archive.data.set,DISP=SHR
//SYSTOUT  DD   DSN=&&SYST,DISP=(NEW,PASS),STORCLAS=TEMP,
//              SPACE=(CYL,(200,100),RLSE),
//              DCB=*.SMFIN
//SYSIN    DD   *
  INDD(SMFIN,OPTIONS(DUMP))
  OUTDD(SYSTOUT)
  SID(SYST)
/*
//STEP004  EXEC PGM=SORT,REGION=6M
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//STEPLIB  DD  DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01 DD  SPACE=(CYL,(300,50)),UNIT=SYSDA
//SORTWK02 DD  SPACE=(CYL,(300,50)),UNIT=SYSDA
//SORTWK03 DD  SPACE=(CYL,(300,50)),UNIT=SYSDA
//SORTWK04 DD  SPACE=(CYL,(300,50)),UNIT=SYSDA
//SORTIN  DD  DSN=&&SYST,DISP=SHR
//SORTOUT DD  DSN=&&SYSTS,DISP=(NEW,PASS),STORCLAS=TEMP,
//             DISP=(NEW,CATLG,DELETE),
//             SPACE=(CYL,(200,100),RLSE),
//             DCB=*.SORTIN
//SYSIN   DD   *
  SORT FIELDS=(11,4,PD,A,7,4,BI,A)
/*
//*
//STEP005  EXEC PGM=SMFD,REGION=0M
//STEPLIB  DD   DSN=CAI.CASFLOAD,DISP=SHR
//SCDS     DD   DSN=CAI.SMFD.SCDS,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSXDIAG DD   SYSOUT=*
//SMFIN    DD   DSN=&&SYSPS,DISP=SHR
//SYSIN    DD   *
 SOURCE DUMP(SMFIN).
 DUMP.
//*
//STEP006  EXEC PGM=SMFD,REGION=0M
//STEPLIB  DD   DSN=CAI.CASFLOAD,DISP=SHR
//SCDS     DD   DSN=CAI.SMFD.SCDS,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSXDIAG DD   SYSOUT=*
//SMFIN    DD   DSN=&&SYSTS,DISP=SHR
//SYSIN    DD   *
 SOURCE DUMP(SMFIN).
 DUMP.
//
```

Note that this example assumes the following:

- The history files are being written to tape or to a virtual tape subsystem (VTS).
- The DYNAM option is set so that the history files will be allocated dynamically by CA SMF Director.
- The original recording method for the SMF data was to a MAN file. If the SMF data was originally from a logstream file, it can be imported and archived with the correct logstream using the LSNAME and SID operands on the SOURCE statement.

Once the older SMF data archive has been placed under the control of CA SMF Director, the old archive can be deleted or migrated offsite. There is no need to keep the original archive.

Attributes of History Files

The SMF files are emptied into standard OS data sets. These are CA SMF Director's history data sets, which are available for selective processing (extracting and printing).

This product automatically generates the names of these history data sets, since they are created as needed. This assures that all collected SMF data is uniquely identified.

You can have a measure of control over the naming of history data sets by specifying a standard prefix or high level qualifier for them. This is done with the PREFIX and APREFIX keywords of the DUMPOPTIONS command.

```
DUMPOPTIONS PREFIX(SYS1.SMFDATA).
```

The default prefix is SYS1.SMFDATA. The prefix can be single or multilevel, and is initially set at installation. If no value is specified for APREFIX, the name used is taken from the PREFIX operand.

For further details regarding the above control statement see Chapter 4, Command Reference.

Note: We recommend that a unique prefix be set for different configurations.

The structure of a history file data set name is as follows:

- For SYSx.MANx file dumps:

  ```
  prefix.Mfmmddyy.Thhmmtnn
  ```

- For logstream dumps:

  ```
  prefix.iimmddyy.Thhmmtnn
  ```

- For substream dumps:

  ```
  prefix.ifmmddyy.jhhmmtnn
  ```

where:

prefix

Represents the standard data set name prefix that this product automatically supplies whenever it dumps an SMF file.

Default: SYS1.SMFDATA.

M

Represents a SYSx.MANx history file.

f

Represents the suffix for the specific SMF file name (0-9, A-L). A value of 'S' indicates that previously dumped data from the SYSx.MANx files was manually added to CA SMF Director control.

mmddyy

Identifies the date this product dumped the SMF data.

T

Represents the time prefix.

hhmm

Specifies the time the SMF data was dumped (twenty-four hour clock).

t

Represents the history file pool type where the SMF data is located. Values are:

P

Primary or main history file. (Default.)

A

Alternate (duplex data set)

nn

Represents a sequence number that starts at one and is incremented by one each time an EOV occurs for a single dumped SMF MAN file.

i

Represents the first character of a unique substream identifier for a history file. The identifiers are generated when the logstream and substream definitions are added at COMPILE time, and can be seen in a LISTC report or in the ISPF online interface.

j

Represents the second character of a unique substream identifier for a history file. The identifiers are generated when the logstream and substream definitions are added at COMPILE time, and can be seen in a LISTC report or in the ISPF online interface.

The following is a sample:

```
SYS1.SMFDATA.MS090108.T1418P01
```

CA SMF Director history files are created with a logical record length of 32760, and can contain variable length spanned records. The default block size used depends on the device type used for the history files. For DASD devices the default block size is 4096, and for a magnetic tape the default is 32748. If a block size or logical record length other than the default is desired, it can be specified on the HISTORY1, HISTORY2, HISTij1, or HISTij2 DD JCL statements.

Chapter 3: Selective Processing of SMF Data

This chapter explains how to access the SMF data managed by CA SMF Director for billing/chargeback, performance measurement, job tracking/scheduling, and other processing.

This product allows you to EXTRACT or PRINT the SMF records contained in the history file data sets with minimal knowledge of where these history file data sets exist. Using the EXTRACT and/or PRINT control statements, you may select SMF records for further processing by SMF ID (SID), record type, or date/time range. You may also limit the number of SMF records placed in the output file by count, or call your own user exit to determine by some other criteria which SMF records are placed in the output file.

The remainder of this chapter illustrates how to specify the JCL and control statements to extract or print SMF records in various ways.

Creating Extract Files (CASFXSMF)

You may retrieve SMF data from history files by specifying the EXTRACT or PRINT commands with the CASFXSMF procedure, shown later in this chapter. The operands for these commands specify the type, quantity, and time period of the records to be retrieved.

In addition, the EXTRACT command has operands specifying:

- where the records are to be written
- whether or not SMF type 110 records that are compressed should be expanded in the output SMF file
- the name of an exit that must be called to process them before they are written

In a single run, you may code multiple EXTRACT/PRINT statements. Each EXTRACT statement normally creates a separate output file, although with a user exit, more than one output file can be created by a single EXTRACT statement.

A single SMF record may be routed to a number of destinations; however, it will never be written to the same destination more than once.

The EXTRACT control statement can also be used to rebuild split files in the event that a split file was not generated properly during a dump process. To facilitate application processing that uses the split file index, the EXTRACT process can also update a split file index. Because of this, applications can run unchanged whether the SMF data is in a split file or in an extracted file.

When SMF data is coming from history volumes that contain more than one SMF ID (SID), logstreams, or substreams:

- the SIDs are processed one at a time
- logstreams or substreams, if in use, are processed one at a time within a selected SID
- records within a SID appear in the output in chronological order if logstreams and substreams are not in use
- records within each logstream or substream appear in the output in chronological order within the logstream or substream
- if you use the SOURCE command, CA SMF Director outputs all data in input sequence

The following illustrates the JCL procedure CASFXSMF:

```
//   JOB (ACCTINFO),PGMR,CLASS=A,MSGCLASS=A
//*
//*
//CASFXSMF PROC A='*',                   /* SYSOUT CLASS
//         SCDDSN='CAI.JSMF.SCDS',       /* SCDS DSNAME
//         STEPLIB='CASFLOAD',           /* LOAD LIBRARY DSNAME
//         PRIUNIT=DUMMY,                /* PRIMARY POOL DEVICE
//         ALTUNIT=DUMMY                 /* ALTERNATE POOL DEVICE
//*******************************************************************
//* PROCEDURE CASFXSMF - CA SMF Director DATA EXTRACT PROCEDURE
//*                      USED TO EXTRACT SMF DATA.
//*
//* NOTES:
//* (1) THIS PROCEDURE HAS BEEN SETUP WITH THE DYNAM OPTION OF
//*     CA SMF Director ENABLED,  WHICH  IS RECOMMENDED.  IF  THIS
//*     OPTION  IS NOT  USED,  CHANGE  PRIUNIT AND/OR ALTUNIT
//*     TO 'UNIT=(XXXX,,DEFER)' WHERE XXXX IS THE UNIT
//*     OR  DEVICE  NAME  (E.G. SYSDA, DISK, TAPE, CART).  FOR
//*     FURTHER DETAILS,  REFER TO THE CA SMF DIRECTOR SYSTEMS
//*     PROGRAMMER GUIDE OR THE USER GUIDE
//*
```

```
//* NOTE: AFTER MODIFICATION, THE PROC CAN BE COPIED INTO A
//*       PROCLIB OF YOUR CHOICE, USED INSTREAM, OR THE CAI
//*       COMMON PROCEDURE LIBRARY CAN BE ADDED TO THE SYSTEM
//*       PROCLIB CONCATENATIONS.
//*
//* REFERENCE: DSN=CASFPROC(CASFXSMF)
//******************************************************************
//EXTRACT EXEC PGM=SMFD,REGION=2048K
//STEPLIB  DD  DSN=&STEPLIB.,DISP=SHR
//SYSPRINT DD  SYSOUT=&A.
//SYSXDIAG DD  SYSOUT=&A.
//SCDS     DD  DISP=SHR,DSN=&SCDDSN.
//HISTORY1 DD  &PRIUNIT.,DISP=OLD
//HISTORY2 DD  &ALTUNIT.,DISP=OLD
//*
//    PEND
//*
```

EXTRACT One Day's Data

The following example extracts all SMF data on a given day for an SMF ID (SID).

```
//jobname  JOB  job accounting parameters
//STEP0001 EXEC CASFXSMF,A='A',PARM.STEP1='READ',  (3)
//         STEPLIB='CASFLOAD',
//         SCDDSN='CAI.MSMF.SCDS'
//EXTRACT  DD   DISP=OLD,                 (1)
//         DSN=CAI.MSMF.EXTRACT
//SYSIN    DD   *
EXTRACT DAY(012508) SID(SID1).            (2)
/*
```

Note:

1. Since the EXTRACT command does not specify the TODD name, the ddname EXTRACT automatically becomes the default. This is where extracted records will be written.
2. Extract all SMF data for SID1 on January 25, 2008. If this job ran on January 26, 2008, then EXTRACT SID(SID1) YESTERDAY could have been used.
3. For read-only execution, replace the EXEC statement with this statement containing the PARM to specify read-only access of the SCDS file.

See Chapter 4 of this guide for a complete description of the EXTRACT command and its operands.

EXTRACT One Week's Data

The following example illustrates how to extract SMF data on a weekly basis for a specific SMF ID (SID).

```
//jobname  JOB  accounting information
//STEP001  EXEC CASFXSMF,A='*'
//SMFOUT   DD  DISP=(,CATLG),UNIT=(CART,,DEFER),DSN=CAI.PMA.WEEKLY(+1)    (1)
//         DCB=(LRECL=32760,RECFM=VBS,BLKSIZE=32748)
//SYSIN    DD  *
   EXTRACT  SID(XAT1)  LWEEK  TODD(SMFOUT)   (2)
/*
//
```

Note:

1. SMF records for system XAT1 will be extracted and sent to the output file SMFOUT.
2. The LWEEK operand indicates the date and time range for the SMF records that will go to the SMFOUT file. The SMF records written to the SMFOUT DD will be for seven days starting from the first day of the previous week. This product assumes that a week starts on a Sunday and ends on a Saturday.

See Chapter 4 of this guide for a complete description of the EXTRACT command and its operands.

EXTRACT One Month's Data

The following example performs two extractions of designated SMF data over a period of one month.

```
//jobname  JOB  job accounting parameters
//STEP0001 EXEC CASFXSMF,A='A',          /* SYSOUT CLASS
//         STEPLIB='CASFLOAD',
//         SCDDSN='CAI.MSMF.SCDS'
//JOBRECS  DD   DISP=(NEW,CATLG,DELETE),
//         DSN=CAI.MSMF.JOBRECS,
//         UNIT=SYSDA,
//         SPACE=(CYL,(50,20),RLSE),
//         DCB=(RECFM=VB,BLKSIZE=13030,LRECL=13022)
//TSORECS  DD   DISP=(NEW,CATLG,DELETE),
//         DSN=CAI.MSMF.TSORECS,
//         UNIT=SYSDA,
//         SPACE=(CYL,(50,20),RLSE),
//         DCB=(RECFM=VB,BLKSIZE=13030,LRECL=13022)
//SYSIN    DD   *
EXTRACT MONTH(0208) SID(SYS1) SELECT(4:6 20 26) TODD(JOBRECS).  (1)
EXTRACT MONTH(0208) SID(SYS1) SELECT(34 35) TODD(TSORECS).     (2,3)
/*
```

Note:

1. For system SYS1, extract all February 2008 occurrences of SMF record types 4, 5, 6, 20, and 26. Write them to the data set defined by the JOBRECS DD statement, where they are available for further analysis and processing.
2. Extract SMF record types 34 and 35 for system SYS1 during February 2008. Write these records to ddname TSORECS.
3. If this job ran on any day in March 2008, the EXTRACT command: EXTRACT LMONTH SID(SYS1) SELECT (34 35) TODD (TSORECS) would perform the same function.

See Chapter 4 of this guide for a complete description of the EXTRACT command and its operands.

EXTRACT Using a User Exit

The following example illustrates the use of a user exit written to separate SMF records by JOB name, and place them in separate output files. The user exit CASFXXIT contains logic to count the number of SMF type 14 and 15 records and write every 8th record to the output file.

```
//jobname  JOB  accounting information
//*
//STEP001 EXEC CASFXSMF,A='*'
//SMF01   DD DISP=(,CATLG),UNIT=SYSDA,DSN=CAIMICS.JOB01,
//        SPACE=(CYL,(10,10),RLSE),
//        DCB=(LRECL=32760,RECFM=VBS,BLKSIZE=32748)
//SYSIN DD  *
    EXTRACT SID(ALL) YESTERDAY EXIT(CASFXXIT) TODD(SMF01)
```

See Chapter 4 of this guide for a complete description of the EXTRACT command and its operands.

PRINT SMF Records

The PRINT command is processed like the EXTRACT command and also uses the CASFXSMF procedure. All selected records are written to SYSPRINT.

The following example requests a printout of up to 15 SMF records, types 4 through 6 and 20 and 26, for system SID1. The date of the SMF dump is January 18, 2008, and the time period to search for these records is 4 to 5 a.m. Records with timestamps exactly equal to 4 a.m. will be selected if they match any of the selected types; records with timestamps exactly equal to 5 a.m. will be excluded.

```
//jobname  JOB  job accounting parameters
//STEP0001 EXEC CASFXSMF,A='A',           /* SYSOUT CLASS
//         STEPLIB='CASFLOAD',
//         SCDDSN='CAI.MSMF.SCDS'
PRINT   FROM(011808 0400) TO(011808 0500) SID(SID1)
        SELECT (4:6 20 26) COUNT(15).
/*
```

See Chapter 4 of this guide for a complete description of the PRINT command and its operands.

Chapter 4: SMFD Control Statement Reference

This chapter contains detailed descriptions of all CA SMF Director control statements and their operands. Control statements are listed in alphabetical order. Within each control statement, required operands are listed first, followed by optional operands.

The following control statements allow you to conduct inventory functions and do selective processing on SMF data. Use the control statements to perform these functions:

BEGIN

Start a series of configuration control statements.

COMPILE

Compile the configuration definition.

END

End the configuration definition.

DUMPOPTIONS

Define specifications for SMF history files for each system configuration and define any logstreams or substreams that will contain data that CA SMF Director is to manage.

STREAMOPTIONS

Set overrides for the DUMPOPTIONS for specific logstreams and substreams.

OPTIONS

Define specifications for all systems.

DUMPTAPES

Modify the pool of tapes available for dumped SMF data.

TAPEINIT

Switch the current dump tape.

DUMP

Empty the contents of one or more SMF files.

SPLIT

Write selected SMF records to additional files so they are available for processing by other applications without an intermediate EXTRACT.

EXTRACT

Copy SMF data to be processed by another application or product.

PRINT

Print SMF data.

ADDX

Add index entries of SMF data.

DELETEX

Remove one index entry of SMF data at a time.

UPDTX

Modify a range of index entries.

LISTC

List configuration information for one or more SIDs.

LISTH

List index information for SMF history files.

SOURCE

Redefine the input source of data to be used in processing by CA SMF Director.

CHECKIT

Diagnostic tool used to display SCDS records in dump format.

Specifying Control Statements

The following illustrates how the various commands and control statements can be specified to CA SMF Director. See member CASFCTL1 in the CASFJCL library, loaded during the installation process.

```
//CASFCTL1  JOB...
//STEP1     EXEC  PGM=SMFD,REGION=0M
//STEPLIB    DD   DISP=SHR,DSN=CASFLOAD
//SCDS       DD   DISP=SHR,DSN=CORP.SMFD.SCDS
//SYSPRINT   DD   SYSOUT=A
//SYSXDIAG   DD   SYSOUT=A
//SYSIN      DD *
*
*...PLACE CONTROL STATEMENTS HERE...
*
/*
//
```

Control Statement Language Rules

- Statements are free-form. Control statements and operands need not start in a specific column. Operands may be specified in any order.
- Each statement must end with a "." (period). This permits you to write statements that exceed one card image (80 bytes) in length. There is no special continuation indicator.
- There must be at least one blank space between a control statement and its first operand. Also, we recommend at least one blank space between operands.
- Operands and their values can be written with or without an intervening blanks; for example, SID(sid1) or SID (sid1).
- Values for operands are enclosed by parentheses. When you want to specify two or more values, separate them by a "," (comma), a blank, or both. We recommend that you use either commas or blanks consistently.
- When you are specifying a range of values, the lower and upper limits may be separated by a ":" (colon), to save coding time and space.
- Comments are specified by placing an "*" (asterisk) in column one. Blank cards are also treated as comments.
- CA SMF Director checks for sequence numbers only on the first control statement (in positions 73-80; 73-76 must be numeric). If sequence numbers are present in the first statement, it ignores positions 73-80 in all following statements.
- Date fields entered as operands to CA SMF Director control statements may be in one of two formats, Julian (cyyddd) or Gregorian (mmddyy). Some control statements accept either format. For Gregorian date formats, the two-digit year (yy) having a value of 00 through 59 represents the year 2000 through 2059, and 60 through 99 represents the years 1960 through 1999. 'mm' represents the two-digit month (01-12) and 'dd' represents the two-digit day of the month (01-31).

  For Julian date formats, the three-digit year (cyy) having a value of 060 through 099 represents the year 1960 through 1999, and values of 100 through 141 represent the year 2000 through 2041. The three-digit day (ddd) represents the day of the year relative to the first day of the year. For example, January 28, 2008 in Julian date format would be 108028.
- If the description of an operand indicates that an asterisk (*) is an acceptable value, you can use the asterisk to copy the value specified at a higher level, for example on the OPTIONS statement.

Note: Two special expiration dates are supported:

099000 indicates that the data set will not expire as long as it remains in the system catalog.

099366 indicates that the data set will never expire.

Reading Syntax Diagrams

Syntax diagrams are used to illustrate the format of statements and some basic language elements. Read syntax diagrams from left to right and top to bottom.

The following terminology, symbols, and concepts are used in syntax diagrams:

- Keywords appear in uppercase letters, for example, COMMAND or PARM. These words must be entered exactly as shown.
- Variables appear in italicized lowercase letters, for example, *variable*.
- Required keywords and variables appear on a main line.
- Optional keywords and variables appear below a main line.
- Default keywords are identified by a single arrowhead pointing to the left.
- Double arrowheads pointing to the right indicate the beginning of a statement.
- Double arrowheads pointing to each other indicate the end of a statement.
- Single arrowheads pointing to the right indicate a portion of a statement, or that the statement continues in another diagram.
- Punctuation marks or arithmetic symbols that are shown with a keyword or variable must be entered as part of the statement or command. Punctuation marks and arithmetic symbols can include the following:

,	Comma	>	greater than symbol
.	Period	<	less than symbol
(	open parenthesis	=	equal sign
)	close parenthesis	¬	not sign
+	addition	-	subtraction
*	Multiplication	/	division

The following is a diagram of a statement without parameters:

Statement Without Parameters

```
►►─ COMMAND ──────────────────────────────────────────────────►◄
```

For this statement, you must write the following:

```
COMMAND
```

Required parameters appear on the same horizontal line, the main path of the diagram, as the command or statement. The parameters must be separated by one or more blanks.

Statement with Required Parameters

```
▶▶─ COMMAND ─ PARM1 ─ PARM2 ──────────────────────────────────────────────▶◀
```

You must write the following:

```
COMMAND PARM1 PARM2
```

Delimiters, such as parentheses, around parameters or clauses must be included.

Delimiters Around Parameters

```
▶▶─ COMMAND ─ (PARM1) ─ PARM2='variable' ─────────────────────────────────▶◀
```

If the word *variable* is a valid entry, you must write the following:

```
COMMAND (PARM1) PARM2='variable'
```

When you see a vertical list of parameters as shown in the following example, you must choose one of the parameters. This indicates that one entry is required, and only one of the displayed parameters is allowed in the statement.

Choice of Required Parameters

```
▶▶─ COMMAND ─┬─ PARM1 ─┬──────────────────────────────────────────────────▶◀
             ├─ PARM2 ─┤
             └─ PARM3 ─┘
```

You can choose one of the parameters from the vertical list, such as in the following examples:

```
COMMAND PARM1
COMMAND PARM2
COMMAND PARM3
```

When a required parameter in a syntax diagram has a default value, the default value is identified by an arrowhead pointing to the left. The default value is the value automatically assigned to the parameter if the command is not specified. If you specify the command, you must code the parameter and specify one of the displayed values.

Default Value for a Required Parameter

```
▶▶─ COMMAND ─ PARM1= ─┬─ YES ◀ ─┬─ PARM2 ─────────────────────────────────▶◀
                      └─ NO ────┘
```

If you specify the command, you must write one of the following:

```
COMMAND PARM1=NO PARM2
COMMAND PARM1=YES PARM2
```

A single optional parameter appears below the horizontal line that marks the main path.

Optional Parameter

```
►►─ COMMAND ─┬───────────┬──────────────────────────────────────────►◄
             └─ PARAMETER ─┘
```

You can choose (or not) to use the optional parameter, as shown in the following examples:

```
COMMAND
COMMAND PARAMETER
```

If you have a choice of more than one optional parameter, the parameters appear in a vertical list below the main path.

Choice of Optional Parameters

```
►►─ COMMAND ─┬─────────┬────────────────────────────────────────────►◄
             ├─ PARM1 ─┤
             └─ PARM2 ─┘
```

You can choose any of the parameters from the vertical list, or you can write the statement without an optional parameter, such as in the following examples:

```
COMMAND
COMMAND PARM1
COMMAND PARM2
```

In some statements, you can specify a single parameter more than once. A repeat symbol indicates that you can specify multiple parameters.

Repeatable Variable Parameter

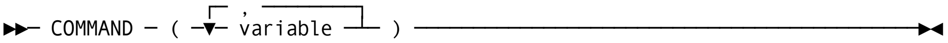
```
             ┌──────────────┐
►►─ COMMAND ─▼─ variable ─┴─────────────────────────────────────────►◄
```

In the preceding diagram, the word *variable* is in lowercase italics, indicating that it is a value you supply, but it is also on the main path, which means that you are required to specify at least one entry. The repeat symbol indicates that you can specify a parameter more than once. Assume that you have three values named VALUEX, VALUEY, and VALUEZ for the variable. The following are some of the statements you might write:

```
COMMAND VALUEX
COMMAND VALUEX VALUEY
COMMAND VALUEX VALUEX VALUEZ
```

If the repeat symbol contains punctuation such as a comma, you must separate multiple parameters with the punctuation. The following diagram includes the repeat symbol, a comma, and parentheses:

Separator with Repeatable Variable and Delimiter

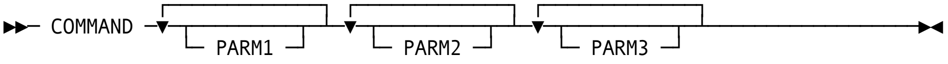


In the preceding diagram, the word *variable* is in lowercase italics, indicating that it is a value you supply. It is also on the main path, which means that you must specify at least one entry. The repeat symbol indicates that you can specify more than one variable and that you must separate the entries with commas. The parentheses indicate that the group of entries must be enclosed within parentheses. Assume that you have three values named VALUEA, VALUEB, and VALUEC for the variable. The following are some of the statements you can write:

```
COMMAND (VALUEC)
COMMAND (VALUEB,VALUEC)
COMMAND (VALUEB,VALUEA)
COMMAND (VALUEA,VALUEB,VALUEC)
```

The following diagram shows a list of parameters with the repeat symbol:

Optional Repeatable Parameters

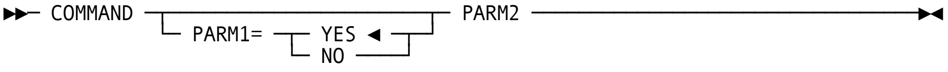


The following are some of the statements you can write:

```
COMMAND PARM1
COMMAND PARM1 PARM2 PARM3
COMMAND PARM1 PARM1 PARM3
```

The arrowhead pointing to YES in the following diagram indicates that it is the default value for the parameter. If you do not include the parameter when you write the statement, the result is the same as if you had actually specified the parameter with the default value.

Default Value for a Parameter



For this command, COMMAND PARM2 is the equivalent of COMMAND PARM1=YES PARM2.

In some syntax diagrams, a set of several parameters is represented by a single reference.

Variables Representing Several Parameters

```
►►─ COMMAND ─┬────────────────────┬──────────────────────────────►◄
             ├─ PARM1 ────────────┤
             └─┤ Parameter Block ├┘
```

Expansion of Parameter Block

```
├──┬──────────────────────┬──────────────────────────────────────────┤
   ├─ PARM2 ──────────────┤
   └─ PARM3 ──┬─────────┬─┘
              ├─ PARM4 ─┤
              └─ PARM5 ─┘
```

The Parameter Block can be displayed in a separate syntax diagram.

Choices you can make from this syntax diagram therefore include, but are not limited to, the following:

```
COMMAND PARM1
COMMAND PARM3
COMMAND PARM3 PARM4
```

Note: Before you can specify PARM4 or PARM5 in this command, you must specify PARM3.

ADDX

This control statement adds entries to the index of SMF data contained in the SCDS. ADDX allows you to create manual index entries so you can easily reference SMF data that was not dumped by CA SMF Director.

Statement Format:

ADDX

```
►►─ ADDX ─ SID(sid) ─ VOLUME(volser) ─ FILESQ(filesq) ─ DSNAME(dsn) ──────────►

►─┬────────────────────┬─┬─────────────────────┬─ DUMPD(date time) ─────────►
  ├─ LOW(date time) ───┤ ├─ HIGH(date time) ───┤
  └─ JLOW(date time) ──┘ └─ JHIGH(date time) ──┘

►─┬───────────────────────┬─┬─────────────┬─┬───────────────────────────┬─ . ──►◄
  └─ UNIT(device name) ───┘ └─ ALTERNATE ─┘ ├─ LSNAME(logstream name) ──┤
                                            └─ SUBNAME(substream name) ─┘
```

Operands and Descriptions:

SID(sid)

Specifies the SMF ID for the history-file index, and defines the origin of the SMF input data.

"sid" is the value defined in the BEGIN statement.

VOLUME(volser)

Specifies the serial number of the volume on which the indexed data set resides.

FILESQ(filesq)

Specifies the file sequence number of the indexed data set if it resides on tape. If the indexed data set resides on DASD, this value should be 0.

DSNAME(dsn)

Specifies the complete name of the data set indexed by this request.

LOW(date time)

Specifies the date and time of the first SMF record in the file being added to the data set index.

"date" is in Gregorian (mmddyy) format.

"time" is in hour/minute/second format (hhmmss).

Note: The year portion of the date (yy) must be on or after the year specified in the FIRSTYEAR option on the DUMPOPTIONS statement. If not, ADDX will fail.

JLOW(date time)

Specifies the date and time in Julian format of the first SMF record in the file being added to CA SMF Director inventory. "date" is in the format of (cyyddd) and "time" is in the format hhmmss. You must specify either the LOW or JLOW keyword but not both.

HIGH(date time)

Specifies the date and time of the last SMF record in the file being added to the data set index. "date" is in Gregorian (mmddyy) format.

"time" is in hour/minute/second (hhmmss) format.

JHIGH(date time)

Specifies the date and time in Julian format of the last SMF record in the file being added to CA SMF Director inventory. "date" is in the format of (cyyddd) and "time" is in the format of hhmmss. You must specify either the HIGH or JHIGH keyword but not both.

DUMPD(date time)

Specifies the date (mmddyy) and time (hhmmss) that the SMF data was created on the indicated volume. This date is used with the expiration date to determine when this entry should be removed from CA SMF Director inventory if the AUTODEL option is enabled.

UNIT (device name)

Specifies the esoteric name used to store the history file data. This can be 3490, 3480, and so on. It can also be an SMF storage class.

ALTERNATE

Specifies that the file being indexed is the duplexed history file for this SMF ID.

LSNAME(logstream name)

Specifies the logstream name for the history file index that this new entry will be added to. The logstream must be defined to the specified SID. The LSNAME and SUBNAME operands are mutually exclusive.

SUBNAME(substream name)

Specifies the substream name for the history file index that this new entry will be added to. The substream must be defined to the specified SID. The LSNAME and SUBNAME operands are mutually exclusive.

Sample ADDX Statement:

```
ADDX SID(XE44) LOW(052198 010010) HIGH(052198 215900)
  VOLUME(115193) FILESQ(1)  UNIT(3490)
      DSNAME(SYSTEM.USER.SMFDATA.D052198) DUMPD(052298 000500)  .
```

ADDX statement output:

```
CA                                                AUDIT TRAIL REPORT                           SMFD       PAGE     1
CA SMF Director     12.6                                                             SYSA      23 JUL yyyy  15:27:18


  ADDX SID(XE44) LOW(052198 010010) HIGH(052198 215900)
    VOLUME(115193) FILESQ(1)  UNIT(3490)
        DSNAME(SYSTEM.USER.SMFDATA.D052198) DUMPD(052298 000500)  .

