# CA SMF Director®

## Systems Programmer Guide

### Release 12.6.00

# CA Technologies Product References

This document references the following CA Technologies products:

- CA Common Services for z/OS
- CA MII Data Sharing
- CA MIM™ Resource Sharing
- CA MICS® Resource Management
- CA JARS® Resource Accounting
- CA Service Desk
- CA ACF2™ for z/OS
- CA Top Secret® for z/OS
- CA 7® Workload Automation
- CA Auditor for z/OS

# 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 1: System Description

CA SMF Director is an automated facility for dumping, storing, and extracting SMF data for all z/OS environments. It performs automatic dumping and clearing of the SMF SYSx.MANx data sets, as well as archival dumping of SMF logstreams. It also allows you to access SMF data by date, time, SMF ID, record type, and logstream name, without knowing where the SMF data is located. It performs file management for dumping SMF records, as well as logging or inventory for remote tapes containing SMF data.

CA SMF Director provides the functions available with the IBM SMF Dump program IFASMFDP, and enhances them to more effectively manage SMF data.

## SCDS File

CA SMF Director stores the SMF data inventory as well as its processing options on a Direct Access file called the SMF Control Data Set (SCDS). When the SMFD program is used to dump and clear the SMF SYSx.MANx data sets or archive the SMF logstreams, an index entry is created within the SCDS by date, cross referencing the location of the SMF data to the newly created history file. This product also stores an optional pool of tape volume serials on the SCDS that are used to store the SMF data history files. The size of the SCDS primarily depends upon how often SMF dumps occur and how long the inventory of SMF data is to be kept. For further details on how to calculate the size of the SCDS, see the *Installation Guide* or Chapter 4 of this guide.

CA SMF Director also provides initialization, backup, restore, and rebuilding functions of the SCDS. The LISTC and LISTH control statements can be used to report on the contents of the SCDS.

If the SCDS is to be shared by two or more LPARs, all LPARs sharing the SCDS must either belong to the same Global Resource Serialization (GRS) ring or be connected to each other via CA MIM. If CA MIM is to be used, you must define a QNAME of CAIMSMF for CA SMF Director. For more information on QNAME definitions, see the *CA MII Data Sharing for z/OS Systems Programmer Guide*.

**Note:** CA SMF Director includes an ISPF-based online interface named SMFDOL, which lets you examine information in the SCDS file. See the *Installation Guide* for information on setting up SMFDOL and the *User Guide* for information on using SMFDOL.

# The SMFD Program

SMFD is an authorized program that acts as an intelligent dispatcher for CA SMF Director. It does the following:

- tests the syntax of commands and control statements
- calls other CA SMF Director programs to perform these tasks:
  - dumping SMF data sets and archiving SMF logstreams
  - creating an inventory of the SMF data
  - printing and/or extracting SMF data
  - reporting on processing options
  - reporting on SMF inventory
  - backup, restore, initialize, add, delete, and update utility functions

The SMFD program accepts specific processing options from the PARM keyword of the EXEC JCL statement. These options are BACKUP, DUMP, FORCE, IGNORE, INIT, NOSTAE, READ, RESTORE, PERMIT and NODUPS.

# The SMFDLS Program

SMFDLS is an authorized program that acts as a wrapper for the IBM SMF logstream dump utility IFASMFDL. The program also interfaces with SMFD by attaching it as a subtask when archiving data from logstreams. It does the following:

- Tests the syntax of the control statements for SMFDLS (which are different from SMFD and are placed in a different data set).
- Invokes the IBM SMF logstream utility IFASMFDL to read data from SMF logstreams. IFASMFDL will be called with specific user exits that will process the records from the logstreams.
- Attaches SMFD to perform the dump process for the records in the logstream and create history files, dump indexes, and so on.

**Note:** SMFDLS is only used at dump time for the archival of logstream records. The SMFD program handles all other functions for the archived SMF data, whether or not it came from a logstream or MAN file.

# PARM Options for SMFD

Specify these options using the standard JCL PARM parameter on the EXEC statement for SMFD. If multiple options are needed, specify them within quotes separated by a comma, like this:

```
PARM="READ,PERMIT"
```

**BACKUP**

causes this product to copy the contents of the SCDS file to a sequential file, for a subsequent restore of the contents of the SCDS file.