CAF3602I  DATASET INDEX UPDATE PERFORMED

CAF2900I  RUN COMPLETE
```

BEGIN

This control statement signals the start of a series of statements defining a configuration. BEGIN appears in:

- the input stream after a COMPILE control statements.
- a data set referenced by a SOURCE operand in a COMPILE statement.

Each BEGIN must have a corresponding END to conclude the definition of that particular configuration. Each configuration definition remains in the SCDS file. It can later be replaced but not deleted.

Statement Format:

BEGIN

```
▶▶─ BEGIN ─ CONFIG(n) ─ SID(sid) ─ NAME(name) ─ . ────────────────────────────▶◀
```

Operands and Descriptions:

CONFIG(n)

Identifies the configuration that is defined in the statements that follow. This is a reference number and is not actually used by the SMFD program.

(n) is a positive integer no greater than 255. It is unique to this configuration definition.

SID(sid)

Identifies the SMF ID for this configuration. The SID is the 4-character identifier that appears in the SMF data. If this SID already exists in the SCDS, the configuration definition is completely replaced.

NAME(name)

Specifies a comment of up to 20 characters long within quotes, which is printed on the heading portion of CA SMF Director reports for this configuration definition.

Sample BEGIN Statement:

```
COMPILE CONFIGURATION STORE.
    BEGIN CONFIG(22) SID(XE22) NAME('C.A. DEMO SYSTEM-2') .
     DUMPOPTIONS COPIES(2) CATLG SCRATCH NOVERIFY
       RETPD(5) PREFIX(SYS3.SMFDATA)
      ARETPD(10) .
    END CONFIG(22) .
```

BEGIN statement output:

```
 CA                                                AUDIT TRAIL REPORT                                 SMFD       PAGE     1
CA SMF Director     12.6                                                                      SYSA      23 JUL yyyy  15:34:10

   COMPILE CONFIGURATION STORE.
       BEGIN CONFIG(22) SID(XE22) NAME('C.A. DEMO SYSTEM-2') .
        DUMPOPTIONS COPIES(2) CATLG SCRATCH NOVERIFY
          RETPD(5) PREFIX(SYS3.SMFDATA)
         ARETPD(10) .
       END CONFIG(22) .

 CAF1301I  CONFIGURATION DEFINITION XE22 STORED

 CAF2900I  RUN COMPLETE
```

CHECKIT

Note: Use this control statement only when requested by CA Technical Support.

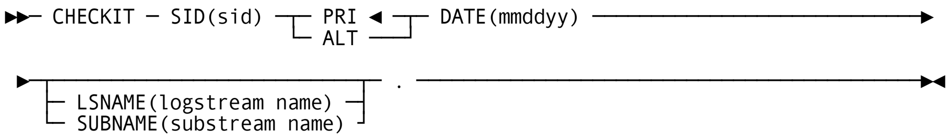
This control statement provides diagnostic displays of the record chain used for an EXTRACT control statement. The diagnostic displays provide CA Technical Support with the information needed to see if there are key or chain errors when compared with a LISTH printout.

The CHECKIT control statement should be run when any of the following situations occur:

1. LISTH printout shows data out of sequence.
2. EXTRACT run does not contain all of the SMF data for a period of time although the proper operands **(from** time and **to** time) were used.

Statement Format:

CHECKIT



Operands and Descriptions:

SID(sid)

Specifies the SMF ID being checked.

PRIMARY

Specifies that index entries defining primary history files are the ones being checked.

ALTERNATE

Specifies that index entries defining alternate history files are the ones being checked.

DATE(mmddyy)

Specifies the day being checked in Gregorian date format.

LSNAME(logstream name)

Checks index entries in the specified logstream. If the name is omitted, the check will be performed on dumps that are not associated with a logstream. The LSNAME and SUBNAME operands are mutually exclusive.

SUBNAME(substream name)

Checks index entries in the specified substream. If the name is omitted, the check will be performed on dumps that are not associated with a substream. The SUBNAME and LSNAME operands are mutually exclusive.

Sample CHECKIT Statement:

```
CHECKIT SID(XE44) PRI DATE (041698).
```

CHECKIT statement output:

```
CA                                              AUDIT TRAIL REPORT                           SMFD       PAGE     1
 CA SMF Director     12.6                                                                      21 APR yyyy  11:22:52

  CHECKIT  SID(XE44) DATE(041698) PRI.

CA                                              AUDIT TRAIL REPORT                           SMFD       PAGE     2
 CA SMF Director     12.6                                                                      21 APR yyyy  11:22:52

                              PRIMARY INDEX CHECK FOR SYSID  XE44  DATE:   98.106

          1 HISTORY FILE INDEX ENTRIES REFERENCING  15,809 SMF RECORDS CHECKED, WITH  0 ERRORS FOUND

 SYSTEM CONTROL BLOCK
         0000    C3E3D3D9 00000000 0098107F 00507C42   0000000C 00000000 00000000 00000000    *CTLR............................*
         0020    00000000 00000000 00000030 00000001   00000020 00000005 00000000 00000000    *................................*
         0040    00000000 00000000 00000000 00000000   ...                                    *................................*

 CONFIGURATION CONTROL DATA FOR SID  XE44
         0000    00000003 E7C5F4F4 E2E8E2E3 C5D440E4   E2C9D3C4 C1D4E840 40404040 0097353F    *....XE44SYSTEM USILDAMY     ....*
         0020    000000B7 70000000 00000000 FFFF0030   ...                                    *................................*

         0000    E7C5F4F4 0000000F 00000000 0000000F   00000000 00000000 00000000 00000000    *XE44............................*
         0020    00000000 00000000 00000000 00000000   ...                                    *................................*

 HISTORY FILE INDEX BLOCK(S)
         0000    C4D7C9E7 0000000D 00000000 00000000   E7C5F4F4 00620000 00000000 00000000    *DPIX............XE44............*
         0020    00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000    *................................*
         0120    00000000 00000000 00000C00 00000000   00000000 00000000 00000000 00000000    *................................*
         0140    00000000 00000000 00000000 00000000   00000000 00000000 00000000 000C0000    *................................*
         0160    00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000    *................................*
         0FE0    00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000    *................................*

         0000    C4E2C9D9 00000006 00000000 0000000C   0097344F 003535AC 0097344F 0039AD73    *DSIR............................*
         0020    0F550087 0097344F 003535AC 0097344F   0039AD73 0098106F 005440B6 D4E5E2E7    *.......................... .MVSX*
         0040    C5F9F64B E2C6F1F6 4BD4E2F0 F4F1F6F9   F84BE3F1 F5F2F0C9 F1404040 40404040    *E96.SF16.MS041698.T1520I1       *
         0060    40404040 40404040 F4F8F8F2 F2F40001   40404040 4040F3F4 F8F0E740 40400098    *        488224..      3480X   ..*
         0080    111CC000 000041CC 00036522 8060216E   E3F28068 0E021A00 84000000 00000014    *................T2..............*
         00A0    00000000 0000008B 00000000 00000000   00000000 00000000 00000000 00000000    *................................*
         00C0    00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000    *................................*
         0FE0    00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000    *................................*

         0000    C4E2C9D9 0000000C 00000006 00000000   0098089F 0042CD6C 0098106F 0058CE40    *DSIR........................... *
         0020    0E470195 0098089F 0042CD6C 0098089F   00439BB5 0098092F 0050D791 D4E5E2E7    *..........................P.MVSX*
         0040    C5F9F64B E2C6F1F6 4BD4F1F0 F4F0F2F9   F84BE3F1 F4F4F3C9 F1404040 40404040    *E96.SF16.M1040298.T1443I1       *
         01C0    00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000    *................................*
         0FE0    00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000    *................................*

CAF2900I  RUN COMPLETE
```

Record dumps in hex format are provided for all records that could contain indexes for the date specified.

COMPILE

This control statement initiates the compiling and optional storing on the SCDS of a configuration's definition at installation and whenever a configuration changes. COMPILE is followed immediately by the BEGIN control statements; the entire configuration definition must be concluded by an END control statements.

Statement Format:

COMPILE

```
►►─ COMPILE ─ CONFIGURATION ─┬──────────────────┬─┬─────────┬─ . ──────────────►◄
                             └─ SOURCE(ddname) ─┘ └─ STORE ─┘
```

Operands and Descriptions:

CONFIGURATION

Instructs the system to compile the definition of a configuration.

SOURCE(ddname)

Indicates that the statements defining this configuration are not in the input stream but, rather, in a file specified by "(ddname)".

"(ddname)" may consist of a 1 to 8-character ddname for which there is a DD statement in the execution JCL for the SMFD program.

STORE

Instructs the system to place the compiled configuration definition in the SCDS file. The definition can be brand-new or a replacement. A replacement must redefine all DUMPOPTIONS or STREAMOPTIONS characteristics.

If you do not specify STORE, the definition is temporary. It is no longer active after this execution of SMFD.

If the SCDS is to be shared by two or more LPARs, review the system requirements for sharing a SCDS in the *Installation Guide*.

Sample COMPILE Statement:

```
COMPILE CONFIGURATION STORE.
    BEGIN CONFIG(22) SID(XE22) NAME('C.A. DEMO SYSTEM-2') .
     DUMPOPTIONS COPIES(2) CATLG SCRATCH NOVERIFY
       RETPD(5) PREFIX(SYS3.SMFDATA)
      ARETPD(10) .
    END CONFIG(22) .
```

COMPILE statement output:

```
CA                                              AUDIT TRAIL REPORT                        SMFD       PAGE     1
CA SMF Director     12.6                                                         SYSA      23 JUL yyyy  15:39:41

  COMPILE CONFIGURATION STORE.
      BEGIN CONFIG(22) SID(XE22) NAME('C.A. DEMO SYSTEM-2') .
       DUMPOPTIONS COPIES(2) CATLG SCRATCH NOVERIFY
         RETPD(5) PREFIX(SYS3.SMFDATA)
        ARETPD(10) .
      END CONFIG(22) .

CAF1301I  CONFIGURATION DEFINITION XE22 REPLACED

CAF2900I  RUN COMPLETE
```

DELETEX

This control statement removes entries from the index of SMF data that is maintained on the SCDS. DELETEX is useful for removing old or duplicate entries for overlapping data. See the *Systems Programmer Guide* for more details.

Important! We highly recommend that the SCDS be backed up before execution of any DELETEX command, especially with the ALL parameter coded.

Statement Format:

DELETEX

```
▶▶─ DELETEX ─ SID(sid) ─ LOW(date time) ─ HIGH(date time) ─────────────────────▶

 ▶─┬─────────────┬─┬───────┬─┬──────────────────────────┬─ . ──────────────────▶◀
   └─ ALTERNATE ─┘ └─ ALL ─┘ ├─ LSNAME(logstream name) ─┤
                             └─ SUBNAME(substream name) ┘
```

Operands and Descriptions:

SID(sid)

Specifies the SMF ID for the history-file index. This is the 4-character identifier that appears in the SMF data.

LOW(date time)

Specifies the date and time of the first SMF record in the entry being deleted from the data set index.

"date" - Gregorian (mmddyy) format.

"time" is in hour/minute/second format (hhmmss).

HIGH(date time)

Specifies the date and time of the last SMF record in the entry being deleted from the data set index. "date" is in Gregorian (mmddyy) format. "time" is in hour/minute/second format (hhmmss).

ALTERNATE

Specifies that the index being deleted is the alternate history index for this SID. If not specified, the primary index is deleted.

ALL

Deletes all index entries within the date and time range specified on the LOW and HIGH operands. The LOW and HIGH operands must exactly equal, respectively, the low date/timestamp of a SCDS entry and the high date/timestamp of a SCDS entry; otherwise, nothing will be deleted. If ALL is not coded, the LOW and HIGH operands must exactly equal, respectively, the low and high timestamps of a single SCDS entry in order for that entry to be deleted.

LSNAME(logstream name)

Deletes entries in the specified logstream. If the name is omitted, the delete will be performed on dumps that are not associated with a logstream. The LSNAME and SUBNAME operands are mutually exclusive.

SUBNAME(substream name)

Deletes entries in the specified substream. If the name is omitted, the delete will be performed on dumps that are not associated with a substream. The LSNAME and SUBNAME operands are mutually exclusive.

Sample DELETEX Statement:

```
DELETEX SID(XE44) LOW(042998 103019) HIGH(042998 104444)
  ALTERNATE .
```

DELETEX statement output:

```
 CA                                              AUDIT TRAIL REPORT                          SMFD       PAGE     1
CA SMF Director     12.6                                                             SYSA      23 JUL yyyy  15:42:00

   DELETEX SID(XE44) LOW(042998 103019) HIGH(042998 104444)
     ALTERNATE .

 CAF3602I  DATASET INDEX UPDATE PERFORMED

 CAF2900I  RUN COMPLETE
```

DUMP

This control statement copies SMF data from z/OS data sets or an alternate input source, to CA SMF Director history files, and creates an index entry to the SMF data for future reference. When the SMF data is from the z/OS SMF data sets, this product automatically clears these data sets for reuse after successfully copying the SMF data. This control statement may be used instead of specifying PARM=DUMP on the EXEC JCL statement invoking CA SMF Director.

Statement Format:

DUMP

```
▶▶─ DUMP ─ . ──────────────────────────────────────────────────▶◀
```

Sample DUMP Statement:

```
SOURCE DUMP(SMFDATA).
  DUMP.
```

The above example will copy SMF data from a sequential data set to CA SMF Director history files.

DUMP statement output:

```
CA                                              AUDIT TRAIL REPORT                          SMFD
CA SMF Director     12.6                                                            SYSA      20 MAY yyyy  15:42:05

 SOURCE DUMP(SMFDATA).

CAF2800I  SOURCE OPTION SET

 DUMP.

CAF2207I  SMF DATA WILL BE READ FROM FILE:  SMFDATA  DSN: PMA.DEVLTEST.SF16.SMFDATA1 ON VOLUME PMA003
CAF2218I  SMF DATA USING CONFIGURATION XE44 IS BEING PROCESSED
CAF2401I  PRIMARY DUMP IS TO  DISK VOLUME: PMA003  FILE: 0  DSN: PMA.DEVLTEST.MS052008.T1542I19
CAF2406I  XE44 SMF DATA WAS FROM 12/10/1997 AT  9:41:11 TO 12/10/1997 AT 10:29:59
CAF2406I  ALL SMF DATA WAS FROM 12/10/1997 AT  9:41:11 TO 12/10/1997 AT 10:29:59

                                                 SUMMARY OF RECORDS COPIED

            TYPE   AVGL   MAXL      TOTAL   ............................... SUBTYPE COUNTS ...............................
               14    321    848        2,825
               15    301    326        1,985
               17    100    100           24
               18    144    144            2
               21     72     72            4
               23    134    134            1
               26    404    413           34
               30   1290  32732          896      1 -        70       2 -       148       3 -       304       4 -       303
                                                5 -        58       6 -        13
               32    282    332           51      1 -        34       2 -        17
               41    159    172            6      1 -         1       2 -         2
               42    383  15172        4,429      1 -         1       2 -         8       5 -         2       6 -     4,418
               50    114    114           13
               55     58     58            1
               57    116    116            1
               58     42     42            1
               60    498    663          129
               61    482   5832           33
               62    176    176           61
               64    459    524          103
               65    306    314           23
               66    304    304            7
               70   1396   1396            3      1 -         3
               71   1308   1308            3      1 -         3
               72    517   1040           75      1 -        75
               73   5368   5368            3      1 -         3
               74  16976  32744           75      1 -        24       5 -        51
               75    264    264            9      1 -         9
               78   6190  11240            6      2 -         3       3 -         3
               80    256    256        3,690
               89   1441   2140            3      1 -         2       2 -         1
               90    164    164            1
               92    212    212            1     11 -         1
              100   1111   1794           16
              101    830    830           10
              102    850   1470            6
              110    609   1178            3      2 -         3
              115    553    660            6
              116    420    420          148
              118    128    128           46    118 -        46
              128    163    163          713
              133    239    840           69
              187    117    685          258
              189    856   2680            8      1 -         2       3 -         2       5 -         2    1025 -         2
              248    156    156            1
              252     87     87            1
              254    164    196          963  54465 -       860   54467 -       103
              255    433   1024           98      1 -        41       4 -        37       7 -         8      17 -         2
                                               23 -         8      25 -         2
              ALL    436  32744       16,844
```

The following list gives an explanation of the fields contained in the Summary of Records Copied section of the report:

TYPE

SMF record type (numeric 0-255).

AVGL

Average record length of records for this type.

MAXL

Maximum record length of records for this type.

TOTAL

Number of records for this type successfully written.

SUBTYPE COUNTS

A list of SMF record subtypes and how many records for each subtype were successfully written. The first number shown is the numeric SMF record subtype, followed by the number of records. Counts for records with no subtype (zero) are not shown on the report. However, they are included in the total number of records.

ALL

Indicates the grand total line for all SMF record types.

DUMPOPTIONS

This control statement allows systems personnel to define the specifications for dumping SMF records to history files managed by CA SMF Director. The DUMPOPTIONS control statement must follow the BEGIN control statement defining the dump options for the specified system configuration, and precede the END CONFIG control statement.

Statement Format:

DUMPOPTIONS

```
►►─ DUMPOPTIONS ─ COPIES( ─┬─ 1 ◄ ─┬─ ) ─┬──────────────┬─┬──────────────┬──────────►
                           ├─ 2 ───┤     ├─ NOCATLG ◄ ─┤ ├─ SCRATCH ◄ ─┤
                           └─ * ───┘     ├─ CATLG ─────┤ ├─ NOSCRATCH ─┤
                                         └─ UPCATLG ───┘ └─ UPSCRATCH ─┘

►─┬──────────────┬────────────────────────────────────────────────────────────────►
  ├─ NOVERIFY ◄ ─┤
  └─ VERIFY ─────┘

►─┬────────────────────────────┬──────────────────────────────────────────────────►
  ├─ PREFIX(─┬──*─────────┬─)──┤
  │          └ prefix ────┘    │
  └─ NOPREFIX ◄ ───────────────┘

►─┬──────────────────────────────┬────────────────────────────────────────────────►
  ├─ APREFIX(─┬──*─────────┬─)───┤
  │           └ prefix ────┘     │
  └─ NOAPREFIX ──────────────────┘

►─┬──────────────────────────┬────────────────────────────────────────────────────►
  ├─ RETPD(───┬─0 ◄ ──┬─)─┤
  │           ├ nnnn ─┤   │
  └           └─ * ───┘  ─┘
  ├─ EXPDT(───┬─0 ◄ ──┬─)─┤
  │           ├ yyyddd┤   │
  └           └─ * ───┘  ─┘

►─┬──────────────────────────┬────────────────────────────────────────────────────►
  ├─ ARETPD(─┬──*────┬─)─────┤
  │          └ nnnn ─┘       │
  ├─ AEXPDT(─┬──*──────┬─)───┤
  └          └ yyyddd ─┘    ─┘

►─┬────────────────────────┬──────────────────────────────────────────────────────►
  └─ TRTCH(──┬──*─────┬─)──┘
             └ value ─┘

►─┬────────────────────────┬──────────────────────────────────────────────────────►
  └─ ATRTCH(─┬──*─────┬─)──┘
             └ value ─┘

►─┬──────────────────────────────────┬─────┬────────────────────┬─────────────────►
  ├─ LSNAME(lsname lsname ...)      ─┤     └─ FIRSTYEAR(yyyy) ─┘
  └─ SUBNAME(subname subname ...)   ─┘

►─┬──────────────────────────────────┬────────────────────────────────────────────►
  └─ MAXFILESONVOL( ─┬─ * ───┬─ )   ─┘
                     └─ n ───┘

►─┬────────────────────────────┬──────────────────────────────────────────────────►
  └─ DAYSTACK( ─┬─ NO ◄ ─┬─ ) ─┘
                ├─ YES ──┤
                └─ * ────┘

►─┬─────────────────────────────┬─┬──────────────────────────────┬────────────────►
  └─ PDSNLARGE( ─┬─ NO ◄ ─┬─) ─┘ └─ ADSNLARGE( ─┬─ NO ◄ ─┬─ ) ─┘
                 ├─ YES ──┤                      ├─ YES ──┤
                 └─ * ────┘                      └─ * ────┘
```

```
►─┬─────────────┬──────────────┬─────────────────────────────┬──────────────►
  ├─ NODYNAM ───┤              ├─ PDEVN( ─┬─ dname ─┬─ ) ───┤
  ├─ DYNAM ─────┤              │          ├─ NONE ──┤       │
  ├─ DYNAMVTS ──┤              │          └─ * ─────┘       │
  └─ UPDYNAM ◄ ─┘              └─ PSTORC( ─┬─ class ─┬─ ) ──┘
                                           ├─ NONE ──┤
                                           └─ * ─────┘

►─┬──────────────────────────────┬──────────────┬──────────────┬─── . ──►◄
  │ ADEVN( ─┬─ dname ─┬─ ) ──────┤              └─ SDAY(hhmm) ─┘
  │         ├─ NONE ──┤          │
  │         └─ * ─────┘          │
  └─ ASTORC( ─┬─ class ─┬─ ) ────┘
              ├─ NONE ──┤
              └─ * ─────┘
```

Operands and Descriptions:

COPIES(<u>1</u>|2|*)

Specifies whether or not CA SMF Director should create a duplicate copy of the dumped SMF data (duplexing). If 1 is specified, no duplexing is performed. If 2 is specified, duplexing of the history file is performed. A duplicate index entry is also created on the SCDS file to maintain an inventory of the SMF data for the duplex file. Duplex files are placed in the alternate archive. The default is COPIES(1), but we recommend that COPIES(2) be set for all SMF data that needs to be kept on a long-term basis. To use the value coded on the OPTIONS statement, specify COPIES(*).

<u>NOCATLG</u>|CATLG|UPCATLG

Indicates whether CA SMF Director should catalog the history file data set, creating an entry within the System Catalog, or not. The default is NOCATLG. To indicate that the system being configured should use the value coded on the OPTIONS statement, specify the UPCATLG operand.

Note: If the PSTORC or ASTORC option is in effect, SMS may override the NOCATLG/CATLG specification.

<u>SCRATCH</u>|NOSCRATCH|UPSCRATCH

Specifies whether CA SMF Director should request specific or nonspecific magnetic tape volumes when creating the history file. If the default SCRATCH is specified, a nonspecific mount request for a scratch tape is performed when a new tape volume is needed for a history file. If NOSCRATCH is specified, it selects a volume serial from the predefined pool of volumes when a new tape volume is needed for a history file. (See the DUMPTAPES control statement). To indicate that the system being configured should use the value coded on the OPTIONS statement, specify the UPSCRATCH operand.

Note: If Tape Management products such as CA 1, CA Vtape Virtual Tape System, or CA TLMS are used, SCRATCH is recommended.

Note: If logstreams or substreams are going to be dumped in this configuration, SCRATCH is required.

NOVERIFY|VERIFY

Indicates whether or not CA SMF Director should enable the Write Validity Check Feature if available on the device used to create a history file. This ensures the integrity of the SMF data written. This product also checks the standard header of each SMF record for validity. The record length, date, and time are all checked for validity. If an error is detected, a dump of the record in question is dumped to the SYSPRINT DD statement, and processing is continued if the default NOVERIFY is specified. When VERIFY is specified, the hardware supplied Write Validity Check feature is enabled, and any errors detected by CA SMF Director results in termination of processing.

Note: This feature is obsolete but is still supported for backward compatibillity.

NOPREFIX|PREFIX(prefix|*)

Specifies the high level qualifiers to be used for the primary history file data set names. If NOPREFIX is specified, CA SMF Director uses SYS1.SMFDATA, which is the default. To use the value of the PREFIX operand on the global OPTIONS statement, code PREFIX(*).

See Attributes of History Files for more details. The prefix can be up to 26 characters.

NOAPREFIX|APREFIX(prefix|*)

Specifies the high level qualifiers to be used for the alternate or duplex history file data set names. If NOAPREFIX is specified or this operand is not specified, the value specified by the PREFIX keyword is used. To use the value of the APREFIX operand on the global OPTIONS statement, code PREFIX(*).

See Attributes of History Files for more details.

EXPDT(0|yyyddd)|*|RETPD(0|nnnn|*)

Specifies the expiration date or retention period of the history file data sets within the primary pool. If the expiration date is specified, it must be in Julian date format. The default is EXPDT(0) or RETPD(0) which causes CA SMF Director not to set an expiration date for the history file. To use the value of the EXPDT or RETPD operand on the global OPTIONS statement, code either EXPDT(*) or RETPD(*).

See Control Statement Language Rules (see page 37) for a description of two special expiration dates: 099000 and 099366.

We recommend that expiration dates or retention periods be used, as it increases the integrity of the SMF data. If the AUTODEL feature is desired, setting expiration dates or retention periods is required.

Note: If the PSTORC option is in effect, SMS may override the specified expiration date or retention period.

AEXPDT(yyyddd|*)|ARETPD(nnnn|*)

Specifies the expiration date or retention period of the history file data sets within the alternate pool. If the expiration date is specified, it must be in Julian date format. For data sets created on a DASD device it requires an operator response and confirmation if the data set is to be scratched or overwritten. If AEXPDT or ARETPD is not specified, CA SMF Director uses the values specified for EXPDT or RETPD. To use the value of the AEXPDT or ARETPD operand on the global OPTIONS statement, code either AEXPDT(*) or ARETPD(*).

See Control Statement Language Rules (see page 37) for a description of two special expiration dates: 099000 and 099366.

We recommend that expiration dates or retention periods be used, as it increases the integrity of the SMF data. If the AUTODEL feature is desired, setting expiration dates or retention periods is required.

Note: If the ASTORC option is in effect, SMS may override the specified expiration date or retention period.

TRTCH(value|*)

ATRTCH(value|*)

Lets you use the z/OS JCL TRTCH parameter on dynamically allocated history files in CA SMF Director. The values of the TRTCH and ATRTCH operands are the same as the values allowed in z/OS JCL when writing an output file to tape. TRTCH is used to set the recording technique for the primary history file, while ATRTCH is used to set the TRTCH for the alternate history file.

Valid values are as follows:

NOCOMP

For cartridge tapes, writes the history file without data compression.

COMP

For cartridge tapes, writes the history file with data compression.

C

Performs data conversion for history files on reel tapes.

E

Writes history files to reel tape with even parity.

ET

Writes history files to reel tape with even parity and perform EBCDIC translation.

T

Writes history files to reel tape and perform EBCDIC translation.

Note: Although CA SMF Director does support TRTCH options C, E, ET, and T, they are not recommended for history files.

If neither TRTCH nor ATRTCH is specified, the system defaults are used. These values can be set in SYS1.PARMLIB or within SMS.

To use the value of the TRTCH or ATRTCH operand on the global OPTIONS statement, code them with (*).

LSNAME(lsname lsname ...)

Specifies which logstreams are recording data that will be managed. Each logstream must be indicated in the configuration for each system that is writing data to that logstream. Up to 21 logstreams can be indicated in one compile. If more than 21 logstreams are needed, you can run the compile more than once with the additional names in the LSNAME operand, and these logstreams will be added to the configuration being processed.

SUBNAME(subname subname ...)

Specifies any substreams you are defining to manage SMF MAN file recorded data. By using substreams, you can define specific recording management attributes for SMF data recorded to SYS1.MANx files based on the SMF record type. The substream names can be up to 26 characters long in a standard 8.8.8 format similar to logstream names. Unlike logstream names, the subnames cannot begin with “I” or "SYS" and must contain at least two nodes, so SUBNAME(VALID.NAME) is allowed but SUBNAME(INVALID) is not allowed.

FIRSTYEAR(yyyy)

Specifies the first year of data that will be allowed in the history file index. During the dump, if an SMF record for this configuration has a lower year than this, it will be rejected as invalid. When adding a history file manually via ADDX, the low date also has to be higher than or equal to this year; otherwise the ADDX will fail. Valid values for this option range from 1960 to the current year or 2059, whichever is lower. The default is 1960.

Note: Changing this option will have no effect on history files already in the index.

MAXFILESONVOL(n|*)

Lets you override the global MAXFILESONVOL at the configuration level. (See the description of MAXFILESONVOL on the OPTIONS statement.) To use the value of the MAXFILESONVOL operand on the global OPTIONS statement, code either MAXFILESONVOL(0) or MAXFILESONVOL(*).

DAYSTACK(NO|YES|*)

Specifies whether the daystacking feature should be turned on or off. The default is NO, which turns daystacking off. To use the value of the DAYSTACK operand on the global OPTIONS statement, code DAYSTACK(*). See Features for a description of the daystacking feature.

PDSNLARGE(NO|YES|*)

Allocates the primary history file with the DSNTYPE=LARGE operand. This operand is only used for DASD history files when the DYNAM operand is specified on the OPTIONS statement; otherwise it is ignored. To use the value of the PDSNLARGE operand on the global OPTIONS statement, code PDSNLARGE(*).

ADSNLARGE(NO|YES|*)

Allocates the alternate history file with the DSNTYPE=LARGE operand. This operand is only used for DASD history files when the DYNAM operand is specified on the OPTIONS statement; otherwise it is ignored. To use the value of the ADSNLARGE operand on the global OPTIONS statement, code ADSNLARGE(*).

DYNAM|DYNAMVTS|UPDYNAM|NODYNAM

Controls whether or not CA SMF Director will dynamically allocate history files when performing dump, extract, and print operations. DYNAM allows dynamic allocation of history files for all dumps and EXTRACT/PRINT operations. NODYNAM prevents dynamic allocation of history files under any circumstances. DYNAMVTS allows dynamic allocation of input history files to EXTRACT and PRINT operations, but suppresses dynamic allocation of output history files from dump operations. To indicate that the system being configured should use the value coded on the OPTIONS statement, use the UPDYNAM operand, which is the default.

Note: If the DYNAM option is in effect and a history DD statement is coded, the parameters in the DD statement override the dynamic allocation options such as PSTORC, ASTORC, and so on. See the descriptions of the PSTORC and ASTORC operands for more information.

Note: The NODYNAM option may cause problems involving incorrect mount messages at data centers that employ virtual tape. The DYNAMVTS option can be used to avoid this problem while maintaining control over allocation of output history files during dumps.

Note: If logstream or substream recorded SMF data will be managed with this configuration, DYNAM must be active for this configuration. This can be set either by specifying DYNAM on the DUMPOPTIONS statement or by specifying UPDYNAM on this DUMPOPTIONS statement and having DYNAM coded on the enterprise-wide OPTIONS statement. We recommend that you code DYNAM on the DUMPOPTIONS statement in case the OPTIONS statement changes in the future.

PDEVN(dname|NONE|*)

'dname' is the esoteric device name to be used when creating SMF history files in the primary pool. Up to eight characters can be specified. It is mutually exclusive with PSTORC. If no device name or SMS storage class is specified and the DYNAM option is enabled, the system default device name is used. This default name can be set in the active ALLOCxx member of the system PARMLIB. Specify NONE to reset the device name. To use the value of the PDEVN operand on the global OPTIONS statement, code PDEVN(*).

Note: If you are changing the PDEVN parameter from one tape unit name to another, from tape to DASD, or DASD to tape, you must immediately run a TAPEINIT command against this SID.

If using DASD for history files, this parameter must only be used for non-SMS managed volumes.

PSTORC(class|NONE|*)

'class' is the IBM DFSMS storage class to be used when creating SMF history files in the primary pool. Up to eight characters can be specified. It is mutually exclusive with the PDEVN keyword. Specify NONE to reset the storage class. To use the value of the PSTORC operand on the global OPTIONS statement, code PSTORC(*).

If using DASD for history files, this operand will allocate files only to SMS-managed volumes. Note that some issues may arise with DASD history file allocations based on how SMS is set up at your data center. For DASD history allocations with the DYNAM option set, there must still be a temporary history file allocated in the JCL that indicates how large the history file should be. SMS may not allow the allocation on the storage class indicated in the PSTORC operand and will indicate an allocation error for the history file, causing the dump process to end prematurely.

To avoid this issue, here are some remedies you can use:

- Include a data set name on the history DD statement. This can be any valid data set name that would be allocated in the storage class indicated in the PSTORC operand.

  Note: This is the simplest remedy and the one that CA recommends.

- Alter the ACS routines in the SMS DASD allocation to assign no class to temporary data sets, where the fourth node of the name is the same as the dump process JCL job name or procedure name.

  Note: These changes are only for DASD history files allocated with SMS. Non-DASD and non-SMS allocations will not be affected by this.

ADEVN(dname|NONE|*)

'dname' is the esoteric device name to be used when creating SMF history files in the alternate pool. Up to eight characters can be specified. It is mutually exclusive with ASTORC. If no device name or SMS storage class is specified and the DYNAM option is enabled, the default device name used will be the device name used for the primary pool. Specify NONE to reset the device name. To use the value of the ADEVN operand on the global OPTIONS statement, code ADEVN(*).

Note: If you are changing the ADEVN parameter from one tape unit name to another, from tape to DASD, or DASD to tape, you must immediately run a TAPEINIT command against this SID.

If using DASD for history files, this parameter must only be used for non-SMS managed volumes.

ASTORC(class|NONE|*)

'class' is the IBM DFSMS storage class to be used when creating SMF history files in the alternate pool. Up to eight characters can be specified. It is mutually exclusive with the ADEVN keyword. Specify NONE to reset the storage class. To use the value of the ASTORC operand on the global OPTIONS statement, code ASTORC(*).

If using DASD for history files, this operand will allocate files only to SMS-managed volumes. Note that some issues may arise with DASD history file allocations based on how SMS is set up at your data center. For DASD history allocations with the DYNAM option set, there must still be a temporary history file allocated in the JCL that indicates how large the history file should be. SMS may not allow the allocation on the storage class indicated in the ASTORC operand and will indicate an allocation error for the history file, causing the dump process to end prematurely.

To avoid this issue, here are some remedies you can use:

- Include a data set name on the history DD statement. This can be any valid data set name that would be allocated in the storage class indicated in the PSTORC operand.

  Note: This is the simplest remedy and the one that CA recommends.

- Alter the ACS routines in the SMS DASD allocation to assign no class to temporary data sets, where the fourth node of the name is the same as the dump process JCL job name or procedure name.

  Note: These changes are only for DASD history files allocated with SMS. Non-DASD and non-SMS allocations will not be affected by this.

SDAY(hhmm)

Specifies the hour (hh) and minute (mm) that CA SMF Director is to treat as the logical start of a day. If this operand is not coded, days start at the value indicated in the OPTIONS statement for this configuration. If no value is set, the value defaults to midnight, coded as SDAY(0000).

Note: The global SDAY value is used to determine the starting and ending times for all EXTRACT and PRINT requests, except when the FROM, JFROM, TO, or JTO operand gives a specific starting or ending time. The value specified here is used for the daystacking feature only.

Sample DUMPOPTIONS Statement:

```
COMPILE CONFIGURATION STORE.
  BEGIN CONFIG(002) SID(XE96) NAME('C.A. XE96').
    DUMPOPTIONS COPIES(2) NOCATLG SCRATCH NOVERIFY
      RETPD(002) ARETPD(001) PREFIX(SYSPROG.CASMF.XE96)
      LSNAME(IFASMF.ALL,IFASMF.T30,IFASMF.RMF)
      TRTCH(NOCOMP) ATRTCH(COMP).
  END CONFIG(002).
```

DUMPOPTIONS statement output:

```
CA                                              AUDIT TRAIL REPORT                                             SMFD       PAGE     1
 CA SMF Director     12.6                                                          SYSA      19 SEP yyyy  10:40:51


  COMPILE CONFIGURATION STORE .

  BEGIN CONFIG(002) SID(XE96) NAME('C.A. XE96') .

  DUMPOPTIONS COPIES(2) NOCATLG SCRATCH NOVERIFY
  RETPD(002) ARETPD(001) PREFIX(SYSPROG.CASMF.XE96)
  LSNAME(IFASMF.ALL,IFASMF.T30,IFASMF.RMF)
  TRTCH(NOCOMP) ATRTCH(COMP).

  END CONFIG(002)  .
CAF1301I  CONFIGURATION DEFINITION XE96 STORED
```

DUMPTAPES

This control statement controls the pool of magnetic tape volume serials available to CA SMF Director for SMF dump processing. A pool of volume serials must be defined to this product if any system configuration was defined with the NOSCRATCH operand of the DUMPOPTIONS or OPTIONS control statement. During dump processing for systems with a configuration definition having NOSCRATCH, this product selects which tape volume serial in the free pool should be used for creating its SMF history files. If no volumes are in the free pool, new tape volume serials must be added.

Note: To list the available volumes in the SCDS pool of tapes, issue the control statements: LISTH VOLUMES (FREE).

Statement Format:

DUMPTAPES

```
              ┌ , ──────┐
►►─ DUMPTAPES ─┬ ADD ────┬─ ( ─▼─ volser ─┴─ ) ──────────────────────────►◄
               └ DELETE ─┘
```

Operands and Descriptions:

ADD|DELETE

Indicates whether the system should ADD or DELETE the specified volume serial numbers from the pool of available dump tapes.

Volser may contain up to 255 volume serial numbers in a single DUMPTAPES statement.

To ADD and DELETE volumes, you must include separate DUMPTAPES statements.

Sample DUMPTAPES Statements:

```
DUMPTAPES  ADD(SMF100 SMF101 SMF102 SMF103 SMF104 SMF105 SMF106) .

DUMPTAPES  DELETE(SMF106 SMF107) .
```

DUMPTAPES statement output:

```
CA                                             AUDIT TRAIL REPORT                              SMFD       PAGE     1
CA SMF Director     12.6                                                             SYSA      23 JUL yyyy  16:33:32

    DUMPTAPES  ADD(SMF100 SMF101 SMF102 SMF103 SMF104 SMF105 SMF106) .

CAF2705I  DUMPTAPES ADD FUNCTION WAS SUCCESSFUL

    DUMPTAPES  DELETE(SMF106 SMF107) .

CAF2500E  VOLUME SMF107 IS NOT IN AVAILABLE VOLUME POOL

CAF2901I  RUN COMPLETE, HIGHEST CONDITION CODE WAS 8
```

END

This control statement concludes the definition of a configuration to CA SMF Director. END is always used in association with the COMPILE and BEGIN control statements.

Statement Format:

END

```
►►─ END ─ CONFIG(n) ─ . ─────────────────────────────────────────────────►◄
```

Operands and Descriptions:

CONFIG(n)

Specifies the end of a configuration's definition.

"(n)" must match the number specified in the associated BEGIN statement.

Sample END Statement:

```
COMPILE CONFIGURATION STORE.
    BEGIN CONFIG(22) SID(XE22) NAME('C.A. DEMO SYSTEM-2') .
     DUMPOPTIONS COPIES(2) CATLG SCRATCH NOVERIFY
       RETPD(5) PREFIX(SYS3.SMFDATA)
      ARETPD(10) .
    END CONFIG(22) .
```

END statement output:

```
 CA                                              AUDIT TRAIL REPORT                          SMFD       PAGE     1
CA SMF Director     12.6                                                         SYSA      23 JUL yyyy  15:34:10

   COMPILE CONFIGURATION STORE.
       BEGIN CONFIG(22) SID(XE22) NAME('C.A. DEMO SYSTEM-2') .
        DUMPOPTIONS COPIES(2) CATLG SCRATCH NOVERIFY
          RETPD(5) PREFIX(SYS3.SMFDATA)
         ARETPD(10) .
       END CONFIG(22) .

 CAF1301I  CONFIGURATION DEFINITION XE22 STORED

 CAF2900I  RUN COMPLETE
```

EXTRACT/PRINT

These control statements select the SMF data that you want to be:

- copied and processed by another program
- printed in its original form
- both of the above

If this product detects duplicate or overlapping SMF data, it will by default not allow any records to be extracted or printed from the history file data set causing this condition. To allow duplicate or overlapping SMF data to be extracted or printed, specify the PERMIT parameter on the EXEC JCL statement. To have overlapping data automatically discarded during the EXTRACT/PRINT process, specify the NODUPS parameter on the EXEC JCL statement. For further information, see the *Systems Programmer Guide*.

EXTRACT and PRINT take the same operands with the exception of the EXIT and TODD parameters, which pertain only to EXTRACT. You may include more than one EXTRACT or PRINT control statement in the same execution of CA SMF Director.

By default, EXTRACT and PRINT requests will be satisfied from the primary pool of history files. You may use the SOURCE control statement to specify the Alternate pool.

Only those SMF records having a standard header section as defined in the IBM *z/OS Systems Management Facilities Manual* will be selected for processing. CA SMF Director does not extract SMF record types 2 and 3, which are dump header and trailer records written by the IBM SMF dump programs.

Special Characteristics of the EXTRACT Control Statements

CA SMF Director honors user-defined record formats and truncates or pads records, as required. During extraction, a message is issued indicating the number of records truncated if the logical record length (LRECL) is not large enough to accommodate the largest record encountered. If the DD statement for the output file does not specify a block size, record length, or record format, CA SMF Director uses default values of LRECL=32760 and RECFM=VBS.

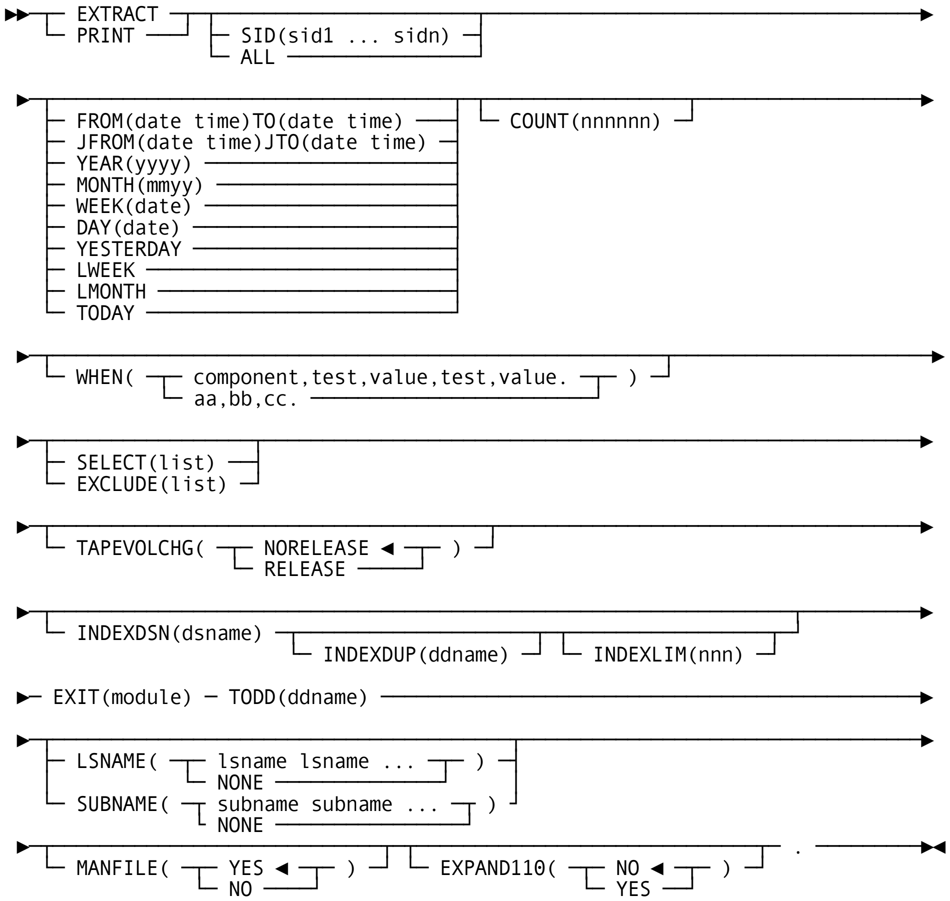
In addition, the EXTRACT statement can also create entries in a split file index to facilitate the automated application processing of SMF data that would normally have been saved to a split file during the dump process. See the SPLIT control statement for more information on split files and the index.

Special Characteristics of the PRINT Control Statements

Data is printed in standard dump format (character and hex), with a formatted heading of the information in the standard SMF record header. The record descriptor word (RDW) of the SMF record is not printed.

Statement Format:

EXTRACT/PRINT



Operands and Descriptions:

SID(sid1...sidn)

Specifies the identification of one or more (operating) systems for which data is to be extracted.

"(sid)" may identify up to 20 systems. You may code this part of the statement as follows: SID(sid1,sid2...). Either SID or ALL operand must be specified.

ALL

Causes data to be extracted for all the indexed systems, for which CA SMF Director has an entry, that fall within the specified time range. Either SID or ALL operand must be specified.

FROM (date time)

TO (date time)

Specifies the beginning and ending of the period for which SMF data is to be extracted. SMF records whose timestamps exactly match the FROM specification are considered to be inside the time range. SMF records whose timestamps exactly match the TO specification are considered to be outside the time range.

"date" is in Gregorian (mmddyy) format. (e.g., July 1, 2008 would be 070108.)

"time" is coded in hour/minute (hhmm) format. For EXTRACT, time can be coded in hour/minute/second (hhmmss) format.

JFROM(date time)

Specifies the beginning date and time for the SMF records to be extracted or printed. "Date" is in Julian date format (cyyddd), where "c" is 0 for 1960 through 1999 and 1 for 2000 and thereafter. Consequently, 098182 represents July 1, 1998, and 101001 represents January 1, 2001. "Time" is coded in hour, minute and second format (hhmmss or hhmm). Code either the JFROM keyword or the FROM keyword, but not both.

JTO(date time)

Specifies the ending date and time for the SMF records to be extracted or printed. "Date" is in the Julian format (cyyddd). "Time" is coded in hour, minute, and second format (hhmmss or hhmm). Code either the JTO keyword or the TO keyword, but not both.

Note: If an EXTRACT statement includes at least one FROM, TO, JFROM, or JTO parameter with a starting or ending time explicitly coded, the installation's logical day specification (the SDAY value defined on the OPTIONS statement) will be ignored. If only a starting time is coded, the ending time will default to midnight at the end of the TO/JTO date. If only an ending time is coded, the starting time will default to midnight at the beginning of the FROM/JFROM date.

- SMF records whose timestamps exactly match the JFROM specification are considered to be inside the time range.
- SMF records whose timestamps exactly match the JTO specification are considered to be outside the time range.

YEAR(yy)

Specifies that the SMF data for an entire calendar year is to be processed. If "yy" is 59 or less, the year is treated as 20yy. If "yy" is 60 or greater, the year is treated as 19yy. Consequently, YEAR(2000) requests extraction of data beginning January 1 and ending December 31, 2000.

MONTH(mmyy)

Specifies that the SMF data for an entire calendar month is to be processed. For example, "(0408)" specifies April 2008.

WEEK(date)

Specifies that the SMF data for an entire calendar week (168 continuous hours) beginning with the specified date, is to be processed.

"Date" is in Gregorian (mmddyy) format.

DAY(date)

Specifies that the SMF data for one calendar day is to be processed.

"date" is in Gregorian (mmddyy) format.

YESTERDAY

Specifies processing of SMF data for yesterday's calendar day. "Yesterday" is the current logical date minus one, at the time the extract is run.

LWEEK

Specifies that SMF data is to be extracted or printed for the last calendar week, running Sunday through Saturday.

LMONTH

Specifies that SMF data is to be extracted or printed for the last calendar month. For example, if an EXTRACT LMONTH is run on July 2008, CA SMF Director will extract all data from June 2008.

TODAY

Specifies that SMF data is to be extracted or printed for the current logical day.

Note: In determining what is the current day, week, or month for the purpose of processing the YESTERDAY, LWEEK, LMONTH, or TODAY parameter, CA SMF Director takes into account the installation's system-wide SDAY option. For example, suppose that an installation's SDAY option is set to 0900 and an EXTRACT is run at 0800 on Sunday, July 1, 2008. In that case the LMONTH operand would result in extraction of all data between 0900 on May 1, 2008 and 0900 on June 1, 2008; LWEEK would result in extraction of all data between 0900 on Sunday, June 17 and 0900 on Sunday, June 24; YESTERDAY would cause all data between 0900 on June 29 and 0900 on June 30 to be extracted; TODAY would cause all data between 0900 on June 30 and 0900 on July 1 to be extracted. This is because the logical date would still be Saturday, June 30. Running EXTRACT an hour later would change the results of all three of these operands, as the logical date would become July 1 at 0900 on that date. Thus, the LMONTH, LWEEK, YESTERDAY, and TODAY operands would extract or print the data between:

- 0900 on June 1 and 0900 on July 1
- 0900 on Sunday, June 24 and 0900 on Sunday, July 1
- 0900 on June 30 and 0900 on July 1.
- 0900 on July 1 and 0900 on July 2.

In all cases, SMF records whose timestamps exactly match the beginning of the specified interval are selected, while SMF records whose timestamps exactly match the end of the specified interval are not selected.

COUNT(n)

Specifies the maximum number of records to be extracted.

"(n)" must be a postive integer.

WHEN(aa,bb,cc) or WHEN(component,test,value,test,value...)

Allows the specification of additional selection criteria for SMF records. The WHEN selection criteria can be based on either offsets or up to five components, but not both. For a description of component-based WHEN selection criteria, see the SPLIT statement. The offset-based WHEN selection criteria is described here.

A SMF record is selected for output processing when it meets all of the specified conditions. There may be up to four occurrences of 'aa,bb,cc'.

'aa' is the numeric offset within the SMF record relative to zero (column 1) that is to be checked. The offset can be any number from 14 up to 32767.

Note: If you are looking at a listing from an IDCAMS dump or SMF display from an earlier PRINT command, remember that neither of these methods displays the record descriptor word (RDW). Therefore, the WHEN offset must be coded as 4 greater than the apparent displacement of the field as shown in an IDCAMS dump or a CA SMF Director PRINT command listing.

'bb' is any one of the following two-character operators:

- EQ - Equal
- NE - Not Equal
- LT - Less than
- LE - Less than or equal to
- GT - Greater than
- GE - Greater than or equal to

'cc' is the character or hexadecimal constant up to 32 bytes in length. It may be coded as a hexadecimal value in the format X'zz', where 'zz' is an even number of hexadecimal digits. For example:

```
EXTRACT ALL MONTH(0408)SELECT(14,15)WHEN(68,EQ,'SYS1')
```

results in the extraction of all SMF record type 14 and 15 records generated during April 2008 for data sets whose names begin with SYS1.

SELECT(list)|EXCLUDE(list)

Specifies selection of only certain record types for processing, or exclusion of certain record types for processing. If neither or these two operands is coded, CA SMF Director processes all stored SMF records that fall within the specified date and time range.

'list' may include any valid SMF record types and subtypes; the valid numbers being 0 through 255. They may be expressed as individual record types separated by a comma, and/or as a range separated by a colon. Subtypes, if specified, must be coded as three decimal places; e.g., 30.004. For example, SELECT(30.001:30.005) results in the retrieval of SMF record type 30, subtypes 1 through 5. EXCLUDE(99.02,99.1) causes the retrieval of all stored SMF records except for type 99, subtypes 20 and 100.

Some SMF records have subtypes of zero. To specify subtype zero, a value of zero must be placed to the right of the decimal point. For example, SELECT(116.000) indicates SMF Records of type 116, Subtype 0, while SELECT(116) indicates all type 116 records.

TAPEVOLCHG(NORELEASE|RELEASE)

Specifies whether CA SMF Director should release the history tape unit when it is about to call for the mount of a new history volume. The default is NORELEASE.

Note: TAPEVOLCHG does not apply to history files that reside on DASD.

INDEXDSN(dsn)

(For EXTRACT only.) Specifies the name of a split file index, where information about this extract is recorded. This is an optional parameter provided to facilitate processes that use the split file index to process data that has been archived by CA SMF Director. The parameter should contain the data set name of the split file index that is used in the automated process. For more information about split file indexes, see the description of the SPLIT control statement. For more information on setting up a split file index, see the *Systems Programmer Guide.*

A single, specific SID must be coded when using INDEXDSN. ALL is not supported.

INDEXDUP(ddname)

(For EXTRACT only.) Specifies the ddname in the extract JCL where a copy of the split file index will be written. This operand is valid only if the INDEXDSN operand is present; otherwise it is ignored.

INDEXLIM(nnn)

(For EXTRACT only.) Specifies the maximum number of entries that are kept in the split file index. This operand is valid only if the INDEXDSN operand is present; otherwise it is ignored.

EXIT(module)

(For EXTRACT only.) Specifies the module to be called as an exit to receive extracted SMF data for further processing. TODD must also be coded if this product is to write the extracted output file; otherwise it assumes that the exit module will process the extracted data as desired.

"(module)" is the 1 to 8-character name of a module to be dynamically loaded during the processing of this request. Refer to "Writing Extraction Exits" in Chapter 3 of the *Systems Programmer Guide.*

TODD(ddname)

(For EXTRACT only.) Specifies the name of the DD statement defining the file to which the extracted SMF records are to be written.

"(ddname)" must be a 1 to 8-character ddname, corresponding to a DD statement in the execution JCL for the job.

More than one EXTRACT request may be coded for the same DD. If the TODD parameter is omitted, CA SMF Director will use a default ddname of EXTRACT.

LSNAME(NONE|lsname lsname ...)

Specifies a subset of logstream names that CA SMF Director will include in the search to satisfy an EXTRACT or PRINT request. If this operand is not included, all of the logstreams being managed will be included in the requests. LSNAME(NONE) means that none of the logstreams should be included when searching for records to satisfy a request.

SUBNAME(NONE|subname subname ...)

Specifies a subset of substream names that CA SMF Director will include in the search to satisfy an EXTRACT or PRINT request. If this operand is not included, all of the substreams being managed will be included in the requests. SUBNAME(NONE) means that none of the substreams should be included when searching for records to satisfy a request.

MANFILE(YES|NO)

Specifies whether the MAN file archives should be searched to find records to satisfy an EXTRACT or PRINT request. The default value of YES includes the MAN files in the search. NO limits the request to search only the logstreams and substreams.

EXPAND110(YES|NO)

Indicates whether or not compressed SMF type 110 records should be expanded before they are either written to the extract file or printed. If the value of the operand is YES, CA SMF Director will expand any compressed type 110 records that are to be written to the output file or printed. The default value is NO, meaning that no expansion of type 110 records will be performed.

Sample EXTRACT Statements:

Obtain SMF records from systems XE21, XAE1 and XAE4 from December 16, 2008 at 8:00 a.m. to December 16, 2008 at 5:30 p.m. Exclude SMF record types: 14,15,70,71,72,73,74,75,76,77,78,79:

```
EXTRACT SID(XE21,XAE1,XAE4)
     FROM(121608 080000) TO(121608 173000)
        EXCLUDE(14:15,70:79).
```

Obtain RMF records from all systems for the first quarter of the year 2008, January 1 2008 to March 31 2008:

```
EXTRACT SID(ALL) JFROM(108001 000000) JTO(108091 240000)
     SELECT(70:79).
```

Obtain all SMF records for system XE96 and XE74 from April 16, 2008 at 4:00 p.m. to September 27, 2008 at 12:30 p.m. Place all records in the data set referenced in the JCL by ddname SMFOUT:

```
EXTRACT SID(XE96,XE74)  JFROM(108107 1600) JTO(108271 1230)
     TODD(SMFOUT).
```

Obtain RMF records from 10 a.m. to 4 p.m. on January 17, 2008, but only include data from logstreams IFASMF.LSTRM1 and IFASMF.LSTRM2 and only from system XFF1:

```
EXTRACT SID(XFF1) FROM(011708 100000) TO(011708 160000)
     SELECT(70:79) MANFILE(NO) TODD(SMFOUT)
     LSNAME(IFASMF.LSTRM1 IFASMF.LSTRM2).
```

Extract any records from the last month on system XFF1 that were not captured in a logstream:

```
EXTRACT SID(XFF1) LMONTH TODD(SMFOUT) LSNAME(NONE).
```

EXTRACT statement output:

```
CA HERNDON                                      AUDIT TRAIL REPORT                            SMFD       PAGE     1
 CA SMF Director     12.6                                                            SYSA      20 SEP yyyy   8:50:57

  EXTRACT SID(XE96,XE74)  JFROM(107100 1600) JTO(107270 1230)
       TODD(SMFOUT).

CAF1010I  REQUEST ACCEPTED

CAF2218I  SMF DATA USING CONFIGURATION XE96 IS BEING PROCESSED
CAF2221I  SMF DATA USING CONFIGURATION XE96 IS BEING PROCESSED WITH LOGSTREAM IFASMF.ALL
CAF2221I  SMF DATA USING CONFIGURATION XE96 IS BEING PROCESSED WITH LOGSTREAM IFASMF.RMF
CAF2221I  SMF DATA USING CONFIGURATION XE96 IS BEING PROCESSED WITH LOGSTREAM IFASMF.T30
CAF2218I  SMF DATA USING CONFIGURATION XE74 IS BEING PROCESSED
CAF2221I  SMF DATA USING CONFIGURATION XE74 IS BEING PROCESSED WITH LOGSTREAM IFASMF.SMFDATA
CAF2221I  SMF DATA USING CONFIGURATION XE74 IS BEING PROCESSED WITH LOGSTREAM IFASMF.WHISTLE
CAF3100I  15669 RECORDS WRITTEN TO SMFOUT FILE
CAF3105I  6954 RECORDS PROCESSED FROM: 08/01/2008  9:33:37  TO: 09/19/2008 10:13:33 FOR SYSID XE96
CAF3105I  8715 RECORDS PROCESSED FROM: 07/23/2008 10:52:36  TO: 07/24/2008 11:14:03 FOR SYSID XE74

                                              SUMMARY OF RECORDS EXTRACTED

                     FROM: 07/23/2008 (107.204) 10:52:36.56    TO: 09/19/2008 (107.262) 10:13:33.83

         TYPE      AVGL   MAXL      TOTAL   ............................... SUBTYPE COUNTS ...............................
            0        60     60            1
            4       238    319           71
            5       148    154           33
            8       624    624            1
           14       359    632          356
           15       333    344          134
           17       100    100           20
           19        72     72        1,425
           20        95    101           35
           22       302    302            1
           23       134    134           24
           26       443    451           52
           30      1062   3773        2,418      1 -       104       2 -     1,485       3 -       128       4 -       127
                                                 5 -        64       6 -       510
           32       399    548           14      1 -         4       2 -         3       3 -         4       4 -         3
           34       700    703            3
           35       154    154            3
           40       219    546           84
           41       136    212          104      1 -         2       3 -       102
           42       472  20972          771      5 -        51       6 -       720
           43        32     32            1
           50       164    242           50
           52        62     62           10
           60       516    600          104
           61       615   3648           11
           62       188    188           92
           64       458    458           88
           65       316    381           12
           80       256    256        1,851
           88       222    308          109      1 -        55      11 -        54
           89      2363   4478          100      1 -        50       2 -        50
           90       326    596           32
           92       213    276        1,245      1 -         9      10 -        26      11 -     1,210
           99       907   2996        3,323      1 -       373       2 -     1,848       3 -       370       6 -       372
                                                 9 -       360
          119       236    644           63      5 -         2       6 -         2       7 -         1      10 -        58
          231       292    292            3
          255       760    760        3,025      2 -     3,025
          ALL       624  20972       15,669

CAF2900I  RUN COMPLETE
```

The following list gives an explanation of the fields contained in the Summary of Records Extracted section of the report.

TYPE

SMF record type (numeric 0-255).

AVGL

Average record length of records for this type.

MAXL

Maximum record length of records for this type.

TOTAL

Number of records for this type successfully written to output.

SUBTYPE COUNTS

A list of SMF record subtypes and how many records for each subtype were successfully written. The first number shown is the numeric SMF record subtype, followed by the number of records. Counts for records with no subtype (zero) are not shown on the report. However, they are included in the total number of records.

ALL

Indicates the grand total line for all SMF record types written.

Sample PRINT Statement:

```
PRINT SID(XE44) FROM(041698 131500)  TO(041698 132000)
              COUNT(5).
```

PRINT statement output:

```
CA                                              AUDIT TRAIL REPORT                             SMFD       PAGE     1
CA SMF Director     12.6                                                               SYSA      27 JUL yyyy  11:45:24

 SOURCE PRIMARY.

CAF2800I  SOURCE OPTION SET

 PRINT SID(XE44) FROM(041698 131500)  TO(041698 132000) COUNT(5) .
CAF1010I  REQUEST ACCEPTED


CAF3200I  SMF RECORD TYPE 21 BUFFER DATE 04/16/2008 ( 98/106) TIME 13:16:44
         0000    1E150048 F1980098 106FE7C5 F4F40036   F3F0F5F0 F9F00752 78048081 00000033    *....1.....XE44..305090..........*
         0020    00000000 00000000 00000000 00000000   00000000 00000100 00000000 00000000    *................................*
         0040    00000000                                                                     *....                            *

CAF3200I  SMF RECORD TYPE 21 BUFFER DATE 04/16/2008 ( 98/106) TIME 13:16:56
         0000    1E150048 F6700098 106FE7C5 F4F40036   F3F0F3F3 F0F2074D 78048081 00000049    *....6.....XE44..303302..........*
         0020    00000000 00000000 00000000 00000000   00000000 00037800 00000000 00000000    *................................*
         0040    00000000                                                                     *....                            *
CAF3100I  2 RECORDS WRITTEN TO SYSPRINT FILE

CAF2900I  RUN COMPLETE
```

LISTC

This control statement enables you to list information associated with the configurations of one or more SIDs (SMF IDs).

Statement Format:

LISTC

```
►►─ LISTC ─┬─────────────────────┬─ . ──────────────────────────────►◄
           ├─ SID(sid1 ... sidn) ─┤
           └─ ALL ────────────────┘
```

Operands and Descriptions:

SID(sid1...sidn)

Specifies one or more SMF IDs, up to a maximum of 20, for which the associated options are to be printed. This may be coded (sid1,sid2...).

ALL

Requests printing of the options for all configurations.

If no operands are specified, a list of all SID names on the SCDS file, including temporary SIDs defined in the input stream before the LISTC control statements, will be generated.

Sample LISTC Statement:

```
LISTC ALL.
```

LISTC statement output (note that operands preceded by an asterisk have picked up their values from a higher level):

```
CA HERNDON                                        AUDIT TRAIL REPORT                                SMFD        PAGE     1
CA SMF Director   12.6                                                                    XE96      26 JAN YYYY   8:53:40


LISTC ALL.
```

```
CA HERNDON                                      AUDIT TRAIL REPORT                            SMFD       PAGE     2
CA SMF Director  12.6                                                              XE96      26 JAN YYYY   8:53:40

COMPLEX LEVEL CONFIGURATION:
     SITE NAME:  CA HERNDON           LAST DUMP ON: 01/26/YYYY  3:50:40
          SDAY(00:00)  MAXLINES(000)  OVERLAP(0005)    AUTODEL    DYNAM
          PDEVN(3390    )  ADEVN(3390    )  MAXFILESONVOL(00002)
          COPIES(1)  NOCATLG  NOSCRATCH  DAYSTACK(NO) PDSNLARGE(NO) ADSNLARGE(NO)
          EXPDT(0) NOPREFIX  AEXPDT(0) NOAPREFIX
SYSTEM LEVEL CONFIGURATION:
  SYSID: XAD1  SEQ#: 1  DATE DEFINED: 01/07/YYYY  NAME: C.A. XAD1
    OPTIONS:
    COPIES(2)   NOVERIFY    SCRATCH     NOCATLG     RETPD(2)      PREFIX(PREFIX.CASMF.TEST.XAD1)
      ARETPD(1) APREFIX(PREFIX.CASMF.TEST.XAD1) TRTCH(NOCOMP ) ATRTCH(COMP   ) DAYSTACK(NO)
      *MAXFILESONVOL(00002) FIRSTYEAR(1960) PDSNLARGE(NO) ADSNLARGE(NO)
      *DYNAM    *PDEVN(3390    )  *ADEVN(3390    )  *SDAY(00:00)

SYSTEM LEVEL CONFIGURATION:
  SYSID: XE96  SEQ#: 2  DATE DEFINED: 01/07/YYYY  NAME: C.A. XE96
    STATISTICS:
     LAST DUMP ON: 01/26/YYYY  3:50:38  ON SYSTEM: XE96      RECORD COUNT: 255  AVERAGE: 513
         CPU TIME:     0.126  AVERAGE:     0.023    SMF DATA TIME RANGE:   0:01  AVERAGE:   1:27
         LAST SMF RECORD TIMESTAMP: 01/26/YYYY  3:50:30.98
    OPTIONS:
    COPIES(2)   NOVERIFY    SCRATCH     NOCATLG     RETPD(2)      PREFIX(PREFIX.CASMF.TEST.XE96)
      ARETPD(1) APREFIX(PREFIX.CASMF.TEST.XE96) TRTCH(NOCOMP ) ATRTCH(COMP   ) DAYSTACK(NO)
      *MAXFILESONVOL(00002) FIRSTYEAR(1960) PDSNLARGE(NO) ADSNLARGE(NO)
      *DYNAM    *PDEVN(3390    )  *ADEVN(3390    )  *SDAY(00:00)

STREAM LEVEL CONFIGURATION:
    STREAMNAME: DUMMY.STREAM               ID: RB  TYPE: SUBSTREAM  SYSID: XE96
    STATISTICS:
     DUMPS: 0
    OPTIONS:
    *COPIES(2)   *NOVERIFY    *SCRATCH     *NOCATLG     *RETPD(2)      *PREFIX(PREFIX.CASMF.TEST.XE96)
      *ARETPD(1) *APREFIX(PREFIX.CASMF.TEST.XE96) *TRTCH(NOCOMP ) *ATRTCH(COMP   ) *DAYSTACK(NO)
      *MAXFILESONVOL(00002) FIRSTYEAR(1960) *PDSNLARGE(NO) *ADSNLARGE(NO)
      *DYNAM    *PDEVN(3390    )  *ADEVN(3390    )  *SDAY(00:00)
    RECORDS:
      NONE

    STREAMNAME: IFASMF.ALL                 ID: LA  TYPE: LOGSTREAM  SYSID: XE96
    STATISTICS:
     DUMPS: 35            LAST SMF RECORD TIMESTAMP: 01/26/YYYY  3:43:46
    OPTIONS:
    *COPIES(2)   *NOVERIFY    *SCRATCH     *NOCATLG     *RETPD(2)      *PREFIX(PREFIX.CASMF.TEST.XE96)
      *ARETPD(1) *APREFIX(PREFIX.CASMF.TEST.XE96) *TRTCH(NOCOMP ) *ATRTCH(COMP   ) *DAYSTACK(NO)
      *MAXFILESONVOL(00002) FIRSTYEAR(1960) *PDSNLARGE(NO) *ADSNLARGE(NO)
      *DYNAM    *PDEVN(3390    )  *ADEVN(3390    )  *SDAY(00:00)

    STREAMNAME: IFASMF.RMF                 ID: LC  TYPE: LOGSTREAM  SYSID: XE96
    STATISTICS:
     DUMPS: 0
    OPTIONS:
    *COPIES(2)   *NOVERIFY    *SCRATCH     *NOCATLG     *RETPD(2)      *PREFIX(PREFIX.CASMF.TEST.XE96)
      *ARETPD(1) *APREFIX(PREFIX.CASMF.TEST.XE96) *TRTCH(NOCOMP ) *ATRTCH(COMP   ) *DAYSTACK(NO)
```

```
CA HERNDON                                      AUDIT TRAIL REPORT                            SMFD       PAGE     3
CA SMF Director  12.6                                                                  XE96      26 JAN YYYY   8:53:40

      *MAXFILESONVOL(00002) FIRSTYEAR(1960) *PDSNLARGE(NO) *ADSNLARGE(NO)
      *DYNAM    *PDEVN(3390    )  *ADEVN(3390    )  *SDAY(00:00)

    STREAMNAME: IFASMF.T30                 ID: LB  TYPE: LOGSTREAM  SYSID: XE96
    STATISTICS:
     DUMPS: 23            LAST SMF RECORD TIMESTAMP: 01/26/YYYY  3:43:24
    OPTIONS:
    *COPIES(2)   *NOVERIFY    *SCRATCH     *NOCATLG     *RETPD(2)      *PREFIX(PREFIX.CASMF.TEST.XE96)
      *ARETPD(1) *APREFIX(PREFIX.CASMF.TEST.XE96) *TRTCH(NOCOMP ) *ATRTCH(COMP   ) *DAYSTACK(NO)
      *MAXFILESONVOL(00002) FIRSTYEAR(1960) *PDSNLARGE(NO) *ADSNLARGE(NO)
      *DYNAM    *PDEVN(3390    )  *ADEVN(3390    )  *SDAY(00:00)

    STREAMNAME: SUBSTR.T30                 ID: RA  TYPE: SUBSTREAM  SYSID: XE96
    STATISTICS:
     DUMPS: 56            LAST SMF RECORD TIMESTAMP: 01/26/YYYY  3:20:42
    OPTIONS:
    *COPIES(2)   NOVERIFY    SCRATCH     *NOCATLG     *RETPD(2)      *PREFIX(PREFIX.CASMF.TEST.XE96)
      *ARETPD(1) *APREFIX(PREFIX.CASMF.TEST.XE96) *TRTCH(NOCOMP ) *ATRTCH(COMP   ) *DAYSTACK(NO)
      *MAXFILESONVOL(00002) *PDSNLARGE(NO) *ADSNLARGE(NO)
      DYNAM    *PDEVN(3390    )  *ADEVN(3390    )  *SDAY(00:00)
    RECORDS:
      030 031 032 033 034 035


SYSTEM LEVEL CONFIGURATION:
```

```
CA HERNDON                                      AUDIT TRAIL REPORT                            SMFD       PAGE     4
CA SMF Director  12.6                                                                  XE96      26 JAN YYYY   8:53:40

  SYSID: XE74  SEQ#: 3  DATE DEFINED: 01/07/YYYY  NAME: C.A. XE74
    OPTIONS:
    COPIES(2)   NOVERIFY    SCRATCH     NOCATLG     RETPD(2)      PREFIX(PREFIX.CASMF.TEST.XE74)
      ARETPD(1) APREFIX(PREFIX.CASMF.TEST.XE74) TRTCH(NOCOMP ) ATRTCH(COMP   ) DAYSTACK(NO)
      *MAXFILESONVOL(00002) FIRSTYEAR(1960) PDSNLARGE(NO) ADSNLARGE(NO)
      *DYNAM    *PDEVN(3390    )  *ADEVN(3390    )  *SDAY(00:00)

STREAM LEVEL CONFIGURATION:
    STREAMNAME: IFASMF.SMFDATA             ID: LB  TYPE: LOGSTREAM  SYSID: XE74
    STATISTICS:
     DUMPS: 0
    OPTIONS:
    *COPIES(2)   *NOVERIFY    *SCRATCH     *NOCATLG     *RETPD(2)      *PREFIX(PREFIX.CASMF.TEST.XE74)
      *ARETPD(1) *APREFIX(PREFIX.CASMF.TEST.XE74) *TRTCH(NOCOMP ) *ATRTCH(COMP   ) *DAYSTACK(NO)
      *MAXFILESONVOL(00002) FIRSTYEAR(1960) *PDSNLARGE(NO) *ADSNLARGE(NO)
      *DYNAM    *PDEVN(3390    )  *ADEVN(3390    )  *SDAY(00:00)

    STREAMNAME: IFASMF.WHISTLE             ID: LA  TYPE: LOGSTREAM  SYSID: XE74
    STATISTICS:
     DUMPS: 0
    OPTIONS:
    *COPIES(2)   *NOVERIFY    *SCRATCH     *NOCATLG     *RETPD(2)      *PREFIX(PREFIX.CASMF.TEST.XE74)
      *ARETPD(1) *APREFIX(PREFIX.CASMF.TEST.XE74) *TRTCH(NOCOMP ) *ATRTCH(COMP   ) *DAYSTACK(NO)
      *MAXFILESONVOL(00002) FIRSTYEAR(1960) *PDSNLARGE(NO) *ADSNLARGE(NO)
      *DYNAM    *PDEVN(3390    )  *ADEVN(3390    )  *SDAY(00:00)
```

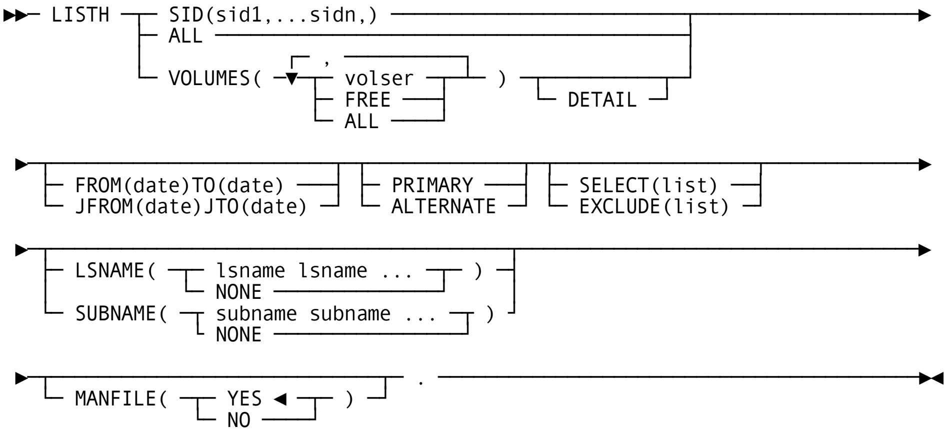
```
CA HERNDON                                      AUDIT TRAIL REPORT                            SMFD       PAGE     5
CA SMF Director  12.6                                                                  XE96      26 JAN YYYY   8:53:40



CAF2900I  RUN COMPLETE
```

LISTH

This control statement enables you to print a list of all or part of the index information concerning its history data sets. It is also used to produce a list of available volumes in the tape pool (if any). This information is maintained in the SCDS file.

Statement Format:

LISTH



Operands and Descriptions:

SID(sid,sid,...)

Specifies one or more SMF IDs, up to a maximum of 20, for the history-file index. This may be coded (sid1,sid2...).

This is the 4-character identifier that appears in the SMF data.

ALL

Specifies that index entries for all configurations are to be listed.

FROM(date) TO(date)

Specifies the beginning and ending of the period for which SMF index entries are to be displayed.

'date' is in Gregorian (mmddyy) format.

JFROM (jdate) JTO (jdate)

Specifies the beginning and ending dates for the period for which

'jdate' is in Julian date format (cyyddd).

PRIMARY

Specifies that index entries in the primary pool are the only ones to be listed.

ALTERNATE

Specifies that index entries in the alternate pool are the only ones to be listed.

VOLUMES (volser, volser, ...)

Produces a list of tape volume serials defined to this product. Only the volumes specified will be listed.

VOLUMES (ALL)

Produces a list of all tape volume serials defined to this product.

VOLUMES (FREE)

Produces a list of all available volume serials in the free pool.

DETAIL

Specifies that the index entries that match the specified volume serials on the VOLUMES keyword will be used.

SELECT(list)|EXCLUDE(list)

Lets you filter the history files index entries on an additional record type criteria. If SELECT is coded, all history files that contain at least one of the specified record types will be listed in the LISTH output. If EXCLUDE is coded, all history files that contain only the specified record types and nothing else will be excluded from the output.

The list may include any valid SMF record types from 0 through 255. They may be coded as individual record types separated by commas or as a range of record types separated by colons. Up to 255 values may be coded. Selection on subtypes is not allowed.

For example, SELECT(0,255,30:71) lists the history files that contain record types 0, 255, and 30 through 71. EXCLUDE(255) suppresses all history files that contain only type 255 records.

Note: Entries that were manually added by ADDX do not contain information about their record types. Therefore, they are always listed in the LISTH output. Record type information may be updated by an EXTRACT run, which gathers information about record types.

LSNAME(NONE|lsname lsname ...)

Lets you filter the history file index entries on a specified logstream. Only the entries that contain the data from the selected logstreams will be listed. Up to 21 logstreams may be coded.

If NONE is coded, no logstream-related history file entry will be included in the output.

If LSNAME is not coded, all history files dumped from all logstreams will be included.

SUBNAME(NONE|subname subname ...)

Lets you filter the history file index entries on a specified substream. Only the entries that contain the data from the selected substreams will be listed. Up to 21 substreams may be coded.

If NONE is coded, no substream-related history file entry will be included in the output.

If SUBNAME is not coded, all history files dumped from all substreams will be included.

MANFILE(YES|NO)

Specifies whether the MAN file archives should be included in the LISTH output. The default value of YES lists all history files containing the data dumped from MAN files. NO limits the output to only logstream-related history files.

Sample LISTH Statements:

```
LISTH ALL .
LISTH  VOLUMES(FREE)    .
```

LISTH statement output:

```
CA                                              AUDIT TRAIL REPORT                           SMFD       PAGE     1
CA SMF Director     12.6                                                           SYSA      27 JUL yyyy  11:09:54

 LISTH ALL .