**DUMP**

indicates that the contents of one or more SMF files are to be written to tape or DASD. (This option performs the same function as the DUMP command.)

**FORCE**

avoids the CAF1232E message and user ABEND 3001 if records with SIDs other than the SID being dumped are encountered in a MAN file, just like the IGNORE parameter. Unlike the IGNORE parameter, FORCE causes the records to be written to the history file anyway.

When the FORCE option is in effect, the location of all nonconforming SMF records is unknown to CA SMF Director and must be tracked manually.

**Note:** FORCE and IGNORE are mutually exclusive parameters. If both are specified, CA SMF Director issues error message CAF0011T and terminates with return code 12.

**IGNORE**

avoids the CAF1232E message and user ABEND 3001 if records with SIDs other than the SID being dumped are encountered in a MAN file. If this parameter is specified, CA SMF Director continues dumping, but all the records with a different SID are ignored.

**Note:** This parameter is not recommended, because it may cause a data loss. Use it only as a workaround solution when a MAN file accidentally contains records from multiple systems.

**Note:** FORCE and IGNORE are mutually exclusive parameters. If both are specified, CA SMF Director issues error message CAF0011T and terminates with return code 12.

**INIT**

causes this product to reinitialize the SCDS file, that is, to erase all Configuration definitions, all data set indexes, etc., from the SCDS file. (Obviously, this option must be used with extreme caution.)

**NODUPS**

This PARM can be used with PRINT/EXTRACT processing when overlapping SMF data has been detected. If overlapping index entries are detected, message CAF2411E is issued and processing ends with a return code of 8. If it is determined that there is duplicate data, use the NODUPS option to skip data in the overlap condition. Specifying PARM NODUPS causes CA SMF Director to continue processing data in overlapping index entries, while writing out records that will not be in the overlap condition.

**NOSTAE**

prevents this product from establishing a recovery environment when it starts up. That is, the diagnostic facilities of the system are turned off so that an OS dump is taken if CA SMF Director abends. (This option may be requested by CA Technical Support to aid in solving a problem.)

**PERMIT**

allows this product to PRINT/EXTRACT SMF data from history files even for SMF files that may overlap each other by date and time. If overlapping index entries are detected and the PERMIT or NODUPS PARM has not been specified, message CAF2411E is issued and processing stops with a return code of 8. If it is determined, upon examination of the data, that there are no duplicate records, the PERMIT option should be used to create the output containing all of the input data.

**Note:** Specifying this parameter may result in duplicate SMF data in the output data set when an overlap condition is detected.

**READ**

causes this product to access the SCDS file in read-only mode.

**RESTORE**

causes this product to perform a restore of the SCDS file from a previously-taken backup file.

# PARM Options for SMFDLS

There are no PARM options for SMFDLS.

# SMF MAN File Dump Processing

SMF MAN file dump processing can be initiated in the following ways using CA SMF Director:

- The computer operator can issue the z/OS START CASFDUMP command from the console.
- Through a batch job that invokes the CASFDUMP procedure.
- Automatically by CA Common Services for z/OS CAIRIM whenever an SMF data set is switched or becomes full.

To allow CA SMF Director to automatically dump a full or switched SMF data set, add the CAIRIM control statement provided in the CASFPARM data set member CASFRIMP to the CARIMPRM member used by the CAS9 procedure that executes CAIRIM. CAIRIM dynamically installs a module that receives control in a similar way to the SMF exit IEFU29. To deactivate this facility, modify the CAIRIM control statement by changing the "ENABLE" keyword to DISABLE and rerunning the CAS9 procedure.

See the *Installation Guide* for more details on CAIRIM and CA LMP.

If you want a procedure name other than CASFDUMP or you want to change the command issued to initiate the SMF MAN file dump process, you can do this with a user exit. A sample user exit, SMFIDTBL, is provided in the CASFSRC library and can be used as a guide for altering the command that is issued by the CA SMF Director MAN file switch process. The exit should be assembled and linked as a separate load module and placed in the system link list (see SMFIDTBL sample JCL in CASFJCL). The exit can have any valid load module name.

The user exit for automated SMF MAN file dump processing is called with the following registers:

**R13**

Calling Save Area. The calling registers must be saved here and restored when the exit program returns to the caller, with the exceptions noted below.

**R14**

The return address to branch to when returning.

**Note**: The addressing mode (AMODE) is not set in the register. The return should be made with a BR R14 instruction.

**R15**

The entry address of your exit.

**R1**

A parameter list with the following parameters:

**+0**

Address of the name of the SMF MAN file to be dumped.

**+4**

Address of the start command to be Issued.

**+8**

Address of a 512 byte work area provided to your exit.

The exit may modify the start command to be issued to initiate the dump process, and perform other processing as needed.

Upon return from the exit, registers 0 through 14 should be returned with the same values they had upon entry. Register 15 should be set to the return code from the exit that will indicate what action the automated dumping process is to take:

**0**

Issue the start command.

**12**

Do not issue the start command.

The automated MAN file dumping exit program includes a prefix that contains the command that is to be issued when dumping as well as a list of pointers to the command and to the actual exit code. To see the layout of this prefix, consult the SMFIDTBL sample in your CA SMF Director samples library.

The exit can be assembled and linked with the SMFIDTBL sample JCL provided in the JCL samples library.

Once you have coded your exit and linked it, you must make it available to CA SMF Director. To do this, change the CAIRIM parameter in CASFRIMP to read "ENABLE,MX=(*exitname*)" where *exitname* is the name of the load module for your exit. This will tell the intercept to invoke your exit before issuing the command to start the dump process.

Once CA SMF Director is invoked to perform dump processing, SMFD does the following:

- determines the SMF data set to be dumped and cleared
- selects the tape volume serial to be used if magnetic tapes are to be used
- dumps the data from the SMF data set and creates a new history file
- creates split files containing subsets of the dumped data if there are SPLIT control statements present
- records an index entry for the newly created history file on the SCDS
- clears the SMF data set for reuse by SMF
- deletes any expired inventory/index entries if the autodelete feature is enabled
- creates a duplicate copy of the newly created history file (duplexing) if enabled
- writes selected SMF records to split files that can be used immediately following the DUMP process, if SPLIT commands are present in the control statements

CA SMF Director verifies the SMF Record Header section according to the definition in the IBM z/OS *System Management Facilities* manual. This product requries that all SMF records adhere to the published standard. Other applications or program products may write SMF records that do not conform to this standard. When detected by this product, these records may be rejected by the SMFD program or cause unpredictable results.

The following illustrates the audit trail from DUMP Processing.

```
CA                                          AUDIT TRAIL REPORT                          SMFD       PAGE     1
CA SMF Director       r12 SPn                                                         SYSA      10 JUL yyyy  10:36:24


CAF2207I  SMF DATA WILL BE DUMPED FROM MVSXE44.SF16.MAN1 ON VOLUME MVXE44
CAF2218I  SMF DATA USING CONFIGURATION XE44 IS BEING PROCESSED
CAF2401I  PRIMARY DUMP IS TO TAPE VOLUME: 107652 FILE:1 DSN: SYS3.SMFDATA.P1071098.T1036115
CAF2406I  XE44 SMF DATA WAS FROM 04/16/yyyy AT 13:15:00 TO 04/16/yyyy AT 13:29:59
CAF2406I  ALL SMF DATA WAS FROM 04/16/yyyy AT 13:15:00 TO 04/16/yyyy AT 13:29:59

CAF3300I  MVSXE44.SF16.MAN1 HAS BEEN RESET
CAF2204I  DUMP COMPLETE, 8778 RECORDS COPIED


CAF2900I  RUN COMPLETE
```

```
CA                                        AUDIT TRAIL REPORT                          SMFD       PAGE     2
CA SMF Director      r12 SPn                                                          SYSA      10 JUL yyyy  10:36:27

                                          SUMMARY OF RECORDS COPIED

            TYPE   AVGL   MAXL      TOTAL   ............................. SUBTYPE COUNTS .............................
              17    100    100         14
              18    144    144          8
              21     72     72          4
              32    230    236          2      1 -         2
              41    172    172          1
              42    198    212          6      2 -         6
              62    176    176         14
              99    188    246      5,146      3 -       440       4 -       704       5 -     4,002
             100    230    230          1
             118    128    128         15    118 -        15
             128    163    163        129
             187    111    177        123
             248    156    156          2
             252     87     87          5
             254    162    196      3,284  54465 -       179   54467 -     3,105
             255    182    204         24      1 -        15       7 -         8       8 -         1
             ALL    176    246      8,778
```