CA                                              AUDIT TRAIL REPORT                            SMFD       PAGE     3
CA SMF Director     12.6                                                            SYSA      27 JUL yyyy  11:09:54

                                    INDEXED SMF DATA FOR CONFIGURATION XE44

------ FROM ------- ------- TO -------- ------ DUMPED ----- VOLID/  UNIT  FILE -------------- DATA SET NAME ---------- SID   X
04/29/2008 10:30:19 04/29/2008 10:44:44 07/08/2008 18:33:41 201825  3490  0001 SYSPROG.SF16.MM070898.T1833P79          XE44  P
04/29/2008 10:30:19 04/29/2008 10:44:44 07/10/2008 10:36:27 202338  3490  0001 SYSPROG.SF16.MM071098.T1036P26          XE44  P
05/20/2008 15:42:00 05/20/2008 21:59:00 05/21/2008  0:05:00 PMA003  3490  DISK SYSPROG.SF16.MS052008.T1542I1           XE44  P
05/21/2008  1:00:10 05/21/2008 21:59:00 05/22/2008  0:05:00 115193  3490  0001 SYSTEM.USER.SMFDATA.D052198             XE44  P

CA                                              AUDIT TRAIL REPORT                           SMFD       PAGE     4
CA SMF Director     12.6                                                           SYSA      27 JUL yyyy  11:09:54

                                    INDEXED SMF DATA FOR CONFIGURATION XE44

------ FROM ------- ------- TO -------- ------ DUMPED ----- VOLID/  UNIT  FILE -------------- DATA SET NAME ---------- SID   X
04/17/2008 12:00:00 04/29/2008 10:44:44 06/29/2008 12:32:56 PMA003  3390  DISK SYSPROG.SF16.MM062998.T1232A74          XE44  A
04/29/2008 10:30:19 04/29/2008 10:44:44 07/08/2008 18:33:41 PMA003  3390  DISK SYSPROG.SF16.MM070898.T1833A06          XE44  A
04/29/2008 10:30:19 04/29/2008 10:44:44 07/10/2008 10:36:27 PMA003  3390  DISK SYSPROG.SF16.MM071098.T1036A73          XE44  A


CA                                              AUDIT TRAIL REPORT                           SMFD       PAGE     1
CA SMF Director     12.6                                                           SYSA      27 JUL yyyy  13:17:07

  LISTH  VOLUMES(FREE)    .

AVAILABLE SMF DUMP VOLUMES:
    SMF100 SMF101 SMF102 SMF103 SMF104 SMF105 SMF106


CAF2900I  RUN COMPLETE
```

OPTIONS

The OPTIONS control statement allows the specification of CA SMF Director program product options. The OPTIONS statement can be used as a standalone statement or can appear after the COMPILE CONFIGURATION STORE control statement and within the definition of a new configuration or the updating of an existing configuration. If you are not sure about the current definition of a configuration, use the LISTC SID (sidid) or LISTC All command to get a list of current parameter values for a configuration to use during the process to add OPTIONS values. When you are updating a configuration, any values that are not explicitly defined for a particular configuration will contain default values.

The standalone OPTIONS statement is valid only if there are existing configurations defined. Any values that are not explicitly coded in standalone OPTIONS statements will not be changed and will contain previous values.

Figure 14 shows example output from a successful update of OPTIONS values.

Statement Format:

OPTIONS

```
►►─ OPTIONS ─ SDAY(hhmm) ─ MAXLINES(nnn) ─┬─────────────────┬─────────────────────►
                                          ├─ NOAUTODEL ◄ ─┤
                                          └─ AUTODEL ─────┘

►─┬─────────────┬─ SITE(name) ─┬──────────────────────────────────┬─────────────►
  ├─ NODYNAM ◄ ─┤              ├─ PDEVN( ─┬─ dname ─┬─ ) ─────┤
  ├─ DYNAM ─────┤              │          └─ NONE ──┘         │
  └─ DYNAMVTS ──┘              └─ PSTORC( ─┬─ class ─┬─ ) ────┘
                                           └─ NONE ──┘

►─┬──────────────────────────────────┬─────────────────────────────────────────►
  ├─ ADEVN( ─┬─ dname ─┬─ ) ──────┤
  │          └─ NONE ──┘          │
  └─ ASTORC( ─┬─ class ─┬─ ) ─────┘
              └─ NONE ──┘

►─┬────────────────┬─┬──────────────────────┬──────────────────────────────────►
  └─ OVERLAP(n) ─┘ └─ MAXFILESONVOL(n) ─┘

►──COPIES( ─┬─ 1 ◄ ─┬─ ) ──┬─────────────┬───┬─────────────────┬──────────────►
            └─ 2 ───┘      ├─ NOCATLG ◄ ─┤   ├─ SCRATCH ───────┤
                           └─ CATLG ─────┘   └─ NOSCRATCH ◄ ─┘

►─┬─────────────────┬─┬─────────────────────┬─┬───────────────────┬────────────►
  ├─ EXPDT(yyyddd) ─┤ ├─ AEXPDT(yyyddd) ◄ ─┤ └─ TRTCH(value) ─┘
  └─ RETPD(nnnn) ───┘ └─ ARETPD(nnnn) ─────┘

►─┬──────────────────┬──┬────────────────────────────────┬─────────────────────►
  └─ ATRTCH(value) ─┘  └─ DAYSTACK( ─┬─ NO ◄ ─┬─ ) ─┘
                                     └─ YES ──┘
```

```
►─┬──────────────────────────────┬─┬───────────────────────────────┬─────────►
  └─ PDSNLARGE( ─┬─ NO ◄ ─┬─) ───┘ └─ ADSNLARGE( ─┬─ NO ◄ ─┬─ ) ───┘
                 └─ YES ──┘                       └─ YES ──┘

►─┬─ NOPREFIX ◄ ──────────┬────────┬─ NOAPREFIX ◄ ──────────┬────────────────►
  └─ PREFIX(prefix) ──────┘        └─ APREFIX(prefix) ──────┘

►─┬─ NOSERVICEDESK ◄ ──────────────┬────────────────────────── . ───────────►◄
  ├─ SERVICEDESK ──────────────────┤
  └─┤ SERVICEDESK parameters ├─────┘
```

SERVICEDESK parameters

```
►►─┬──────────────────────┬─┬───────────────────────┬────────────────────────►
   └─ SDHIGHSEVERITY(n) ──┘ └─ SDLOWSEVERITY(n) ────┘

►─┬─────────────────────────┬─┬─────────────────────────────────┬───────────►◄
  └─ SDURL( ─┬─ url ──┬─ ) ─┘ ├─ SDUSERID( ─┬─ user ID ─┬─ ) ───┤
             └─ NONE ─┘       │             └─ NONE ────┘       │
                              └─ SDPASSWORD(password) ──────────┘
```

Operands and Descriptions:

SDAY(hhmm)

Specifies the hour (hh) and minute (mm) that CA SMF Director is to treat as the logical start of a day. If this parameter is not coded, a day will be considered as starting at midnight (0000).

Note: The SDAY value is used to determine the starting and ending times for all EXTRACT and PRINT requests, except when the FROM, JFROM, TO or JTO parameter gives a specific starting or ending time. For example, if option SDAY(0600) is in effect, the command EXTRACT ALL FROM(040208) TO(040608) will cause the extraction of all SMF data from 0600 on April 2, 2008 up to—but not including—0600 on April 7,2008.

MAXLINES(nnn)

Specifies the maximum number of lines per page for CA SMF Director SYSPRINT output. "nnn" can be a numeric value from 0 or 10 through 255. The default is 0, which breaks pages every 60 lines.

AUTODEL|<u>NOAUTODEL</u>

Identifies if CA SMF Director should perform the auto-delete feature. If AUTODEL is specified, this product scans the SCDS file for any SMF inventory entries referencing history files that have expired at least three days earlier. If any entries exist, they are deleted from the SCDS and the system catalog entry is deleted as well, if it exists. The default is NOAUTODEL.

This operand is a global option only, but you can override this setting using the EXPDT/RETPD operands on a system-by-system or stream-by-stream basis on the DUMPOPTIONS or STREAMOPTIONS statements.

DYNAM|NODYNAM|DYNAMVTS

Controls whether or not this product will dynamically allocate history files when performing dump, extract, and print operations:

- DYNAM allows dynamic allocation of history files for all dumps and EXTRACT/PRINT operations.
- NODYNAM prevents dynamic allocation of history files under any circumstances. This is the default.
- DYNAMVTS allows dynamic allocation of input history files to EXTRACT and PRINT operations, but suppresses dynamic allocation of output history files from dump operations.

Note: If the DYNAM option is in effect and a history DD statement is coded, the parameters in the DD statement override the dynamic allocation options such as PSTORC, ASTORC, and so on. See the descriptions of the PSTORC and ASTORC operands for more information.

Note: The NODYNAM option may cause problems involving incorrect mount messages at installations that employ virtual tape. The DYNAMVTS option can be used to avoid this problem while maintaining control over allocation of output history files during dumps.

Note: If logstream or substream recorded SMF data will be managed with any defined configuration, DYNAM must be specified.

SITE(name)

'name' is an optional comment identifying the computing complex of up to 20 characters long and within quotes.

PDEVN(dname)

'dname' is the esoteric device name to be used when creating SMF history files in the primary pool. Up to eight characters can be specified. It is mutually exclusive with PSTORC. If no device name or SMS storage class is specified and the DYNAM option is enabled, the system default device name is used. This default name can be set in the active ALLOCxx member of the system PARMLIB. NONE can be used to reset the device name.

Note: If you are changing the PDEVN parameter from one tape unit name to another, from tape to DASD, or from DASD to tape, you must immediately run a TAPEINIT command against every SID defined in your SCDS.

If using DASD for history files, this parameter must only be used for non-SMS managed volumes.

PSTORC(class)

'class' is the IBM DFSMS storage class to be used when creating SMF history files in the primary pool. Up to eight characters can be specified. It is mutually exclusive with the PDEVN keyword. NONE can be used to reset the storage class.

If using DASD for history files, this operand will allocate files only to SMS-managed volumes. Note that some issues may arise with DASD history file allocations based on how SMS is set up at your data center. For DASD history allocations with the DYNAM option set, there must still be a temporary history file allocated in the JCL that indicates how large the history file should be. SMS may not allow the allocation on the storage class indicated in the PSTORC operand and will indicate an allocation error for the history file, causing the dump process to end prematurely.

To avoid this issue, here are some remedies you can use:

- Include a data set name on the history DD statement. This can be any valid data set name that would be allocated in the storage class indicated in the PSTORC operand.

  Note: This is the simplest remedy and the one that CA recommends.

- Alter the ACS routines in the SMS DASD allocation to assign no class to temporary data sets, where the fourth node of the name is the same as the dump process JCL job name or procedure name.

  Note: These changes are only for DASD history files allocated with SMS. Non-DASD and non-SMS allocations will not be affected by this.

ADEVN(dname)

'dname' is the esoteric device name to be used when creating SMF history files in the alternate pool. Up to eight characters can be specified. It is mutually exclusive with ASTORC. If no device name or SMS storage class is specified and the DYNAM option is enabled, the default device name used will be the device name used for the primary pool. NONE can be used to reset the device name.

Note: If you are changing the ADEVN parameter from one tape unit name to another, from tape to DASD, or from DASD to tape, you must immediately run a TAPEINIT command against every SID defined in your SCDS.

If using DASD for history files, this parameter must only be used for non-SMS managed volumes.

ASTORC(class)

'class' is the IBM DFSMS storage class to be used when creating SMF history files in the alternate pool. Up to eight characters can be specified. It is mutually exclusive with the ADEVN keyword. NONE can be used to reset the storage class.

If using DASD for history files, this operand will allocate files only to SMS-managed volumes. Note that some issues may arise with DASD history file allocations based on how SMS is set up at your data center. For DASD history allocations with the DYNAM option set, there must still be a temporary history file allocated in the JCL that indicates how large the history file should be. SMS may not allow the allocation on the storage class indicated in the ASTORC operand and will indicate an allocation error for the history file, causing the dump process to end prematurely.

To avoid this issue, here are some remedies you can use:

- Include a data set name on the history DD statement. This can be any valid data set name that would be allocated in the storage class indicated in the ASTORC operand.

  Note: This is the simplest remedy and the one that CA recommends.

- Alter the ACS routines in the SMS DASD allocation to assign no class to temporary data sets, where the fourth node of the name is the same as the dump process JCL job name or procedure name.

  Note: These changes are only for DASD history files allocated with SMS. Non-DASD and non-SMS allocations will not be affected by this.

OVERLAP(n)

'n' can be from 0 through 30 seconds, which is the maximum allowable overlap of SMF dump history files before CA SMF Director considers a file to be in an 'overlap' condition. The default is zero.

For example, assume that two SMF data history files contain these date ranges:

- File A's date range is 09/08/00 08:00:00 - 09/08/00 12:15:45
- File B's date range is 09/08/00 12:15:43 - 09/08/00 17:30:55

If OVERLAP(0) was specified, File B would contain overlapped data with File A. If OVERLAP(3) was specified, neither of these files would be considered to contain overlapped data.

Note: We recommend that you observe the overlap conditions that occur and analyze them to see if they are truly overlaps. They may just be records written with timestamps in error. This is especially true with non-IBM SMF records that may be created with timestamps that do not account for leap seconds.

MAXFILESONVOL(n)

Indicates the maximum number of history files that can be written to a single tape volume. With some devices that use very large tapes, performance issues can occur if a large number of files are written to the tape. Using this operand, the number of files that are placed on one tape can be limited so that the performance issues are avoided. The value of the operand can be set to any integer value from 1 to 32760. If this operand is not specified, a default value of 300 is set.

COPIES(1|2)

Specifies whether or not CA SMF Director should create a duplicate copy of the dumped SMF data (duplexing) if the COPIES operand is not specified for the system (with DUMPOPTIONS) or stream (with STREAMOPTIONS). If 1 is specified, no duplexing is performed. If 2 is specified, duplexing of the history file is performed. A duplicate index entry is also created on the SCDS file to maintain an inventory of the SMF data for the duplex file. Duplex files are placed in the alternate archive. The default is COPIES(1), but it is recommended that COPIES(2) be set for all SMF data that needs to be kept on a long-term basis.

CATLG|<u>NOCATLG</u>

Indicates whether CA SMF Director should catalog the history file data set, creating an entry within the system catalog, or not. The default is NOCATLG. This value can be overridden on a system basis using DUMPOPTIONS or on a stream basis using STREAMOPTIONS.

Note: If the PSTORC or ASTORC option is in effect, SMS may override the NOCATLG/CATLG specification.

<u>NOSCRATCH</u>|SCRATCH

Specifies whether CA SMF Director should request specific or nonspecific magnetic tape volumes when creating history files:

- If SCRATCH is specified, CA SMF Director performs a nonspecific mount request for a scratch tape when a new tape volume is needed for a history file for all configurations that do not have a specific setting of NOSCRATCH.
- If NOSCRATCH is specified, CA SMF Director selects a volume serial from the predefined pool of volumes when a new tape volume is needed for a history file unless the configuration has a specific setting of SCRATCH. See the DUMPTAPES control statement for more information on setting up the predefined pool of tape volumes.

The value specified here is used for all configurations if UPSCRATCH is specified on the DUMPOPTIONS statement.

Note: If tape management products such as CA 1, CA Vtape Virtual Tape System, or CA TLMS are used, SCRATCH is recommended.

Note: If logstreams or substreams are going to be dumped in a configuration, SCRATCH must be set for that configuration. We recommend that the SCRATCH operand be set in each configuration if streams are going to be used.

EXPDT(yyyddd)|RETPD(nnnn)

Specifies the expiration date or retention period of the history file data sets within the primary pool. If the expiration date is specified, it must be in Julian date format. The default is EXPDT(0) or RETPD(0), which causes CA SMF Director not to set an expiration date for the history file.

See Control Statement Language Rules (see page 37) for a description of two special expiration dates: 099000 and 099366.

We recommend that expiration dates or retention periods be used, since it increases the integrity of the SMF data. If the AUTODEL feature is desired, setting expiration dates or retention periods is required.

Note: If the PSTORC option is in effect, SMS may override the specified expiration date or retention period.

AEXPDT(yyyddd)|ARETPD(nnnn)

Specifies the expiration date or retention period of the history file data sets within the alternate pool for configurations that do not have a specific EXPDT or RETPD. If the expiration date is specified, it must be in Julian date format. The default is EXPDT(0) or RETPD(0), which causes CA SMF Director not to set an expiration date for the history file.

See Control Statement Language Rules (see page 37) for a description of two special expiration dates: 099000 and 099366.

We recommend that expiration dates or retention periods be used, since it increases the integrity of the SMF data. If the AUTODEL feature is desired, setting expiration dates or retention periods is required.

Note: If the PSTORC option is in effect, SMS may override the specified expiration date or retention period.

TRTCH(value)

ATRTCH(value)

Lets you use the z/OS JCL TRTCH parameter on dynamically allocated history files in CA SMF Director. The values of the TRTCH and ATRTCH operands are the same as the values allowed in z/OS JCL when writing an output file to tape. TRTCH is used to set the recording technique for the primary history file, while ATRTCH is used to set the TRTCH for the alternate history file.

Valid values are as follows:

NOCOMP

For cartridge tapes, writes the history file without data compression.

COMP

For cartridge tapes, writes the history file with data compression.

C

Performs data conversion for history files on reel tapes.

E

Writes history files to reel tape with even parity.

ET

Writes history files to reel tape with even parity and perform EBCDIC translation.

T

Writes history files to reel tape and perform EBCDIC translation.

Note: Although CA SMF Director does support TRTCH options C, E, ET, and T, they are not recommended for history files.

Note: If SMS is used in dynamic allocation for history files, the TRTCH setting for that history file can be changed by SMS.

If neither TRTCH nor ATRTCH is specified, the system defaults are used.

DAYSTACK(NO|YES)

Specifies whether the daystacking feature should be turned on or off. The default is NO, which turns daystacking off. See Features for a description of the daystacking feature.

PDSNLARGE(NO|YES)

Allocates the primary history file with the DSNTYPE=LARGE operand. This operand is only used for DASD history files when the DYNAM operand is specified on the OPTIONS statement; otherwise it is ignored.

ADSNLARGE(NO|YES)

Allocates the alternate history file with the DSNTYPE=LARGE operand. This operand is only used for DASD history files when the DYNAM operand is specified on the OPTIONS statement; otherwise it is ignored.

NOPREFIX|PREFIX(prefix)

Specifies the high level qualifiers to be used for the primary history file data set names. If NOPREFIX is specified, CA SMF Director uses SYS1.SMFDATA, which is the default.

See Attributes of History Files for more details. The prefix can be up to 26 characters.

NOAPREFIX|APREFIX(prefix)

Specifies the high level qualifiers to be used for the primary history file data set names if nothing is specified for a specific stream or system. If NOAPREFIX is specified, the value of SYS1.SMFDATA is used for systems that do not have a PREFIX or APREFIX specified.

See Attributes of History Files for more details. The prefix can be up to 26 characters.

NOSERVICEDESK|SERVICEDESK

SERVICEDESK specifies that an attempt should be made to open a Service Desk request in case of an error during SMP dump processing. NOSERVICEDESK is the default.

SDHIGHSEVERITY

Specifies the severity level of high Service Desk requests that CA SMF Director will open. The value can be set to any value between 1 and 4. The value itself also becomes the lower bound for the value of SDLOWSEVERITY. If this is not specified and the SERVICEDESK operand is specified, the value defaults to 2.

SDLOWSEVERITY

Specifies the severity level of low Service Desk requests that CA SMF Director will open. The value can be set to any value between the value of SDHIGHSEVERITY and 5. The default value, if the operand is not specified is 4.

Note: If only one severity level is desired for all CA SMF Director, the value of SDHIGHSEVERITY and SDLOWSEVERITY can be set to the same value from 1 to 4.

SDURL

Specifies an alternate URL for the Service Desk Web Service if the default Service Desk Web Service in the CA Common Services SOAP Client Service (CAISDI/soap) address space does not indicate the desired Service Desk instance. NONE can be used to reset the URL.

Note: The URL value is case sensitive and must be enclosed in single quotes because it will contain special characters.

SDUSERID

Specifies an alternate user ID for the Service Desk Web Service if the default value set in the CA Common Services SOAP Client Service (CAISDI/soap) address space is not the correct User ID for the Service Desk. If the SDUSERID operand is coded, SDPASSWORD is required. NONE can be used to reset the user ID. If NONE is used, the alternate password for Service Desk is also reset, and the SDPASSWORD is ignored.

SDPASSWORD

Specifies an alternate password for the Service Desk Web Service if the default value set in the CA Common Services SOAP Client Service (CAISDI/soap) address space. If the SDUSERID operand is coded, SDPASSWORD is required; otherwise, it is ignored.

Note: The operand is case-sensitive. If the operand contains special characters, it must be contained in single quotes.

Sample OPTIONS Statement:

```
OPTIONS AUTODEL DYNAM   PSTORC(TAPESILO)
```

OPTIONS statement output:

```
CA                                              AUDIT TRAIL REPORT                              SMFD       PAGE     1
CA SMF Director     12.6                                                                SYSA      27 JUL yyyy  13:23:45

  COMPILE CONFIGURATION STORE.
  OPTIONS AUTODEL DYNAM  PSTORC(TAPESILO) .
   BEGIN CONFIG(001) SID(XAT1) NAME('C.A. HERNDON VA') .
    DUMPOPTIONS COPIES(1) CATLG SCRATCH   NOVERIFY
      RETPD(1095) PREFIX(ELLPH01.SF16)
     ARETPD(1461)
   END CONFIG(001)   .

CAF1301I  CONFIGURATION DEFINITION XAT1 REPLACED

  COMPILE CONFIGURATION STORE.
    BEGIN CONFIG(002) SID(XAD1) NAME('C.A. ISLANDIA TSO1') .
      DUMPOPTIONS COPIES(2) CATLG NOSCRATCH NOVERIFY
        ARETPD(1461) RETPD(1095)
        PREFIX(ELLPH01.SF16).
    END CONFIG(002)    .

CAF1301I  CONFIGURATION DEFINITION XAD1 REPLACED

CAF2900I  RUN COMPLETE
```

SOURCE

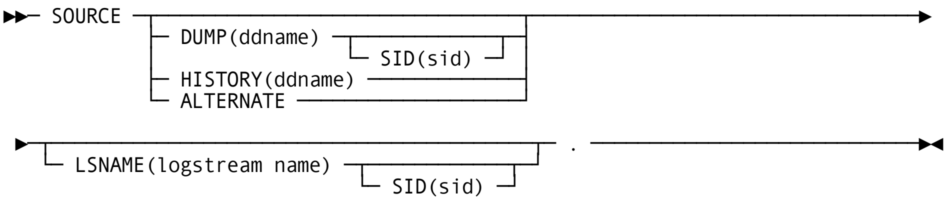
This control statement lets you redefine the normal source of the data that CA SMF Director uses in its processing. It has three forms:

- SOURCE DUMP lets you read in a data set containing SMF data from a single SID as though it were a MAN file or a logstream and feed the data into CA SMF Director.
- SOURCE HISTORY lets you run an extract directly from a specified CA SMF Director history file instead of retrieving it through the SCDS. This form causes the DYNAM option to be disabled.
- SOURCE ALTERNATE tells CA SMF Director to use the alternate index for all subsequent EXTRACT and PRINT requests in the same execution.

You may specify SOURCE only once per run. The control statements can appear anywhere in the statement input stream.

Statement Format:

SOURCE



Operands and Descriptions:

DUMP(ddname)

Specifies a file "(ddname)" to be used as input to the dump process. The normal system SMF data sets are not read, and only data from this file is processed. The file "(ddname)" cannot begin with "HIST."

When using a "SOURCE DUMP(ddname)" statement, SMF record types 2 and 3 will be ignored. If these records were to be processed, they would cause the SMF data to appear out of sequence. They are header and trailer records generated by the IBM SMFDUMP program and written to the dumped file.

HISTORY(ddname)

Specifies input file "(ddname)" for EXTRACT and PRINT requests, bypassing automatic retrieval of data from the history data sets.

"(ddname)" must be a 1 to 8-character name that corresponds to a DD statement in the execution JCL for this run.

ALTERNATE

Specifies that the alternate index is to be accessed for files needed to satisfy any EXTRACT or PRINT requests.

LSNAME(logstream name)

Specifies which logstreams are to be processed during a dump from an archive. If SID is specified, the logstreams must be within the SID.

SID(SID)

Specifies the SMF ID of records that are to be processed during a dump from a source file. This operand is primarily intended to be used when you are source dumping archival data from SMF logstreams because an SMF logstream can contain records from more than one system when the logstreams are backed by a list structure in a coupling facility.

The default SID is the SMF ID of the first record in the source SMF file. This operand is valid only when the DUMP and LSNAME operands are present.

Sample SOURCE Statement:

```
SOURCE ALTERNATE.
```

SOURCE statement output:

```
CA                                              AUDIT TRAIL REPORT                          SMFD       PAGE     1
CA SMF Director     12.6                                                            SYSA      27 JUL yyyy  11:45:24

 SOURCE ALTERNATE.