CA SMF Director automatically determines which SMF data set should be dumped. If you want the SMF data to be dumped automatically when a data set is switched or becomes full, we recommend that the CASFDUMP procedure *not* reference the SYSx.MANx files. This product examines the z/OS control blocks and dynamically allocates the SMF data sets defined. This involves the procedure not having any 4-character DDname beginning with MAN as the first three characters.

If there is a need to point to another set of SMF data sets, they should be specified as follows in the CASFDUMP procedure:

```
//MANx  DD DISP=SHR,DSN=SYS1.MANx
```

where:

x represents any character A-Z, or 0-9. When scheduling a job to invoke the CASFDUMP procedure to dump the SMF data, the currently active SMF data set must be switched before the SMF data can be unloaded (see the z/OS SWITCH SMF command).

Here is a sample of the provided CASFDUMP procedure:

```
//CASFDUMP PROC A='*',                 /* SYSOUT Class
//         SCDDSN='CAI.CASMF.SCDS',    /* SCDS File
//         STEPLIB='CASFLOAD',         /* Load Library
//         PRIUNIT=TAPE,               /* Primary Pool Device
//         ALTUNIT=TAPE,               /* Alternate Pool Device
//         PRIDS='VOL=PRIVATE',        /* Dummy/Use Primary
//*        CTLLIB=CASFMAC,             /* Control Statement Library
//*        DPCTL=SMFDDUMP,             /* Dump Process Ctl Statements
//         ALTDS=DUMMY                 /* Dummy/Use Alternate
//*********************************************************************
```

```
//* PROCEDURE:   CASFDUMP
//* DESCRIPTION: CA-SMF Director utility procedure to unload the
//*              z/OS system SMF files (SYSx.MANy), allowing the
//*              CA-SMF Director Utility to manage SMF data storage
//*              and extraction.
//* REFERENCE:   DSN=CASFPROC(CASFDUMP)
//* NOTES:
//* (1) Optionally, you may specify the names of the SMF data sets
//*     to be dumped and cleared as shown below.  We do not
//*     recommend this, however.  Instead, we recommend that the
//*     CA-SMF Director utility be allowed to dynamically allocate
//*     the SMF data sets defined to your system.
//*
//* (2) If the DYNAM customization option and dumping to tape is
//*     used, comment out the HISTORY1 and HISTORY2 DD statements
//*     below, as well as the PRIDS, ALTDS, PRIUNIT, and ALTUNIT
//*     symbolic parameters above.
//*     If the DYNAM customization option and dumping to DASD is
//*     used, specify SPACE parameters in HISTORY1 and HISTORY2 DD
//*     statements.
//*
//* (3) If the primary pool is to be used for SMF history files with
//*     the NODYNAM customization option, specify:
//*     PRIDS='VOL=PRIVATE'.
//*
//* (4) If duplexing of the primary pool is desired and the NODYNAM
//*     customization option is used, specify:
//*     ALTDS='VOL=PRIVATE'.
//*
//* (5) If you employ tape history files, uncomment the first UNIT
//*     parameter for each history DD (i.e., the one with the
//*     DEFER option).
//*
//* (6) If you employ DASD history files, uncomment the second UNIT
//*     parameter for each history DD (i.e., the one without the
//*     DEFER option).  You will also need to add a space parameter
//*     to the DASD History files.
//*
//* (7) If you need to have more control statements during the dump
//*     process, such as if you are writing to Split Files, then
//*     comment the DUMP1 EXEC statement that has PARM=DUMP coded
//*     and uncomment the DUMP2 EXEC statement.  Also see note (8)
//*     below for further changes that need to be made to this
//*     procedure to indicate the location of the control statements.
//*
```

```
//* (8) The SYSIN DD is required if the PARM=DUMP is not coded on
//*     the EXEC statement.  If you choose to create Split Files,
//*     for example, you will need to put all of the control
//*     statements into the dataset indicated in the DD.  You will
//*     need to change the parameters on the PROC statement above to
//*     point to the control statement library and member within the
//*     library that contain the control statements.  If a
//*     sequential dataset is prefered to a library, change the
//*     DSN and remove the member name reference.
//*
//* (9) For further details regarding customization options, please
//*     refer to the CA SMF Director User Guide and Systems
//*     Programmer Guide.
//*
//*(10) After you have modified it, you may copy this procedure to a
//*     PROCLIB of your choice, or you may use it instream.
//*     Alternatively, you may add the CAI common procedure library
//*     to the system PROCLIB concatenations.
//*********************************************************************
//*
//DUMP1   EXEC PGM=SMFD,REGION=0M,PARM=DUMP     <<- See Note 7.
//*DUMP2   EXEC PGM=SMFD,REGION=0M              <<- See Note 7.
//STEPLIB  DD  DISP=SHR,DSN=&STEPLIB.
//SYSPRINT DD  SYSOUT=&A.
//SYSXDIAG DD  SYSOUT=&A.
//SYSOUT   DD  SYSOUT=&A.
//SCDS     DD  DISP=SHR,DSN=&SCDDSN.
//HISTORY1 DD  &PRIDS.,DISP=(,KEEP),             <<- See Notes 2 and 3.
//         UNIT=(&PRIUNIT.,,DEFER)               <<- See Note 5.
//*        UNIT=&PRIUNIT.                        <<- See Note 6.
//HISTORY2 DD  &ALTDS.,DISP=(,KEEP),             <<- See Notes 2 and 4.
//         UNIT=(&ALTUNIT.,,DEFER)               <<- See Note 5.
//*        UNIT=&ALTUNIT.                        <<- See Note 6.
//*SYSIN    DD   DISP=SHR,
//*              DSN=&CTLLIB(.&DPCTL).           <<- See Note 8.
//*MANX     DD  DISP=SHR,DSN=SYS1.MANX          <<- See Note 1.
//*MANY     DD  DISP=SHR,DSN=SYS1.MANY          <<- See Note 1.
//*
```