CAF2800I  SOURCE OPTION SET

 PRINT SID(XE44) FROM(041698 131500)  TO(041698 132000) COUNT(5) .
CAF1010I  REQUEST ACCEPTED


CAF3200I  SMF RECORD TYPE 21 BUFFER DATE 04/16/2008 ( 98/106) TIME 13:16:44
         0000    1E150048 F1980098 106FE7C5 F4F40036   F3F0F5F0 F9F00752 78048081 00000033    *....1.....XE44..305090..........*
         0020    00000000 00000000 00000000 00000000   00000000 00000100 00000000 00000000    *................................*
         0040    00000000                                                                     *....                            *

CAF3200I  SMF RECORD TYPE 21 BUFFER DATE 04/16/2008 ( 98/106) TIME 13:16:56
         0000    1E150048 F6700098 106FE7C5 F4F40036   F3F0F3F3 F0F2074D 78048081 00000049    *....6.....XE44..303302..........*
         0020    00000000 00000000 00000000 00000000   00000000 00037800 00000000 00000000    *................................*
         0040    00000000                                                                     *....                            *
CAF3100I  2 RECORDS WRITTEN TO SYSPRINT FILE

CAF2900I  RUN COMPLETE
```

SPLIT

This control statement, used only during the DUMP process, indicates if any records should be written to additional files, called split files, that can then be used as input to other applications without the need for any intermediate extractions. A split file has the same attributes as an EXTRACT file, and can be processed the same way as an extract. The actual SPLIT control statement must follow a DUMP statement in the control statement input file or else it will be in error.

Since the SPLIT command runs during a critical process, the dump process, there are no chances taken when creating split files. If any sort of error occurs when a split file is being written (abend or otherwise), the split file that was being dealt with at the time will be disabled and the dump process will continue without interruption. If it cannot be determined which split file was being dealt with, all of the split files will be deactivated and the dump process will continue without interruption.

If there are errors, diagnostic information will be produced in the dump process output which can be used to correct the problem in future dump process executions. In addition, if CA Service Desk is installed and CA SMF Director is set up to open Service Desk requests, an attempt will be made to open a request.

If a split file was disabled, the output can be recreated by using the EXTRACT process after the dump has completed.

Statement Format:

SPLIT

```
►►─ SPLIT ─┬─ SID(smfid,smfid...) ─┬─ TODD(ddname) ─┬───────────────┬──────────►
           └─ ALL ─────────────────┘                └─ EXIT(module) ┘

►─┬──────────────────┬──────────────────────────────────────────────────────────►
  ├─ SELECT(list) ───┤
  └─ EXCLUDE(list) ──┘

►─┬─────────────────────────────────────────────────────────────────┬──────────►
  └─ INDEXDSN(dsname) ─┬─────────────────┬─┬──────────────────────┬─┘
                       └─ INDEXLIM(nn) ──┘ └─ INDEXDUP(ddname) ───┘

►─┬──────────────────────────────────────────┬─────────────────────────────────►
  ├─ WHEN(component,test,value,test,value...) ─┤
  ├─ WHEN(component,test,value,test,value...) ─┤
  ├─ WHEN(component,test,value,test,value...) ─┤
  ├─ WHEN(component,test,value,test,value...) ─┤
  └─ WHEN(component,test,value,test,value...) ─┘

►─┬──────────────────────────────┬─────────────────────────────────────────────►
  └─ EXPAND110( ─┬─ NO ◄ ─┬─ ) ──┘
                 └─ YES ──┘

►─┬─────────────────────────────┬──────────────────────────────────────────────►
  └─ TIMEFILL( ─┬─ NO ◄ ─┬─ ) ──┘
                └─ YES ──┘

►─┬───────────────────────────────┬────────────────────────────────────────────►
  └─ TIMECLOSE( ─┬─ NO ◄ ──┬─ ) ──┘
                 ├─ YES ───┤
                 └─ FORCE ─┘

►─┬────────────────────────────────────────────────────────┬─ . ─────────────►◄
  └─ TIME(hhmm,hhmm,hhmm,hhmm) ─┤ TIME Parameters ├────────┘
```

Expansion of TIME Parameters

```
├─ TIMEGDG(gdgname) ────────────────────────────────────────────────────────────►

►─┬─ TIMESTORC(SMSstorageclass) ─┬─────────────────────────┬───────────┬───────►
  │                              ├─ TIMEVOL(volser) ───────┤           │
  │                              ├─ TIMESPACE(type,pri,sec) ┤          │
  │                              ├─ TIMEEXPDT(yyyyddd) ────┤           │
  │                              └─ TIMERETPD(nnnn) ───────┘           │
  └─ TIMEUNIT(unit) ─┬─────────────────────────┬───────────────────────┘
                     ├─ TIMEVOL(volser) ───────┤
                     ├─ TIMESPACE(type,pri,sec) ┤
                     ├─ TIMEEXPDT(yyyyddd) ────┤
                     └─ TIMERETPD(nnnn) ───────┘

►─┬───────────────────────────────────────────────────┬────────────────────────►
  └─ TIMEBLKSIZE(blksize) ─┬───────────────────────┬──┘
                           └─ TIMELRECL(lrecl) ────┘

►─┬──────────────────────┬─────────────────────────────────────────────────────►
  └─ TIMEVOLCOUNT(nnn) ──┘

►─┬──────────────────────────────┬─────────────────────────────────────────────►
  └─ TIMERECFM( ─┬─ VB ◄ ─┬─ ) ──┘
                 └─ VBS ──┘

►─┬──────────────────────────────────┬─────────────────────────────────────────┤
  └─ TIMEDSNLARGE( ─┬─ NO ◄ ─┬─ ) ───┘
                    └─ YES ──┘
```

Operands and Descriptions:

SID|ALL

Identifies the configurations that the SPLIT statement applies to. Specifying ALL indicates that the SPLIT statement will be applied to all configurations being processed. The SID operand limits the SPLIT statement to the SMF IDs listed. Up to 20 values can be listed. Wildcarding with an asterisk is allowed, and you can mix wildcard and non-wildcard values. For example, to limit a SPLIT statement to only SMF IDs SYSA and SYSB, specify SID(SYSA,SYSB). To apply the SPLIT statement only to production systems that have SMF IDs beginning with PR, specify SID(PR**).

TODD(ddname)

Indicates the ddname of the split file, the file that will receive the records that are written out. To avoid potential conflicts with existing and reserved ddnames, we strongly recommend that the ddname not begin with SYS or CA.

Note: This value is also placed in the split file index and can be used as a logical identifier for split files that are related because they are used in the same definition.

If the TIME operand is not coded in the SPLIT statement, then this ddname must be in the JCL and the file allocated. If the TIME operand is indicated in the SPLIT statement, then this ddname must not be in the JCL since it will be used in dynamic allocation of the GDG members indicated in the TIMEGDG operand.

EXIT(module)

Specifies the load module entry point name for a split exit module. See the *Systems Programmer Guide* for specifications for writing a SPLIT exit. The module must be located in the STEPLIB or JOBLIB concatenation for the dump process, in the system linklist, or in LPA.

SELECT|EXCLUDE

Either selects only certain record types for processing or excludes certain record types from processing.

Note: These operands are optional, but if neither is coded, all SMF records being dumped will be processed. Therefore, the split file will contain every record in the dump.

The list of records may include any valid SMF record types and subtypes from 0 through 255. They may be coded as individual record types separated by commas or as a range separated by a colon. Subtypes, if specified, must be coded in three decimal places, for example 30.004.

For example, SELECT(30.001:30.005) results in the splitting out of SMF record type 30, subtypes 1 through 5. EXCLUDE(99.02,99.1) causes the splitting out of all SMF being dumped **except for** type 99, subtypes 20 and 100.

Some SMF records have subtypes of zero. To specify subtype zero, a value of zero must be placed to the right of the decimal point. For example, SELECT(116.000) indicates SMF Records of type 116, Subtype 0, while SELECT(116) indicates all type 116 records.

INDEXDSN(dsname)

Indicates the DSN of an index data set for split files generated by the SPLIT control statement. If it is not specified, no split file index records will be generated. The file must be created and cataloged with the following DCB attributes:

```
DSORG=PS
RECFM=FB or F
LRECL=192
BLKSIZE=192 or a multiple of 192
```

In the case of RECFM=F, the BLKSIZE must be 192; otherwise the BLKSIZE must be a multiple of 192. The data set must also be created in advance. If the file is not found, the SPLIT statement will be disabled.

INDEXDUP(ddname)

This optional operand should only be coded if INDEXDSN is coded. It makes a copy of the split file index and writes it to the ddname specified. We strongly recommend that the index be copied to a duplicate file if the split file index is to be used for processing other than within CA SMF Director because CA SMF Director needs exclusive control of the real index in order to process it properly.

The attributes of the duplicate index file are the same as for the index file (see INDEXDSN above).

Note: If the dump process cannot get access to the duplicate file, the duplicate file will not be produced. However, the dump process will complete, and the original index will be updated. If necessary, a duplicate of the split file index can be created by copying the index with IEBGENER or a similar utility.

To ensure that neither the dump process nor other applications are delayed due to data set allocation, the duplicate index data set must be allocated with DISP=SHR. Serialization of the duplicate index will be provided by CA SMF Director. When the CA SMF Director dump process updates the duplicate index file, it issues an exclusive SYSTEMS level ENQ with the following characteristics:

- QNAME: CAIMSMF
- RNAME:
- DUPINDEX: data set name of the duplicate index, padded with spaces
- RNAME Length: 53

Applications that read the duplicate index should issue a shared SYSTEMS ENQ for the QNAME and RNAME listed above before attempting to read the duplicate index. The application should also DEQ the QNAME/RNAME immediately after processing the duplicate index. Note that all CA software that accesses CA SMF Director duplicate indexes will honor this ENQ.

INDEXLIM(number)

This optional operand should only be coded if INDEXDSN is coded. It indicates the maximum size of the split file index indicated by INDEXDSN. If the number of records in the split file exceeds this limit, the oldest records in the index will be dropped until the index has as many records as the value indicates. If omitted, the index file will have no limit on the number of records. (Index records are not particularly large; an index file of one 3390 cylinder will hold 4,350 index records at half-track blocking.)

WHEN(component,test,value,test,value...)

Lets you reduce the number of records based on identification characteristics contained within the component SMF data. Components that are supported are as follows:

CICS

Type 110 records, keyed on CICS APPL ID

DB2

Type 100, 101,and 102 records, keyed on DB2 subsystem name

DSN

Type 14, 15, 17, 18, 61, 62, 64, 65, and 66 records, keyed on the DSN

JOB

Type 6, 26, and 30 records, keyed on the job name

SECR

Type 80 records, keyed on the response code

SECU

Type 80 records, keyed on the user ID

TSO

Type 32 records, keyed on the user ID

Valid values for *test* are as follows:

EQ

Equal

NE

Not Equal

GE

Greater than or equal

GT

Greater than

LE

Less than or equal

LT

Less than

The *value* parameter is a text field that represents an identifying characteristic in the SMF record. For CICS, it represents the Application ID (APPLID), an eight-byte field. For DB2, it represents the DB2 subsystem name, a four- byte field.

Up to 4 *test/value* pairs can be listed in a WHEN statement. For a record to be selected it must pass all of the tests within the WHEN statement. For mutually exclusive type tests, multiple WHEN statements for the same component can be used. A SPLIT statement may contain up to 5 WHEN statements.

An asterisk in the value parameter represents a wildcard match. If the asterisk is at the beginning or in the middle of the value string, it represents a single character. If the asterisk is at the end, it represents one or more characters.

Note: Even if a component is mentioned in a WHEN statement, the SMF record types represented by the component must be indicated in a SELECT operand or left out of an EXCLUDE operand; otherwise the records will not be written to the split file. For example if a WHEN clause for CICS records is present in a SPLIT statement, the SELECT statement indicated should select type 110 records (or conversely an EXCLUDE statement should not indicate type 110).

EXPAND110(YES|NO)

Indicates whether or not compressed SMF type 110 records should be expanded before they are written to the split file. If the value of the operand is YES, CA SMF Director will expand any compressed type 110 records that are to be written to the output file. The default value is NO, meaning that no expansion of type 110 records will be performed.

TIME(hhmm,hhmm,hhmm,hhmm)

Indicates a time of day when the split file defined by this SPLIT statement should be closed and marked complete and another split file should be opened to contain further records. This lets you break up the created split files at dump time on logical boundaries that are not determined by the times SMF is forced to dump the contents of a MAN file. If the TIME operand is being used, the split files that are written to must be part of a generation data group (GDG).

Up to 4 times for a switch may be indicated.

Note: There is no automatic switch at midnight (indicated by 0000). If a switch of split files is to take place at midnight, 0000 must be coded as the first value in the TIME operand.

The values in the TIME operand must be coded in sequential order.

If the TIME operand is coded, the TIMEGDG operand must also be coded. Either the TIMESTORC or TIMEUNIT operand must also be coded. In addition, if the output split files are to be on disk, the TIMESPACE operand must be coded.

Note: Because of the way SPLIT files are managed, there are special considerations for records that are recorded on specific intervals when using TIME-based SPLIT files. A TIME-based SPLIT file can only be made ready for processing if it is in the Complete state. It is only Complete if it is closed during the dump process before end-of-input is reached for the SMF data being archived. (The last TIME-based SPLIT file is left in the Incomplete state, and is added to when the next dump process begins.)

To ensure that interval-based SMF data is available as soon as possible for processing, the dump process must begin at least one minute after the values in the TIME operand plus the value of INTVAL in your active SMFPRMxx member. For example, if the TIME operand is coded as TIME(0000,1200) and your active SMFPRMxx member has INTVAL(15) set, dumps should be scheduled for 0016 and 1216 (or later) to ensure that there are records in the data to be archived that are past the current TIME boundary being managed by CA SMF Director. By having records past the current TIME boundary, the TIME-based SPLIT file can be set to Complete and be ready for processing.

TIMEFILL(NO|YES)

Indicates whether CA SMF Director should create empty split files in a TIME-based split file GDG if there are no records for an interval. The default value is NO. If set to YES, split file processing will create empty files and fill in the relative generations in the split file GDG for empty intervals. Up to four consecutive empty intervals will be produced. Setting TIMEFILL to YES is recommended for split files that are being produced for applications that expect the production of input SMF files at regular intervals even if there is no data in the file.

Note: Files that are added to the split file GDG when YES is specified will have a status of EMPTY (E) in the split file index, meaning these files are complete and contain no records now or in the future.

TIMECLOSE(NO|YES|FORCE)

Indicates whether a file in a TIME-based split file can be marked closed and made available for application processing.

If the value defaults to NO, the split file generation for an interval can only be marked as complete if there is another record in the current dump for the split file that is past the current interval.

If YES is specified, a check is made to determine if the time values of all the SMF records in the dump have gone past the end time of the interval defined in the SPLIT statement.

- If the split file is defined only for system record types (0 through 127) the highest time value found for all system record types will be checked against the end of the current interval. If it is found that the highest system record timestamp is past the end of the interval, the current split file will be marked as complete, and a new generation of the split file will be allocated.
- If the split file is defined with user records, or with both system and user records, the time value used to check for the end of the interval will have the value of the OVERLAP option added to it. This means that for the file to be closed, the highest IBM timestamp must be higher than the starting time of the next interval defined in the TIME operand plus the value of OVERLAP.

If FORCE is specified, no time check is made and the current split file being written to at the end of the dump will be marked complete, regardless of the times found in the records. A new generation will be added to the split file GDG.

Note: The following applies if TIMECLOSE is set to YES or FORCE:

- The split file definition must have the INDEXDSN operand pointing to a valid split file index.
- Split files that are created to begin the next interval will have an incomplete (I) status in the split file index. Therefore, these files should not be processed by any application that uses the output split files in the GDG.

TIMEGDG(gdgname)

This operand is required if the TIME operand is coded; otherwise it is ignored. It must contain the name of a predefined generation data group (GDG). There is no requirement that the GDG have any files in it when it is first created. Up to 26 characters may be used in the name, as the value of ".SIDssss" will be appended to the GDG name, ensuring that data with different SMF IDs will not be mixed in the same GDG.

TIMESTORC(SMSstorageclass)

This operand is required if SMS is to be used for allocation of new split files during the dump process. The value should be a predefined SMS storage class.

Note: Both TIMESTORC and TIMEUNIT can be specified, but only one is required.

TIMEUNIT(unit)

This operand is required if a non-SMS allocation is to be made for new split files during the dump process. The value should be a valid value for the UNIT= parameter of a JCL DD statement.

Note: Both TIMESTORC and TIMEUNIT can be specified, but only one is required.

TIMEVOL(volser)

This optional operand is used in conjunction with the TIMEUNIT operand to indicate which specific volume should contain the new split file. It can also be used with TIMESTORC if the storage class indicated by TIMESTORC allows the selection of volumes.

TIMESPACE(type,pri,sec)

This operand is required if the new split files are to be allocated on DASD. This operand requires 2 or 3 values, the first of which must be either TRK (for track allocations) or CYL (for cylinder allocations). The second value is the the primary quantity of tracks or cylinders. The optional third value is the secondary quantity. This parameter should not be used if the split files are being written to tape. It is for disk files only.

TIMEEXPDT(yyyyddd|yyddd)

Sets a specific expiration date for a dynamically allocated split file that is written to tape. The format can be specified as yyyyddd, which is the full four-digit year followed by the day of the year, or as yyddd, which indicates the last two digits of the year followed by the day of the year. If the yyddd format is used, the year is set to be between 1960 and 2059.

See Control Statement Language Rules (see page 37) for a description of two special expiration dates: 099000 and 099366.

Note: TIMEEXPDT and TIMERETPD are mutually exclusive.

TIMERETPD(days)

Can be used if split files are to be written on tape. This will set the retention period for new, dynamically allocated split files.

Note: TIMEEXPDT and TIMERETPD are mutually exclusive.

TIMEBLKSIZE(blksize)

This optional operand indicates the maximum block size for the allocated split files. The default value is four more than the value of TIMELRECL. If neither TIMELRECL and TIMEBLKSIZE are indicated, the value is set to 32760. If TIMELRECL is indicated, then TIMEBLKSIZE must also be present in the SPLIT statement.

TIMELRECL(lrecl)

This is the maximum logical record length for the SMF records that will be written to the split file. This value is optional and should only be used if it is certain that all of the records that qualify to be part of the split file will be less than the SMF maximum of 32756. The default value is 32756.

TIMEVOLCOUNT(count)

This optional operand can be used to indicate the number of volumes a split file will use when it is created. The value must be an integer between 1 and 255.

TIMERECFM(VB|VBS)

Indicates which record format a split file will have when it is defined with the TIME operand. The values are the same as in the RECFM operand on the JCL DD statement, although only the VB and VBS values are supported. The default value is VB.

TIMEDSNLARGE(NO|YES)

Indicates whether a DASD split file that is defined with the TIME operand should be allocated with the DSNTYPE=LARGE attribute. If the split file is allocated to tape, this operand is ignored. The default value is NO.

Sample SPLIT Statements:

Split off all of the SMF Type 30 records to the file allocated to ddname SMF30X1 for dumps on all systems:

```
SOURCE DUMP(SMFDATA).
DUMP.
SPLIT ALL SELECT(30) TODD(SMF30X1).
```

Split off all SMF Type 30, Subtypes 1 through 5 only to ddname SMF30X2 and keep track of the last 10 data sets written:

```
SOURCE DUMP(SMFDATA).
DUMP.
SPLIT ALL SELECT(30.001:30.005) TODD(SMF30X2)
   INDEXDSN(CAI.SMFSPLIT.INDEX) INDEXLIM(10).
```

Split all production CICS SMF 110 records to the SMFCICS file (in this example, all production CICS APPLIDs begin with "PROD") and also dump all of the RMF data to the RMFDATA file, but only for system SYSP:

```
SOURCE DUMP(SMFDATA).
DUMP.
SPLIT ALL SELECT(110) TODD(SMFCICS)
  WHEN(CICS,EQ,PROD*).
SPLIT SID(SYSP) SELECT(70:79) TODD(RMFDATA).
```

Split all production and test CICS SMF 110 records to the SMFCICS file. In this example, proudction CICS APPLIDs begin with P and the test CICS APPLIDs being with T:

```
SOURCE DUMP(SMFDATA).
DUMP.
SPLIT ALL SELECT(110) TODD(SMFCICS)
  WHEN(CICS,EQ,P*)
  WHEN(CICS,EQ,T*).
```

Split all user SMF records to the SMFUSER file and use exit HYD83SMF to further filter the data:

```
SOURCE DUMP(SMFDATA).
DUMP.
SPLIT ALL EXCLUDE(0:127) TODD(SMFUSER) EXIT(HYD83SMF).
```

Split all SMF Type 30 records out to a group of split files that are created every 6 hours and are part of a GDG. These should be allocated on a 3390 disk pack and should be 20 cylinders big with a secondary allocation of 5 cylinders. The index file that manages the split file information is CAI.SMFSPLIT.INDEX. A copy of the Index should be written to the APPLIDX DD.

```
SOURCE DUMP(SMFDATA).
DUMP.
SPLIT ALL SELECT(30) TODD(TYPE30S) TIME(0000,0600,1200,1800)
  TIMEGDG(ACCT.SMF30.RAWFILE) TIMEUNIT(3390)
  TIMESPACE(CYL,20,5) INDEXDSN(CAI.SMFSPLIT.INDEX)
  INDEXDUP(APPLIDX).
```

Restrictions on Use

The SPLIT statement must follow a DUMP statement in the control statement file. SPLIT commands are not allowed at any other time other than DUMP time. If records are needed after dumping, the EXTRACT control statement can be used to create equivalent files. This is also true if an error occurs in split file creation during the dump process.

Split File Processing

The split files themselves have the following characteristics:

```
RECFM=VB or VBS
LRECL= up to 32756 for RECFM=VB; up to 32760 for RECFM=VBS
BLKSIZE= up to 32760 (must be at least 4 more than LRECL when RECFM=VB)
DSORG=PS
```

The default LRECL and BLKSIZE for the split files are 32756 and 32760 respectively. A lower LRECL and BLKSIZE can be specified, but this should be done only if it is certain that the all of the records that will be written to the split file will have a record length less than or equal to the LRECL. If a record is found with a greater record length than the LRECL, an error condition will be raised on the split file and processing for that split file will be terminated. The split file will then need to be generated with an EXTRACT following the termination of the dump process.

There are considerations for processing split files if the index file and TIME operands are in use, both separately and together.

Split File Index Processing

The split file index file has the following characteristics:

```
RECFM=FB or F
LRECL=192
DSORG=PS
BLKSIZE=192 or a multiple of 192
```

To have CA SMF Director use a split file index, the file must be created and cataloged prior to the first dump process that accesses the index. CA SMF Director will not create the index file while running. The index file is dynamically allocated with DISP=OLD when being processed.

The index is a character format file, and can be read by column or by using a space character as a delimiter. The layout is available in member $SPINDEX in CASFMAC.

There are four possible states for a split file indicated in the index. In addition, diagnostic records that do not contain information about a split file may also appear in the index. These diagnostic records provide information that can be used on an EXTRACT process to assist in recreating a split file that has failed.

The split file states are as follows:

Complete (C)

The split file has all of the records it will contain. No further records will be written to it. The file is ready for post processing. Files are marked Complete at the end of a dump process when there is no TIME operand on the SPLIT statement, or when a boundary is crossed as indicated by a TIME or TIMECLOSE operand.

Empty (E)

This split file has no records in it, and will not have any records added to it. This is a special form of the Complete state that can be checked on by post-processors so that the file can be skipped.

Incomplete (I)

This state indicates a split file was created when there was a TIME operand on the SPLIT statement. The file was the last file allocated during the dump process for the SPLIT statement, and more records might be added to this file during the next dump process.

Abnormally Terminated (X)

This state indicates that an abnormal termination was encountered when generating a split file. The statistics in the entry indicate the number of records written to the split file and the time range of the records when the error occurred. The ABEND code is also reported.

When an abnormal termination record appears in the split file index, it is followed by a diagnostic record. A diagnostic record is indicated with a state of D. The information in the abnormal termination record and diagnostic record can be used to create an EXTRACT statement to recreate the failed split file.

At termination time, if the INDEXDUP operand is indicated on the SPLIT statement, a copy of the index will be written to the ddname indicated in the INDEXDUP operand.

Note: If multiple SPLIT statements are sharing a single split file index, the INDEXDUP operand only needs to be on the last SPLIT statement that indicates that index.

Split File Time Processing

In addition to creating raw split files with start and end times that match the history files, you can also create split files that cover specific intervals of time, where the start and end times are user-specified. The TIME operand of the SPLIT statement controls this, along with several other operands (all beginning with TIME).

Here are the similarities between TIME split file processing and non-TIME processing:

- Record selection/exclusion is identical.
- QSAM is used to write the records to the split files.

Here are the differences between TIME split file processing and non-TIME processing:

- Output split files must be part of a GDG for TIME processing.
- Split files are allocated and deallocated dynamically for TIME processing and must be indicated in the JCL for non-TIME processing.

To set up TIME processing for split files, you must do the following in advance:

- Set up GDGs that the split files will be members of, using Access Method Services.
- Determine the times of day (up to 4) when the split files will be separated.
- Define appropriate SMS storage class definitions if SMS is to be used for allocating the split files.

For more information, see the *Systems Programmer Guide*.

TIME processing of split files requires an index file. The TIME process uses the index file to complete files that were started at the end of a dump process. For instance, suppose that the value of the TIME operand is TIME(0000,0800,1600), and an SMF dump process begins at 4:00 a.m. for 5 hours' worth of data. The dump process will find the split file that was begun at 1600 the previous day and add the records in this dump from 2300 the previous day until the date changes. At the point of the date change (time becomes 0000), the current split file will be closed and deallocated. The index will be updated and a new record added for the just-closed split file indicating that the file is complete. The dump process will then allocate a new generation of the GDG to be the new split file and open it for processing of additional records. When the process is complete, an entry will be added to the index indicating the current split file is incomplete. The next dump process will then be able to find the split file and add more records to it until records for 0800 or later are found. Also, any postprocessor of the split files will be able to determine if the file is ready for processing.

Note: While some management of the split files will be done via the index and the TIME operand, there are no additional controls placed on the split files or on the index. The index is a character-based sequential file, so it can be manipulated with almost any text editor on the mainframe. Be sure that users and applications have appropriate access levels set for the index file and the split file GDSs. Do not attempt to access an incomplete time-managed split file (that is, generation 0).

If other applications need to process the information that is found in the split file index, the INDEXDUP operand should be coded on the SPLIT statement in order to generate a copy of the index for use by other applications.

More information:

Using Split Files Generated with the TIME Operand (see page 107)

Using Split Files Generated with the TIME Operand

To ensure that your data is processed effectively and correctly, review the following considerations and requirements when processing split files generated from SPLIT statements with the TIME operand.

Important! CA SMF Director is the owner of the GDG data set with relative generation 0 so it can keep adding records to it, if necessary. Since this generation 0 data set may not be complete, application code should not process or access it.

Older generations in the GDG are freely available for application use and can be processed by any programs or processes that consume the SMF records in the split file. These files can be accessed either via the relative generation number or by the fully qualified data set name that is in the split file index (or a duplicate copy of the index). CA SMF Director does not use the relative index to allocate these split files, so application JCL can be coded to use -1, -2, or any other negative relative value.

If a complete file must be processed immediately, such as a split file that must be processed soon after midnight, the TIMECLOSE operand can be used to ensure that the interval is closed if the SMF timestamp value passes the boundary of a split file. Traditional TIME-based split files will only switch to a new generation if an SMF record for that split file is archived during the dump process. By using TIMECLOSE, the file will be closed if any system SMF record is found to be in the next interval defined in the TIME operand.

For example, to generate a split file that contains a full day of RMF data and that would be available a few minutes past midnight, the following SPLIT statement could be coded:

```
SPLIT TODD(DAILYRMF) SID(PROD) TIME(0000) TIMECLOSE(YES) SELECT(70:79)
TIMEGDG(SYS2.DAILYRMF) TIMEUNIT(SYSDA) TIMEFILL(YES).
```

If a switch command is issued soon after midnight (at about 12:05 a.m. for example), the dump process that runs will finish writing the previous day's generation in the SYS2.DAILYRMF.SIDPROD GDG and will allocate the current day's data set in the GDG. The RMF data for yesterday would be available by allocating this data set:

```
SYS2.DAILYRMF.SIDPROD(-1)
```

The job that processes this daily RMF file could be run every day without changing the JCL.

By setting TIMEFILL to YES, CA SMF Director will create a daily RMF file even on days when RMF data is not being recorded. This way the daily application that processes the RMF data can always use the -1 relative generation without there being a chance of processing the same data twice.

STREAMOPTIONS

This control statement can be used to override options indicated on the OPTIONS and DUMPOPTIONS statements for a particular logstream or substream within a configuration. The STREAMOPTIONS statements must be included in a COMPILE CONFIGURATION block between the BEGIN and END statements. The STREAMOPTIONS statement is optional for logstreams defined within a configuration, since all of the archival properties for a logstream can be defined in the OPTIONS and DUMPOPTIONS statement.

A STREAMOPTIONS statement is required for each substream so that the SMF record types for the stream can be indicated. For substreams, the record types that will be archived to the stream must be indicated in a SELECT operand or excluded in an EXCLUDE operand. Not all record types need to be accounted for as the traditional MAN file archives will still be produced for all records not defined to a stream, but a record type can be designated to only one substream.

Each STREAMOPTIONS statement applies to one logstream or substream that is defined within the configuration being compiled. SMF logstreams that are shared between systems (which is not recommended) that need specific settings must have a STREAMOPTIONS statement within each configuration for each system. Almost all of the operands in the STREAMOPTIONS statement are optional. If they are omitted, the values are taken from either the DUMPOPTIONS statement that is in effect for the configuration or the OPTIONS statement that contains the global options for the SCDS.

Statement Format:

STREAMOPTIONS

```
►►─ STREAMOPTIONS ─┬SUBNAME(subname)────┬SELECT(list) ┬┐
                   │                    └EXCLUDE(list)┘│
                   └LSNAME(lsname) ────────────────────┴──────────────────►

►─┬───────────────────────┬──────────────────────────────────────────────►
  └COPIES( ─┬─ * ◄ ─┬─ ) ─┘
            ├─ 1 ───┤
            └─ 2 ───┘

►─┬──────────────┬───────────────────────────────────────────────────────►
  ├─ UPCATLG ◄ ──┤
  ├─ NOCATLG ────┤
  └─ CATLG ──────┘

►─┬────────────────────────┬─────────────────────────────────────────────►
  ├─ PREFIX(─┬──*◄ ───┬─)──┤
  │          └ prefix ┘    │
  └─ NOPREFIX ─────────────┘

►─┬─────────────────────────┬────────────────────────────────────────────►
  ├─ APREFIX(─┬──*◄ ───┬─)──┤
  │           └ prefix ┘    │
  └─ NOAPREFIX ─────────────┘

►─┬─────────────────────┬────────────────────────────────────────────────►
  ├─ RETPD(─┬──*◄ ─┬─)──┤
  │         └ nnnn ┘    │
  ├─ EXPDT(─┬──*◄ ───┬─)┤
  └         └ yyyddd ┘ ─┘

►─┬──────────────────────┬───────────────────────────────────────────────►
  ├─ ARETPD(─┬──*◄ ─┬─)──┤
  │          └ nnnn ┘    │
  ├─ AEXPDT(─┬──*◄ ───┬─)┤
  └          └ yyyddd ┘ ─┘

►─┬──────────────────────┬───────────────────────────────────────────────►
  └─ TRTCH(─┬──*◄ ──┬─)──┘
            └ value ┘

►─┬───────────────────────┬──────────────────────────────────────────────►
  └─ ATRTCH(─┬──*◄ ──┬─)──┘
             └ value ┘

►─┬─────────────────────────────────┬────────────────────────────────────►
  └─ MAXFILESONVOL( ─┬─ * ◄ ─┬─ ) ──┘
                     └─ n ───┘

►─┬────────────────────────────────┬─────────────────────────────────────►
  └─ DAYSTACK( ─┬─ * ◄ ──┬─ ) ─────┘
                ├─ YES ──┤
                └─ NO ───┘

►─┬──────────────────────────────┬─┬─────────────────────────────┬───────►
  └─ PDSNLARGE( ─┬─ * ◄ ─┬─) ────┘ └─ ADSNLARGE( ─┬─ * ◄ ─┬─ ) ─┘
                 ├─ YES ─┤                        ├─ YES ─┤
                 └─ NO ──┘                        └─ NO ──┘

►─┬─────────────────────┬────────────────────────────────────────────────►
  └─ SDAY(─┬──*◄ ─┬─)───┘
           └ hhmm ┘
```

```
►─┬─────────────────────────────┬──────────────────────────────────────────►
  ├─ PDEVN(────*◄ ───)─┤
  ├─            ├ dname ┤  ──┤
  ├─            └ NONE ─┘  ──┤
  ├─ PSTORC(──*◄ ───)─┤
  ├─            ├ class ┤  ──┤
  └─            └ NONE ─┘  ──┘
```

```
►─┬─────────────────────────────┬────────────────────────────── . ──────►◄
  ├─ ADEVN(────*◄ ───)─┤
  ├─            ├ dname ┤  ──┤
  ├─            └ NONE ─┘  ──┤
  ├─ ASTORC(──*◄ ───)─┤
  ├─            ├ class ┤  ──┤
  └─            └ NONE ─┘  ──┘
```

Operands and Descriptions:

LSNAME(lsname)|SUBNAME(subname)

Identifies the stream that this definition applies to, for the configuration being compiled. The name of the stream must be included either in a DUMPOPTIONS statement processed in this compilation or in a previous DUMPOPTIONS statement. Both logstream names and substream names can be up to 26 characters. Logstream names must also begin with IFASMF and can contain up to two additional 8-byte nodes. Substream names cannot begin with a first node that begins with the letter “I” or the three characters "SYS" and must contain at least two nodes.

SELECT(list)|EXCLUDE(list)

Determines which record types are included or excluded within a substream. The list may include any valid SMF record types, numbers 0 through 255, listed as individual record types separated by a comma and/or as a range separated by a colon.

One of these operands is required if SUBNAME is specified. If neither SELECT nor EXCLUDE is specified, the selection defaults to EXCLUDE(0:255), which effectively disables the substream.

If LSNAME is specified, an error is reported if SELECT or EXCLUDE is also specified.

Note: Multiple logstreams can archive the same records, but this is not allowed for substreams. A record type can only be assigned to one substream.

Records that are not assigned are dumped into the traditional MAN file archive.

COPIES(*|1|2)

Specifies whether the records should be saved in a single archive (the primary) or in two archives (primary and alternate). If this operand is not coded or the asterisk is coded, the default value is taken from the levels above the stream definition.

UPCATLG|NOCATLG|CATLG

Indicates whether the history files generated for this stream during the dump process should be cataloged:

- If NOCATLG is specified, the history files will not be cataloged.
- If CATLG is specified, the history files will be cataloged.
- If the default UPCATLG is specified, the catalog value is taken from the levels above the stream definition.

PREFIX(*|prefix)|NOPREFIX

Sets the first 26 characters of the history file data set names (DSNs) for the primary archive. If this operand is not coded or the asterisk is coded, the default value is taken from the levels above the stream definition.

APREFIX(*|prefix)|NOAPREFIX

Sets the first 26 characters of the history file data set names (DSNs) for the alternate archive. If this operand is not coded or the asterisk is coded, the default value is taken from the levels above the stream definition.

EXPDT(*|yyyddd)|RETPD(*|nnnn)

Specifies the expiration date or retention period of the history file data sets within the primary pool. If the expiration date is specified, it must be in Julian date format. If this operand is not coded or the asterisk is coded, the default value is taken from the levels above the stream definition.

See Control Statement Language Rules (see page 37) for a description of two special expiration dates: 099000 and 099366.

We recommend that expiration dates or retention periods be used. They increase the integrity of the SMF data. If you use the AUTODEL feature, you must set either expiration dates or retention periods.

Note: If the PSTORC option is in effect, SMS may override the specified expiration date or retention period.

AEXPDT(*|yyyddd)|ARETPD(*|nnnn)

Specifies the expiration date or retention period of the history file data sets within the alternate pool. If the expiration date is specified, it must be in Julian date format. If this operand is not coded or the asterisk is coded, the default value is taken from the levels above the stream definition.

See Control Statement Language Rules (see page 37) for a description of two special expiration dates: 099000 and 099366.

We recommend that expiration dates or retention periods be used. They increase the integrity of the SMF data. If you use the AUTODEL feature, you must set either expiration dates or retention periods.

Note: If the ASTORC option is in effect, SMS may override the specified expiration date or retention period.

TRTCH(*|value)

ATRTCH(*|value)

Lets you use the z/OS JCL TRTCH parameter on dynamically allocated history files in CA SMF Director. The values of the TRTCH and ATRTCH operands are the same as the values allowed in z/OS JCL when writing an output file to tape. TRTCH is used to set the recording technique for the primary history file. ATRTCH is used to set the TRTCH for the alternate history file.

Valid values are as follows:

NOCOMP

For cartridge tapes, writes the history file without data compression.

COMP

For cartridge tapes, writes the history file with data compression.

C

Performs data conversion for history files on reel tapes.

E

Writes history files to reel tape with even parity.

ET

Writes history files to reel tape with even parity and performs EBCDIC translation.

T

Writes history files to reel tape and performs EBCDIC translation.

Note: Although CA SMF Director supports TRTCH options C, E, ET, and T, they are not recommended for history files.

If neither TRTCH nor ATRTCH is specified, the value is taken from the levels above the stream definition. If not specified at a higher level, the system default is used.

MAXFILESONVOL(*|n)

Lets you override the value set for MAXFILESONVOL. (See the description of MAXFILESONVOL on the DUMPOPTIONS and OPTIONS statements.)

- Specify a number to set the value for this stream only.
- Specify 0 to reset the value for this stream (the local value) so that the global value from the OPTIONS statement is used instead.

If this operand is not coded or the asterisk is coded, the value is taken from the DUMPOPTIONS statement.

DAYSTACK(NO|YES|*)

Specifies whether the daystacking feature should be turned on or off. NO turns daystacking off. See Features for a description of the daystacking feature. If this operand is not coded or the asterisk is coded, the default value is taken from the levels above the stream definition.

PDSNLARGE(*|YES|NO)

Allocates the primary history file with the DSNTYPE=LARGE operand. This operand is only used for DASD history files when the DYNAM operand is specified; otherwise it is ignored. If this operand is not coded or the asterisk is coded, the default value is taken from the levels above the stream definition.

ADSNLARGE(*|YES|NO)

Allocates the alternate history file with the DSNTYPE=LARGE operand. This operand is only used for DASD history files when the DYNAM operand is specified; otherwise it is ignored. If this operand is not coded or the asterisk is coded, the default value is taken from the levels above the stream definition.

SDAY(*|hhmm)

Specifies the hour (hh) and minute (mm) that CA SMF Director is to treat as the logical start of a day. If this operand is not coded or the asterisk is coded, days start at the value indicated in the DUMPOPTIONS statement for this configuration. If no value is set, the value defaults to midnight, coded as SDAY(0000).

Note: The global SDAY value is used to determine the starting and ending times for all EXTRACT and PRINT requests, except when the FROM, JFROM, TO, or JTO operand gives a specific starting or ending time. The value specified here is used for the daystacking feature only.

PDEVN(*|dname|NONE)

'dname' is the esoteric device name to be used when creating SMF history files for this stream in the primary pool. Up to eight characters can be specified. It is mutually exclusive with PSTORC. Specify NONE to reset the device name. If neither PDEVN nor PSTORC is specified and the DYNAM option is enabled, the device name or storage class used is taken from the DUMPOPTIONS statement.

Note: If you are changing the PDEVN operand from one tape unit name to another, you must immediately run a TAPEINIT command indicating this stream.

If using DASD for history files, this operand must only be used for non-SMS managed volumes.

PSTORC(*|class|NONE)

'class' is the IBM DFSMS storage class to be used when creating SMF history files for this stream in the primary pool. Up to eight characters can be specified. It is mutually exclusive with the PDEVN keyword. Specify NONE to reset the storage class. If neither PDEVN nor PSTORC is specified and the DYNAM option is enabled, the device name or storage class used is taken from the DUMPOPTIONS statement.

If using DASD for history files, this operand will allocate files only to SMS-managed volumes. Note that some issues may arise with DASD history file allocations based on how SMS is set up at your data center. For DASD history allocations with the DYNAM option set, there must still be a temporary history file allocated in the JCL that indicates how large the history file should be. SMS may not allow the allocation on the storage class indicated in the PSTORC operand and will indicate an allocation error for the history file, causing the dump process to end prematurely.

To avoid this issue, here are some remedies you can:

- Include a data set name on the HISTORY1 or HISTcc1 DD statement. This can be any valid data set name that would be allocated in the storage class indicated in the PSTORC operand.

  Note: This is the simplest remedy and the one that CA recommends.

- Alter the ACS routines in the SMS DASD allocation to assign no class to temporary data sets, where the fourth node of the name is the same as the dump process JCL job name or procedure name.

  Note: These changes are only for DASD history files allocated with SMS. Non-DASD and non-SMS allocations will not be affected by this.

ADEVN(*|dname|NONE)

'dname' is the esoteric device name to be used when creating SMF history files for or this stream in the alternate pool. Up to eight characters can be specified. It is mutually exclusive with ASTORC. Specify NONE to reset the device name. If neither ADEVN nor ASTORC is specified and the DYNAM option is enabled, the device name or storage class used is taken from the DUMPOPTIONS statement.

Note: If you are changing the ADEVN operand from one tape unit name to another, you must immediately run a TAPEINIT command against this stream.

If using DASD for history files, this operand must only be used for non-SMS managed volumes.

ASTORC(*|class|NONE)

'class' is the IBM DFSMS storage class to be used when creating SMF history files for this stream in the alternate pool. Up to eight characters can be specified. It is mutually exclusive with the ADEVN keyword. Specify NONE to reset the storage class. If neither ADEVN nor ASTORC is specified and the DYNAM option is enabled, the device name or storage class used is taken from the DUMPOPTIONS statement.

If using DASD for history files, this operand will allocate files only to SMS-managed volumes. Note that some issues may arise with DASD history file allocations based on how SMS is set up at your data center. For DASD history allocations with the DYNAM option set, there must still be a temporary history file allocated in the JCL that indicates how large the history file should be. SMS may not allow the allocation on the storage class indicated in the ASTORC operand and will indicate an allocation error for the history file, causing the dump process to end prematurely.

To avoid this issue, here are some remedies you can use:

- Include a data set name on the HISTORY2 or HISTcc2 DD statement. This can be any valid data set name that would be allocated in the storage class indicated in the PSTORC operand.

  Note: This is the simplest remedy and the one that CA recommends.

- Alter the ACS routines in the SMS DASD allocation to assign no class to temporary data sets, where the fourth node of the name is the same as the dump process JCL job name or procedure name.

  Note: These changes are only for DASD history files allocated with SMS. Non-DASD and non-SMS allocations will not be affected by this.

Sample STREAMOPTIONS Statement:

```
COMPILE CONFIGURATION STORE.
  BEGIN CONFIG(002) SID(XE96) NAME('C.A. XE96').
    DUMPOPTIONS COPIES(2) NOCATLG SCRATCH NOVERIFY
        RETPD(60) ARETPD(365) PREFIX(SYSPROG.CASMF.XE96)
        LSNAME(IFASMF.ALL,IFASMF.T30,IFASMF.RMF)
        TCH(NOCOMP) ATRTCH(COMP).
    STREAMOPTIONS LSNAME(IFASMF.RMF) RETPD(90) PREFIX(SYSPROG.CASMF.XE96RMF)
        COPIES(1).
  END CONFIG(002).
```

TAPEINIT

This control statement forces CA SMF Director to use a new magnetic tape volume the next time an SMF history file needs to be created. When using a tape device, this product writes multiple files to a single tape volume until the end of the volume has been reached, unless this control statement has been issued. TAPEINIT is useful when a tape volume becomes damaged, or a new volume is desired for some other reason.

TAPEINIT must be used if you are migrating from tape to DASD history files.

Note: If a configuration has streams defined that have active tapes being used for dumping, the TAPEINIT process will force all streams to begin using new tapes.

Statement Format:

TAPEINIT

```
►►─ TAPEINIT ─ SID(sid) ─┬─────────────┬─┬──────────────────────────────────┬─ . ─►◄
                         ├─ PRIMARY ◄ ─┤ ├─ LSNAME( ──┬─ lsname ──┬─ ) ───┤
                         └─ ALTERNATE ─┘ │            └─ NONE ────┘       │
                                         └─ SUBNAME( ─┬─ subname ─┬─ ) ───┘
                                                      └─ NONE ────┘
```

Operands and Descriptions:

SID(SID)

Specifies the SMF ID for the system configuration to be used in processing the TAPEINIT control statements.

PRIMARY

Identifies that the history tape to be changed is for the primary pool. The default is primary.

ALTERNATE

Identifies that the history tape to be changed is for the alternate pool.

LSNAME(lsname|NONE)

Forces an end-of-volume on a tape being used for dumping the specified logstream. All other tapes remain valid. If NONE is coded, only the MAN file archive tape and substream tapes are to be reset. All other tapes remain valid. LSNAME and SUBNAME are mutually exclusive unless both are NONE.

SUBNAME(subname|NONE)

Forces an end-of-volume on a tape being used for dumping the specified substream. All other tapes remain valid. If NONE is coded, only the MAN file archive tape and logstream tapes are to be reset. All other tapes remain valid. LSNAME and SUBNAME are mutually exclusive unless both are NONE.

Sample TAPEINIT Statement:

```
TAPEINIT SID(XE44) PRIMARY.
```

TAPEINIT statement output:

```
CA                                           AUDIT TRAIL REPORT                         SMFD       PAGE     1
CA SMF Director     12.6                                                        SYSA      01 JUL yyyy  12:37:35

  TAPEINIT SID(XE44) PRIMARY  .

CAF0801I  PRIMARY  POOL VOLUME 201731 FOR SID(XE44) MARKED COMPLETE - NEW TAPE REQUESTED

CAF2901I  RUN COMPLETE, HIGHEST CONDITION CODE WAS 0
```

UPDTX

This control statement provides the ability to change the device name, expiration date, volume serials, or file sequence number for a range of index entries on the SCDS. The actual history file data set attributes are not changed. All entries within the date range specified will be updated with the new information.

Important! We highly recommend that the SCDS be backed up before execution of any UPDTX command, especially with the ALL parameter coded.

Statement Format:

UPDTX

```
▶▶─ UPDTX ─┬─ LOW(date time) ──┬─┬─ HIGH(date time) ──┬─ SID(sid) ──────────▶
           └─ JLOW(date time) ─┘ └─ JHIGH(date time) ─┘

 ▶─┬───────┬─┬─────────────────┬─┬────────────────┬─┬──────────────────┬─────▶
   └─ ALL ─┘ └─ UNIT(devname) ─┘ └─ EXPDT(date) ──┘ └─ FILESQ(filesq) ─┘

 ▶─┬──────────────────┬─┬──────────────┬─┬───────────────────────────┬─ . ───▶◀
   └─ VOLUME(volser) ─┘ └─ ALTERNATE ──┘ ├─ LSNAME(logstream name) ──┤
                                         └─ SUBNAME(substream name) ─┘
```

Operands and Descriptions:

LOW(date time)

If the ALL parameter is coded, this specifies the exact starting date and time of the **first** SCDS index entry to be updated. Otherwise, it specifies the exact starting date and time of the **only** SCDS entry to be updated.

"date" is in Gregorian (mmddyy) format.

"time" is in hour/minute/second format (hhmmss).

HIGH(date time)

If the ALL parameter is coded, this specifies the exact ending date and time of the **last** SCDS index entry to be updated. Otherwise, it specifies the exact ending date and time of the **only** SCDS entry to be updated.

"date" is in Gregorian (mmddyy) format.

"time" is in hour/minute/second (hhmmss) format.

SID(sid)

Specifies the SMF ID for the history file index, and defines the origin of the SMF input data.

"sid" is a valid SMF SYSID.

JLOW(date time)

If the ALL parameter is coded, this specifies the exact starting date and time of the **first** SCDS index entry to be updated. Otherwise, it specifies the exact starting date and time of the **only** SCDS entry to be updated.

"date" is in Gregorian (mmddyy) format.

"time" is in hour/minute/second format (hhmmss).

JHIGH(date time)

If the ALL parameter is coded, this specifies the exact ending date and time of the **last** SCDS index entry to be updated. Otherwise, it specifies the exact ending date and time of the **only** SCDS entry to be updated.

"date" is in Gregorian (mmddyy) format.

"time" is in hour/minute/second (hhmmss) format.

ALL

Indicates that more than one SCDS index entry is to be updated. If this parameter is coded, the LOW/JLOW specification must exactly equal the starting date and time of a SCDS index entry, and the HIGH/JHIGH specification must exactly equal the ending date and time of a SCDS entry. If ALL is not coded, the LOW/JLOW and HIGH/JHIGH operands must exactly equal, respectively, the starting and ending dates and times of a single SCDS index entry.

UNIT(devname)

Specifies the new device name to be used when allocating the history file.

VOLUME(volser)

Specifies the new serial number of the volume on which the indexed data set resides.

EXPDT(date)

Specifies the new Julian date in the form of (cyyddd) when the index entry will expire, where c designates the century (0 is the 20th century and 1 is the 21st century). If the AUTODEL option is enabled, index entries that are expired will be removed from the SCDS file.

See Control Statement Language Rules (see page 37) for a description of two special expiration dates: 099000 and 099366.

FILESQ(filesq)

Specifies the new file sequence number of the indexed data set if it resides on tape. If the data set resides on DASD, the value should be 0.

ALTERNATE

Specifies that the entry to be updated is for the alternate pool index.

LSNAME(logstream name)

Updates entries in the specified logstream. If the name is omitted, the update will be performed on dumps that are not associated with a logstream. The LSNAME and SUBNAME operands are mutually exclusive.

SUBNAME(substream name)

Updates entries in the specified substream. If the name is omitted, the update will be performed on dumps that are not associated with any substream. The LSNAME and SUBNAME operands are mutually exclusive.

Sample UPDTX Statement:

```
UPDTX    SID(XE44) LOW(040198 010100) HIGH(041898 235800) ALL
         UNIT(3490) ALTERNATE.
```

UPDTX statement output:

```
CA                                              AUDIT TRAIL REPORT                           SMFD       PAGE     1
CA SMF Director     12.6----                                                         SYSA      27 JUL yyyy  13:39:55

  UPDTX    SID(XE44) LOW(040198 010100) HIGH(041898 235800) ALL
           UNIT(3490) ALTERNATE.

CAF3610I  INDEX ENTRY FOR SYSPROG.SF16.MS070698.T1626I15              UPDATED
CAF3610I  INDEX ENTRY FOR SYSPROG.SF16.MS071098.T1036I15              UPDATED
CAF3610I  INDEX ENTRY FOR SYSPROG.SF16.MS070798.T1736I57              UPDATED
CAF3611I  3 INDEX ENTRIES UPDATED

CAF2900I  RUN COMPLETE
```

Chapter 5: SMFDLS Control Statement Reference

This chapter contains detailed descriptions of the control statements used by SMFDLS, the CA SMF Director logstream dump driver. If you are recording SMF data to logstreams or substreams, use this chapter to help set up and customize the way SMFDLS processes data from the logstreams and provides it to CA SMF Director.

If you are not using logstreams to record SMF data, you can skip this chapter.

The SMFDLS Driver

In order to access data that SMF has recorded in logstreams, IBM recommends that their IFASMFDL utility be used (the IBM logstream dump utility). CA SMF Director uses the SMFDLS driver program to interface with the IFASMFDL utility. The driver also interfaces with CA SMF Director, acting as an intermediary.

The driver provides three user exits for IFASMFDL. They are used to move records from the logstreams to CA SMF Director.

Changes to Dump Process JCL

In addition to changing the program name in the JCL from SMFD to SMFDLS, there are some changes that need to be made to your CA SMF Director JCL if data is being dumped from logstreams, due to ddname conflicts.

The table below indicates the non-logstream dump process ddnames that will need to be changed to logstream dump process ddnames:

Change this:	To this:
SYSIN	CAIIN
SYSPRINT	CAIPRINT

Note: Only the ddname needs to be changed. Nothing else about thefile is different.

In addition to these changes, there are four new ddnames for the driver program:

CAILSNM

Placeholder ddname. This data set passes the name of the logstream to be dumped to SMFDLS. This data set is optional, and is only needed if the JCL control statement is coded in the SMFDLSIN file described below.

SMFDLSIN

Input control statement file for SMFDLS control statements. (These are not for SMFD control statements. Those should be allocated to CAIIN).

SMFDLSPR

Output print file for SMFDLS only. SMFD output will be written to CAIPRINT.

SMFDOUT

This is the output data set for SMF records being read from the logstream. Typically, since CA SMF Director is going to be archiving the records, it can be coded as DUMMY. However, if you want to have another copy of the records produced in addition to the archives and split files produced by CA SMF Director, this DD can have a disk or tape data set allocated as well. If the disk or tape allocation is done, an additional copy of the records written to the SMFD archive will be written to the data set allocated to SMFDOUT.

CASFDUML Example Procedure

```
//CASFDUML PROC LS=JCL.ERROR,A=A
//********************************************************************
//* CASFDUML - Logstream Dumping Procedure
//*
//* If automated dumping of SMF data recorded to logstreams is
//* being performed, modify a copy of this procedure and copy it
//* to a PROCLIB that contains started tasks.
//*
//* Modifications to be made (see notes in parentheses in the JCL):
//*
//* (1) Change the DSN to your Authorized CA SMF Director Load
//*     Library
//*
//* (2) The SMFDOUT DD is needed to drive the user exits used by
//*     the SMFDLS wrapper in the IFASMFDL utility.  While the
//*     default is to set the DD up as a DUMMY, it can also be
//*     allocated to a disk or tape output dataset at your
//*     discretion.
//*
//* (3) Change the DSN to the name of your SMF Control Data Set
//*     (SCDS)
//*
```

```
//* (4) Change the dataset name and member name to point to the
//*     required CA SMF Director Control Statement File. The
//*     shipped default contains the following statement:
//*
//*     DUMP.
//*
//* (5) Change the dataset name and member name to point to the
//*     required CA SMF Director logstream dump utility control
//*     statement file. The shipped default contains the following
//*     statements:
//*
//*        UNIT(SYSDA)
//*     *  STORCLAS(TEMP)
//*        JCL
//*
//*       Note: In the above example STORCLAS is a comment.
//*
//* (6) Change the value of UNIT to indicate a temporary DASD
//*     location, or, if SMS is in use, change the UNIT parameter
//*     to STORCLAS and indicate a temporary DASD location
//*
//*
//********************************************************************
//DUMP    EXEC PGM=SMFDLS,REGION=0M,TIME=1440
//STEPLIB  DD  DISP=SHR,DSN=CASFLOAD                  (1)
//SMFDLSPR DD  SYSOUT=&A.                **PRINT FROM SMFDLS
//SYSPRINT DD  SYSOUT=&A.                **PRINT FROM IFASMFDL
//CAIPRINT DD  SYSOUT=&A.                **PRINT FROM SMFD
//SYSXDIAG DD  SYSOUT=&A.
//SMFDOUT  DD  DUMMY                                  (2)
//SCDS     DD  DISP=SHR,DSN=CORP.CASMF.SCDS           (3)
//CAIIN    DD  DISP=SHR,DSN=CORP.CASMF.CNTL(SMFDDUMP) (4)
//SMFDLSIN DD  DISP=SHR,DSN=CORP.CASMF.CNTL(SMFDIN)   (5)
//CAILSNM  DD  DISP=(NEW,DELETE),SPACE=(TRK,1),       (6)
//             UNIT=SYSDA,DSN=&LS.
```

A copy of the CASFDUML procedure is provided in the CASFPROC target library. It is set up so that a start command can be issued to select an appropriate set of control statements, based on the logstream that is to be dumped from. CASFPARM target library member SMFDDUMP contains the DUMP control statement for SMFD, as well as any SPLIT statements that are required. See the *Systems Programmer Guide* for more information about setting up automated dumping with CASFDUML.

The SMFDOUT DD is required for processing since this is the ddname in the generated statements that are passed to IFASMFDL. SMFDOUT can also be an allocated DASD or tape data set, if an additional copy of the records is desired.

The CAILSNM DD is optional. It gives the name of the logstream containing the records to be dumped. By placing this ddname in the procedure, and adding a JCL control statement in the CAIIN control statements, the logstream name is passed to SMFDLS outside of the control statements. That way, there is no need to have a separate set of SMFDLS control statements for each logstream.

Control Statement Syntax

Control statements for SMFDLS differ from control statements for SMFD. They are more similar to control statements for IFASMFDL. The input control statement file (ddname SMFDLSIN) must have the following attributes:

```
LRECL= 80
BLKSIZE=multiple of 80
RECFM=F or FB
```

The file can be a sequential data set or a PDS member. Statement text must be placed in columns 1 through 72 (columns 73 through 80 are ignored as sequence numbers.) Each input record can be a comment or can contain exactly one statement. Since the longest statement possible is 34 characters, there is no facility provided to continue a statement to a second line.

Records that begin with an asterisk in column one are considered to be comments. In addition, records that are blank from columns 1 through 72 are also considered to be comments.

Statements themselves can begin in any column, and can be in any order, but they must be completed in one single record in the file.

All of the control statements used by SMFDLS fall into one of two categories:

Type of Statement	Syntax
Statements with Operands	statement(operand value) comment
Statements without Operands	statement comment

Note: Operand values must be in parentheses right after the operand with no spaces in between. See the sample statements in the rest of this chapter for examples.

Dump Processing of Logstreams

To dump data into CA SMF Director from SMF logstreams, you must use the SMFDLS driver utility. In most cases, the only control statements needed by SMFDLS are the logstream name statement (LSNAME) or the JCL control statement, which indicates that the logstream name is in the JCL. In addition, either a UNIT name or SMS storage class must be provided to SMFDLS so that the data set for the control statements for IFASMFDL can be allocated dynamically to the SYSIN DD.

Additional parameters are provided to tailor dump processing to your needs. For example, since logstreams that are backed by a list structure in a coupling facility can contain data from multiple systems, a statement is provided to allow the dumping of one system's data on another system. This can come in handy when recording data from smaller systems that may not have tape drives. Rather than needing a two-step process on two systems (a dump with the IBM utility followed by a SOURCE dump), the data can be processed directly by CA SMF Director.

Date and Time Management for Logstream Dumps

Parameters let you specify date and time ranges. They are not needed by default because CA SMF Director keeps track of its location in the logstream and begins dumping from that point. You can even switch to MAN file recording, then back to logstream recording; CA SMF Director can handle that possibility and ensure that all of the SMF records are processed and archived.

The default start date/time values are as follows for dump processing:

First Logstream Dump

January 1, 1900 at 00:00

Subsequent Logstream Dumps

The date/time of the last record dumped in the previous dump for that system and logstream

The default end date/time for all logstream dumps is the current date and time less one minute.

We recommend that you allow CA SMF Director to manage the start and end times of SMF logstream dumps. If the recommendation is followed, most dumps can be run with only the logstream name, obtained either via control statement or from the CAILSNM DD, plus either the UNIT or STORCLAS control statements. This ensures that all of the data in the logstream is archived.

Here are typical dump control statements for SYSB. Use STORCLAS WORKDA for dynamic allocation of SYSIN statements for IFASMFDL. The logstream name is IFASMF.LOGGER1. These statements are passed to SMFDLS via the SMFDLSIN DD:

```
LSNAME(IFASMF.LOGGER1)
STORCLAS(WORKDA)
SID(SYSB)
```

Note: If the dump is to be run on SYSB, the SID control statement can also be omitted.

If the CASFDUML procedure is used and the logstream name is passed to the procedure in the START command, SMFDLSIN DD could contain these statements:

```
STORCLAS(WORKDA)
JCL
```

The JCL statement says to use the logstream name identified by the data set name defined in the CAILSNM DD.

When to Use Date and Time Control Statements

For the first dump of a logstream, we recommend that CA SMF Director be allowed to dump the contents without any timestamps, unless you want to begin from a certain point. If that is the case, we recommend that you only specify start date for the dump, and allow the end date/time to default to the current date/time.

The STARTDATE, STARTTIME, ENDDATE, and ENDTIME parameters are available for recovery situations, where a dump may have been missed. They are also available if you want to begin CA SMF Director management of logstream SMF data from a particular date and time.

Sample Date and Time Statements

Manage SMF data from logstream IFASMF.SMFLOGR1 beginning at midnight on January 1, 2008:

```
LSNAME(IFASMF.SMFLOGR1)
UNIT(3390)
STARTDATE(2008001)
```

Dump all data for SMF ID SYSB from logstream IFASMF.SMFLOGR1 from January 15, 2008 to January 19, 2008:

```
LSNAME(IFASMF.SMFLOGR1)
UNIT(3390)
STARTDATE(2008015)
ENDDATE(2008019)
SID(SYSB)
```

Dump all records for SYSB from logstream IFASMF.SMFLOGR1 from 5 a.m. to 8 p.m. on January 19, 2008:

```
LSNAME(IFASMF.SMFLOGR1)
UNIT(3390)
STARTDATE(2008019)
STARTTIME(0500)
ENDDATE(2008019)
ENDTIME(2000)
SID(SYSB)
```

Control Statement Summary

The table below lists the statements and provides brief descriptions:

Statement	Description
ABEND	Converts non-zero return codes of 100 or higher into user abends. This control statement should only be used at CA's direction.
ENDDATE(yyyyddd)	End date for records being dumped from a logstream. The default is the current date or the previous date if the dump begins in the first minute of a day (see ENDTIME for more details).
ENDTIME(hhmm)	End time for records being dumped from a logstream. The default is the current minute less one.
JCL	Indicates that the logstream name for the dump should be picked up from the JCL in the ddname CAILSNM. This overrides the LSNAME control statement, if both are present.
LSNAME(logstream name)	The name of the SMF logstream that is to be dumped. The value of the logstream name must follow the standard IBM logstream naming conventions. In addition, SMF requires that the high-order node of the logstream name must be IFASMF.
NOSTAE	Deactivates any recovery in both the logstream dump driver as well as CA SMF Director. This should be used at CA's direction.

Statement	Description
SID(smfid)	The system ID of the SMF data to be dumped. The default is to dump SMF data from the system where the driver and CA SMF Director will be running. This operand can only be used to dump data from other systems if the SMF logstreams are backed by a list structure in the coupling facility.
STARTDATE(yyyyddd)	Start date for records being dumped. The default is the end date from the last dump for the named logstream for the SID being dumped.
STARTTIME(hhmm)	Start time for records being dumped. The default is the end time from the last dump for the named logstream for the SID being dumped.
STORCLAS(sms storage class)	Which storage class to use when allocating the generated SYSIN statements for the IBM utility IFASMFDL that will be used to copy the data from the logstreams. For non-SMS allocations, the UNIT statement can be used.
UNIT(unit name/address)	Which unit to use when allocating the generated SYSIN statements for the IBM utility IFASMFDL that will be used to copy the data from the logstreams. For SMS allocations, the STORCLAS statement can be used.

Note: Either the LSNAME or JCL control statement must be present in control statements. Also either a UNIT or STORCLAS control statement must be present.

ABEND

Important! The ABEND control statement should only be used at CA's direction.

Under normal circumstances the ABEND statement is not required. Its purpose is to issue a return code when there is a failure in a dump process run. If added to your input statements, all return codes of 100 or higher will be converted into user abends with the value that would have been set as the return code.

Statement Format:

ABEND

```
▶▶─ ABEND ──────────────────────────────────────────────────────────▶◀
```

ENDDATE

The ENDDATE statement is not required. It should only be used if you want to limit the time range of dumps for a logstream.

Statement Format:

ENDDATE

```
▶▶─ ENDDATE(yyyyddd) ──────────────────────────────────────────────▶◀
```

Operands and Descriptions:

(yyyyddd)

> Specifies which records should or should not be dumped. Records in a logstream dated after this date will not be dumped in this run. The format is a seven-digit Julian date. The default value is the current date.

Sample ENDDATE Statement:

To archive one week's worth of data from a logstream:

```
LSNAME(IFASMF.ACCT.DATA)
STORCLAS(WORK)
STARTDATE(2008318)
ENDDATE(2008324)
```

ENDTIME

The ENDTIME statement is not required and can only be used if the ENDDATE statement is used. ENDTIME lets you limit the date range of dumps for a logstream.

Statement Format:

ENDTIME

```
▶▶─ ENDTIME(hhmmssth) ─────────────────────────────────────────────▶◀
```

Operands and Descriptions:

(hhmmssth)

Specifies which records should or should not be dumped. The value indicates that only records with a timestamp value that is less than the value indicated will be dumped. The syntax is a 4, 6, or 8 digit 24-hour clock value.

- If the value is 4 digits long, the digits are processed as a two digit hour and a two digit minute.
- If the value is 6 digits, the value is processed as hours, minutes, and seconds.
- If the value is 8 digits, the value is processed as hours, minutes, seconds, and hundredths of a second.

The default is the current time of day unless the ENDDATE statement is coded and specified as a date other than today's date, in which case the default is 2359.

The control statements that are generated for IFASMFDL will only reflect the hour and minute values from the ENDTIME control statement due to the syntax of the IFASMFDL control statements. The generated control statements are printed in the CAF4035I message in the SMFDLSPR print data set. The exits that are used by SMFDLS to interact with IFASMFDL will ensure that the seconds and fraction are honored during the dump process.

Sample ENDTIME Statement:

To dump 8 hours worth of data from 10 p.m. on August 19, 2008 to 6 a.m. on August 20, 2008:

```
LSNAME(IFASMF.ACCTG)
STORCLAS(WORK)
STARTDATE(2008231)
STARTTIME(2200)
ENDDATE(2008232)
ENDTIME(0600)
```

JCL

The JCL control statement is used to indicate that the logstream to be dumped is set as the DSN of the data set indicated by the CAILSNM DD.

Note: Either the JCL control statement or LSNAME control statement must be present in the SMFDLSIN statements.

Statement Format:

JCL

```
▶▶─ JCL ─────────────────────────────────────────────────────────▶◀
```

Sample JCL Statement:

These are routine dump control statements for a procedure used to dump many different logstreams:

```
UNIT(3390)
JCL
```

LSNAME

The LSNAME control statement identifies the logstream that is going to be dumped in this run of SMFDLS. The logstream must be predefined to CA SMF Director by the LSNAME operand in the DUMPOPTIONS control statement during configuration compilation.

Note: Either the JCL control statement or LSNAME control statement must be present in the SMFDLSIN statements.

Statement Format:

LSNAME

```
▶▶─ LSNAME(logstream name) ─────────────────────────────────────────────▶◀
```

Operands and Descriptions:

(logstream name)

Identifies the logstream that is going to be dumped. The name must begin with IFASMF and follow standard logstream naming conventions.

Sample LSNAME Statement:

These are routine dump control statements for logstream IFASMF.ACCTG:

```
LSNAME(IFASMF.ACCTG)
UNIT(3390)
```

NOSTAE

Important! The NOSTAE control statement should only be used at CA's direction.

Under normal circumstances the NOSTAE statement is not required. When used, it disables all recovery within SMFDLS and CA SMF Director.

Statement Format:

NOSTAE

```
▶▶─ NOSTAE ─────────────────────────────────────────────────────────▶◀
```

SID

If you have SMF logstreams that are backed by a list structure in a coupling facility, this operand can be used to perform dumps of records from another system. The statement might be used if you have a system without tape drives, but you want to archive the SMF data to tape. If the logstreams are available to a system with tape drives, then the dumps can be performed on the remote system.

Statement Format:

SID

```
▶▶─ SID(smf id) ────────────────────────────────────────────────────▶◀
```

Operands and Descriptions:

(smf id)

Specifies which system's records to dump. This operand can only be used if the logstreams are backed by a list structure in the coupling facility, as DASD-only logstreams are single-system in scope. The default is the SMF ID of the system where CA SMF Director is currently running.

Sample SID Statement:

To dump any data not archived for SYSB from logstream IFASMF.ACCTG:

```
LSNAME(IFASMF.ACCTG)
UNIT(3390)
SID(SYSB)
```

STARTDATE

The STARTDATE statement is not required. It should only be used if you want to start dumping records in a logstream from a specified date, rather than have CA SMF Director start with the most recent dump or with January 1, 1900, if this is the first dump. After the first dump, we recommend that you only use it if you do not want CA SMF Director to manage the date range of dumps for you.

Statement Format:

STARTDATE

```
▶▶─ STARTDATE(yyyyddd) ──────────────────────────────────────────▶◀
```

Operands and Descriptions:

(yyyyddd)

Specifies which records should or should not be dumped. Records in a logstream dated on or after this date will be dumped in this run. The format is a seven-digit Julian date.

The default value is the date of the most recent record dumped from a logstream. If this is the first dump for a logstream, the default is January 1, 1900.

Sample STARTDATE Statement:

To archive one week's worth of data from a logstream:

```
LSNAME(IFASMF.ACCT.DATA)
STORCLAS(WORK)
STARTDATE(2008318)
ENDDATE(2008324)
```

STARTTIME

The STARTTIME statement is not required and can only be used if the STARTDATE statement is used. STARTTIME lets you specify a beginning point for managing data from a logstream, rather than have CA SMF Director begin managing data from the oldest record in the logstream. After the first dump, we recommend that you only use it if you do not want CA SMF Director to manage the time range of dumps for you.

Statement Format:

STARTTIME

```
▶▶─ STARTTIME(hhmmssth) ─────────────────────────────────────────▶◀
```

Operands and Descriptions:

(hhmmssth)

Specifies which records should or should not be dumped. The value indicates that only records with a timestamp value that is less than the value indicated will be dumped. The syntax is a 4, 6, or 8 digit 24-hour clock value.

- If the value is 4 digits long, the digits are processed as a two digit hour and a two digit minute.
- If the value is 6 digits, the value is processed as hours, minutes, and seconds.
- If the value is 8 digits, the value is processed as hours, minutes, seconds, and hundredths of a second.

The default is midnight of the STARTDATE. If STARTDATE is not specified, the start time is set to the time when the last record was dumped from the logstream.

The control statements that are generated for IFASMFDL will only reflect the hour and minute values from the STARTTIME control statement due to the syntax of the IFASMFDL control statements. The generated control statements are printed in the CAF4035I message in the SMFDLSPR print data set. The exits that are used by SMFDLS to interact with IFASMFDL will ensure that the seconds and fractions are honored during the dump process.

Sample STARTTIME Statement:

To dump 8 hours worth of data from 10 p.m. on August 19, 2008 to 6 a.m. on August 20, 2008:

```
LSNAME(IFASMF.ACCTG)
STORCLAS(WORK)
STARTDATE(2008231)
STARTTIME(2200)
ENDDATE(2008232)
ENDTIME(0600)
```

STORCLAS

If you use SMS to allocate temporary DASD data sets, use this statement to identify the storage class to be used during the allocation of the generated SYSIN data set for IFASMFDL.

Note: Either STORCLAS or UNIT must be specified, but not both. Neither statement has a default so one must be present so that a dynamic allocation request can be made by SMFDLS for disk space for the generation of the input control statements for IFASMFDL.

Statement Format:

STORCLAS

```
▶▶─ STORCLAS(sms storage class) ──────────────────────────────────▶◀
```

Operands and Descriptions:

(sms storage class)

Specifies which SMS storage class is typically used for temporary DASD data sets.

Sample STORCLAS Statement:

To dump the most recent records from logstream IFASMF.ACCTG, and allocate the SYSIN data set on DASD with a storage class of WORKTEMP:

```
LSNAME(IFASMF.ACCT.DATA)
STORCLAS(WORKTEMP)
```

UNIT

If you do not allocate temporary data sets using SMS, the UNIT operand can be used to indicate a DASD unit type or specific device where SMFDLS can dynamically allocate the SYSIN data set for IFASMFDL.

Note: Either STORCLAS or UNIT must be specified, but not both. Neither statement has a default so one must be present so that a dynamic allocation request can be made by SMFDLS for disk space for the generation of the input control statements for IFASMFDL.

Statement Format:

UNIT

```
▶▶─ UNIT(unit name or address) ───────────────────────────────────▶◀
```

Operands and Descriptions:

(unit name or address)

Identifies the unit by the name or address you use for the DASD device where the SYSIN data set for IFASMFDL will be allocated. There is no default.

Sample UNIT Statement:

To dump all recent data for the local system from logstream IFASMF.ACCTG, and allocate the SYSIN data set for IFASMFDL on a 3390:

```
LSNAME(IFASMF.ACCTG)
UNIT(3390)
```

Return Codes from Logstream Dumping

During the interaction between SMFDLS and IFASMFDL, issues can arise. Return codes are provided to assist you in resolving these issues.

Return Codes from SMFDLS and its IFASMFDL exits

- Non-zero return codes from SMFDLS are numbered from 100 to 999.
- Any return code that is less than 100 comes from IFASMFDL, with the exception of return code 4, which can come either from IFASMFDL or from CA SMF Director.
- Return codes of 505 through 599 originate in CA SMF Director itself.
- Return codes under 1000 are from SMFDLS itself, from IFASMFDL, or from the CA SMF Director dump process.
- Return codes from 1000-1999 are from the IFASMFDL Exit 1 program, and 2000-2999 are from the IFASMFDL Exit 2 program.
- The IFASMFDL Exit 3 program does not issue any return codes or abends.

Return Code	Explanation and Action
100	Region error. Increase the REGION on the EXEC statement in the JCL and retry the dump.
110	SMFDLSPR did not open. Make sure it is allocated and set up as a print file and retry the dump.
111	SMFDLSIN did not open. Make sure it is allocated and set up as an input file with LRECL=80 and retry the dump.
120	Syntax error in SMFDLSIN control statements. Messages indicating which statements contain errors will appear in the SMFDLSPR printed report. Correct any errors and retry the dump.
121	Invalid logstream name. Logstream names can be up to 26 characters, consisting of three nodes of 8 alphanumeric or national characters. The nodes themselves must begin with an alphabetic character or national character. Additionally for SMF recording, the logstream name must begin with the string IFASMF. Correct any errors and retry the dump.
122	No LSNAME control statement found and the JCL control statement was also not found in the control statements. Place either of these control statements into the input stream and retry the dump.
123	No UNIT or STORCLAS control statement found. Make sure one or the other is included in the input stream and retry the dump.

Return Code	Explanation and Action
124	Either STARTTIME was found in the control statements without a STARTDATE, or ENDTIME was found in the control statements without an ENDDATE. Correct any errors and retry the dump.
140	Failure to obtain the necessary ENQs for dumping. Contact your systems programmer.
141	Name/token services failure. Contact your systems programmer.
150	SCDS not allocated. Ensure the SCDS data set is allocated to the ddname SCDS. Correct any errors and retry the dump.
160	SCDS failed to open. Ensure the SCDS data set is allocated to the ddname SCDS. If it is, consult the error messages and error code in the resulting IBM messages. Correct any errors and retry the dump.
161	Region error for SCDS record buffer. Increase the REGION parameter in the JCL and retry the dump.
162	An unexpected error occurred when reading the SCDS data set. Contact CA Technical Support. Be prepared to provide a copy of your SCDS data set as well as any printed output.
163	SCDS not up to version level for logstream support. Perform a backup and restore of the SCDS with the latest version of CA SMF Director with all maintenance applied and retry the dump.
164	Configuration not found. There is no configuration in the SCDS for the system that produced the SMF records to be dumped. Define the system to the SCDS using the COMPILE CONFIGURATION control statement, making sure to include the logstream names needed in the DUMPOPTIONS. After the compilation, retry the dump.
165	Configuration has no logstreams. There is a configuration in the SCDS for the system that produced the SMF records, but there are no logstreams defined to the configuration. Recompile the configuration and ensure that the needed logstreams are named in the DUMPOPTIONS. After the compilation is complete, retry the dump.
166	Logstream not found. The configuration in the SCDS for the system that produced the SMF records has logstreams defined, but not the logstream named in the LSNAME control statement. After verifying that the logstream name in the LSNAME control statement is correct, recompile the configuration adding the logstream name. After successful compilation, retry the dump.
167	Region error for SCDS record buffer. Increase the REGION parameter in the JCL and retry the dump.

Return Code	Explanation and Action
170	Dynamic allocation failure for the SYSIN data set that will be used as input to IFASMFDL. Check and make sure that the SYSIN DD is not present in your JCL. If it is not present, ensure that the value of the UNIT or STORCLAS contains a valid value for your data center. If all of these check out, contact CA Technical Support and be prepared to provide the return codes and messages produced in the SMFDLSPR printed output. Otherwise correct any errors and retry the dump.
171	SYSIN failed to open for output. After allocating the SYSIN data set, SMFDLS could not open it for output. See the error messages produced by IBM that come with the OPEN failure. Correct any errors and retry the dump.
172	CLOSE failed for SYSIN. The constructed control statements for IFASMFDL were written to the SYSIN file, but the file did not close. See the error messages produced by IBM that come with the CLOSE failure. Correct any errors and retry the dump.
180	Attach failed for CA SMF Director. The main CA SMF Director load module could not be attached by SMFDLS so that the SMF records can be processed. See any IBM error messages, correct any errors found, and retry the dump.
181	IFASMFDL encountered an error in processing and SMFDLS needed to post termination for CA SMF Director as opposed to IFASMFDL User Exit 3. Likely causes for this condition are as follows: ■ A logstream has been defined in a system configuration in the SCDS but does not exist in the SMF configuration for that system. ■ One or more of the CA SMF Director logstream dump process exits are not registered with SMF. Examine the messages that have been produced by IFASMFDL and take appropriate action based on the information in the messages and the message descriptions in the IBM *MVS System Messages* documentation.
400	LOAD failed for IFASMFDL. Ensure that the dump job is being run on a system that is z/OS 1.9 or higher. If the system is z/OS 1.9 or higher, consult with your systems programmer.
401	No logstreams were generated in the SYSIN control statements. Contact CA Technical Support. Be prepared to provide your output as well as a copy of your SCDS data set.
500	CA SMF Director abended during the run. The attached subtask will have diagnostics available and error messages. See the *Message Guide* and the *Systems Programmer Guide*, correct any errors found, and retry the dump.

Return Code	Explanation and Action
505 up to 599	CA SMF Director encountered an error during processing and completed with a non-zero return code. See the output in the CAIPRINT output file as well as the *Message Guide*. Correct any errors and retry the dump.
1001*	Insufficient region for exit processing. Increase the region size and retry the dump.
1002*	Name/token services failure. Contact your systems programmer.
1003*	Corrupt token found. Contact CA Technical Support. Have ready the printed output from your dump job.
1004	Mismatched logstream name. The logstream name indicated in IFASMFDL is not the same as the logstream name indicated in the parameters provided to SMFDLS.
1005*	CA SMF Director early termination. The attached subtask of CA SMF Director terminated. Look at the messages in the CAIPRINT Audit Trail Report and correct any errors; then retry the dump.
2001*	CA SMF Director early termination. The attached subtask of CA SMF Director terminated. Look at the messages in the CAIPRINT Audit Trail Report and correct any errors; then retry the dump.

* Return codes marked with an asterisk are issued as user abends and not as actual return codes from SMFDLS.

Retrying a Logstream Dump

CA SMF Director keeps track of the last SMF record dumped from a logstream, so there is no need to make any major changes to retry a logstream dump. Make any corrections that are needed, and restart the procedure or resubmit the dump job.

Unlike the MAN files, the logstream files do not get emptied when dumping and are more like a continuous queue of data. While it is important to archive data properly, the process of archival is not as urgent as it has been before. The only concern is to make sure that the dump is completed before the records are deleted from the logstream based on the logstream's autodelete policy.

Chapter 6: Online Interface to SCDS

This chapter describes the CA SMF Director Online, an ISPF interface to the CA SMF Director Control Data Set. This online interface lets you view most of the information in the SCDS.

Note: You can only use the online interface if the SCDS format is at CA SMF Director r12 or later.

You can use this online interface instead of batch commands, like LISTC and LISTH. You can browse the history file index (primary or alternate) for any configuration (select MAN File History Files) or logstream/substream. You can see details about the history file, including specific record types contained in that file. You can browse the lists or jump to a specific point in time by entering a date in the Locate Start Date Field when you get to the History File panel.

You can use the other panels to see the following:

- Configuration-specific options (as specified by the DUMPOPTIONS command)
- Stream-specific options (as specified by the STREAMOPTIONS command)
- Site-specific options (as specified by the OPTIONS command)
- SCDS statistics (such as the date and time of the last dump)
- Current tape pool (similar to the LISTH VOLUMES command)
- Free space in the SCDS

To start the online interface, run the SMFDOL EXEC under ISPF. (For information on how to set up the SMFDOL EXEC, see the step that sets up the online interface in the *Installation Guide*.)

When the online interface is started, an introductory panel prompts you to select the SCDS you want to view. When the SCDS is opened, the main menu appears, and you can use its options to navigate through the interface. Every panel in the online interface has a help panel with complete information.

Note the following about using the online interface:

- The SCDS is opened with shared ENQ, so that dumps to it and extracts from it are not blocked when you are viewing it online.
- Utility functions such as BACKUP, RESTORE, ADDX, DELETEX, and UPDTX are blocked, since they require exclusive access to the SCDS.
- Only one SCDS can be opened at a time, but multiple instances of the online interface can be opened with the same SCDS.

Index