The parameters specified for the CASFDUMP procedure depend on the options specified for the system configuration defined to CA SMF Director.

**SCDDSN**

Specifies the name of the control data set.

**A**

Specifies the JES SYSOUT class to be used.

**STEPLIB**

Specifies the name of the Load Library containing this product.

**ALTDS**

Allows for allocation of the alternate or duplex history file. If duplexing is not used, or the DYNAM complex-wide customization option is enabled, specify ALTDS=DUMMY. Otherwise specify ALTDS='VOL=PRIVATE'.

**ALTUNIT**

Specifies the name of the device to be used for history files in the alternate pool.

**PRIUNIT**

Specifies the name of the device to be used for the primary history file data sets.

**PRIDS**

Allows for allocation of the primary history file. If the primary history file is not used or the DYNAM complex-wide customization option is enabled, specify PRIDS=DUMMY. Otherwise specify PRIDS='VOL=PRIVATE'.

Sample JCL members (loaded during the install process) CASFDMP1 and CASFDMP4 illustrate 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
```

Optionally, CA SMF Director can also write records out to individual files based on the record type, subtype, and other selection criteria using the SPLIT command in the control statements. To do this, the DD statements for these files must be in the DUMP procedure JCL if the TIME operand is not included in the SPLIT statement. The files should be defined the same way as the output files for an EXTRACT process.

If the TIME operand is included in the SPLIT statement, there is no need to code the ddname in the JCL. The files will be allocated using the attributes in the other TIME operands on the SPLIT statement. For more information on the SPLIT command, see the *User Guide*.

If 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 this product (see the next two sections for further details). In addition, if CA Service Desk is installed and configured, CA SMF Director can open requests to call attention to issues during the dump.

# SMF Logstream Dump Processing

## Overview

SMF logstream dump processing can be initiated in the following ways using CA SMF Director:

- By the computer operator, who can issue the z/OS START CASFDUML command from the console.
- Through a batch job that either invokes the CASFDUML procedure, or executes SMFDLS. This can be started either by a user or by an automated scheduling product such as CA-7.
- Automatically by CA Common Services for z/OS CAIRIM whenever a SWITCH SMF (I SMF) command is issued

To allow CA SMF Director to automatically dump a logstream, add a modified version of the CAIRIM control statement in the CASFPARM data set member CARIMPRM to the CAIRIMPRM member used by the CAS9 procedure that executed CAIRIM. The modification needed to set up automatic logstream dumping is made by changing the value of ENABLE to ENABLELS. CAIRIM dynamically installs a module that receives control in a similar way to the SMF exit IEFU29L. To deactivate this facility, modify the CAIRIM control statement by changing the ENABLELS keyword to DISABLE and then rerun the CAS9 procedure.

**Note:** ENABLELS will establish active exit intercepts for both the IEFU29 and IEFU29L SMF exits. This is because z/OS will switch back to recording SMF data to MAN files if there is a catastrophic failure of the logstreams. To learn more about how automatic dumping of MAN files is done, see the previous section in this chapter entitled SMF MAN file Dump Processing.

If you want a procedure name other than CASFDUML, we provide a user exit. This user exit allows you to change the name of the automated dump procedure. In addition, the exit can also be used to suppress a dump for one or more logstreams if multiple logstreams are in use. If the exit is not coded, the default command issued is:

```
S CASFDUML,LS=logstream.name
```

The logstream name is then passed to CASFDUML so that SMFDLS can process the logstream and archive the data. This command is issued once for each active SMF logstream unless the exit suppresses the command.

## Logstream Dumping Exit Registration

z/OS 1.12 and above (and z/OS 1.9 through 1.11 with maintenance) require that all user exits called by the IBM SMF utility programs IFASMFDP and IFASMFDL be registered. The registration is done with the SMFDLEXIT and SMFDPEXIT parameters in the SMFPRMxx member in PARMLIB.

CA SMF Director makes use of IFASMFDL when running the dump process if SMF logstream recording is being used. To control IFASMFDL, CA SMF Director makes use of the three user exit points in IFASMFDL and requires that the exits be registered.

The three exits are as follows:

**SMFDLX1**

Used as User Exit 1 in IFASMFDL

**SMFDLX2**

Used as User Exit 2 in IFASMFDL

**SMFDLX3**

Used as User Exit 3 in IFASMFDL

To ensure that these exits are allowed to be called by IFASMFDL, the following statement must be in the active SMFPRMxx member in PARMLIB on z/OS systems that have the SMFDLEXIT parameter available:

```
SMFDLEXIT(USER1(SMFDLX1),USER2(SMFDLX2),USER3(SMFDLX3))
```

The values in the USER1, USER2, and USER3 subparameters can also contain other exits that may be needed at your site when running IFASMFDL. The only requirement by CA SMF Director is that the names of the exits used in the logstream dump process be present.

CA SMF Director does not require the presence of or changes to the SMFDPEXIT parameter.

For more information on the SMFDLEXIT parameter, including the coding rules, see IBM's *MVS Initialization and Tuning Reference* guide.

## Logstream Dump Process User Exit

This exit has two purposes:

- It can alter the command that is going to be issued by the IEFU29L exit intercept. This way the command to start the automated dump procedure will be tailored to your data center's needs.

  **Note:** If the command length is altered, you are responsible for changing the value of the last byte of the command to be issued in the parameter list.

- It can be used to suppress some or all of the START commands that will be issued through the use of return codes.

This exit lets you tailor the command that will be issued to the system to initiate automatic SMF logstream dump processing. The exit is invoked in AMODE 31, and must remain in AMODE 31 for the entire time of operation. The exit follows typical z/OS linkage for entry and return.

The following registers are meaningful on entry to the routines:

**R13**

Calling Save Area. The calling registers must be saved here and restored from here upon return, with the exceptions noted below.

**R14**

Return Address. The address to branch to when exit processing is completed.

**Note:** The AMODE is not set in the address, and return should be done with a BR R14 instruction.

**R15**

The Entry Address. This is the address of the exit.

**R1**

Four Fullword Parameter List:

Register 1 points to this four-word parameter list:

**+0**

Address of the first byte of the command to be issued. The command is:

```
S CASFDUML,LS=logstream.name
```

**+4**

Address of the last byte of the command to be issued. If the command is altered by the exit and the length changes, it is the responsibility of the exit to update the value of this parameter with the new address of the last byte of the command.

**+8**

Address of the last byte of the command buffer (in case you wish to lengthen the command.)

**+C**

A 256 byte work area for the exit's use. It is expected, but not required, that the exit will use this as a save area and work area.

The return codes are as follows:

**Note:** If a return code is not listed, follow the directions for return code 0.

**0**

Issue the command that is in the buffer

**4**

Do not issue the command in the buffer, but call with the next logstream name, if any.

**8**

Do not issue the command in the buffer, and do not call the exit again.

**12**

Issue the command in the buffer, and do not call the exit again.

**16**

Terminate intercept processing. The exit will not be called again, and no further START commands will be issued for any subsequent logstreams

**Note:** If a return code of 8 or 12 is set, a START command will be issued to initiate automated dump processing for all remaining active logstreams that have not yet been processed by the intercept. If you do not want to have any more commands issued, you must use a return code of 16.

A sample exit named SMFLSTBL is provided in the CASFSRC library. The sample can be used as is, or it can be modified as needed. Once you have constructed the exit, it must be assembled, linked, and placed into a link list library (see LSTBLAL sample JCL). To activate the exit, modify the PARM settings for automated dumping as indicated in the CASFRIMP sample from “ENABLE” to “ENABLELS,LX=(*exitname*).”

## SMF Record Processing

Once SMFDLS is invoked to perform the logstream archive processing, it does the following:

- Determines which logstream is to be dumped and where to begin dumping the logstream.
- Generates control statements for IFASMFDL, the IBM utility that is used to read SMF data from logstreams.
- Attaches SMFD, which will process the records and build history files and dump indexes, for the data copied from the logstream.
- Calls IFASMFDL, passing the built control statements and using three IFASMFDL user exits to control which records are copied to SMFD, as well as to manage access to the logstream.

SMFD has been modified for the logstream environment to process as a subtask to SMFDLS. The process by which records are archived in history files and indexed is different: the dump index entries for logstreams are managed by both SMF ID and logstream name.

At this point, the processing of the logstream SMF data is very similar to the MAN file processing, with the exception that the data is not being read in directly by SMFD, but rather it is passed to SMFD by an IFASMFDL user exit. The history records are written to history files and archived; then the index is updated just as before.

The following is a sample of the CASFDUML 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.
```

The variables in the procedure allow a SYSOUT class to be set by the START command as well. The LS operand is typically the only one set.

The CAIIN file contains the control statements for SMFD, and the SMFDLSIN file contains the control statements for SMFDLS. Typically, the CAIIN file should only contain the DUMP statement as well as any desired SPLIT statements.

The SMFDLSIN will typically have two statements in this environment:

- JCL that indicates that the logstream name to dump is the data set name on the CAILSNM DD statement.
- Either UNIT or STORCLAS, which indicates a valid unit esoteric or SMS storage class. This control statement tells SMFDLS where to allocate the generated control statements for the IBM IFASMFDL utility.

For more information on the SMFD or SMFDLS control statements, see the *User Guide*.

# Overlapping SMF Data

Overlapping SMF data is detected at SMF dump processing time and a warning message is issued. The index record is written to the appropriate system configuration (SID) or logstream configuration, and the dump process will not abend or stop processing when an overlap or duplicate record condition occurs. At this time you can use the LISTH commands to display data around the time frame of the overlapping data, and use ADDX and DELETEX to "fix" the SCDS index entries.

If no action is taken to correct the SCDS entries after Dump processing, when an EXTRACT or PRINT command is executed that encompasses the overlapped index entries, an error message will be issued and processing will stop. At this point there are three options. Option one is to specify the SMFD PARM "PERMIT", which will cause the EXTRACT output to contain all the data from the overlapping indexes. Use this option with caution and only after you have determined there is no duplicate data in the overlapping index entries. Option two is to specify the SMFD PARM "NODUPS", which will cause the EXTRACT or PRINT process to execute and produce in the resultant output, data that does not overlap. This method produces the desired results in the output but leaves the overlapping index entries in the SCDS file. The major benefit with option two is that no manual intervention against the SCDS file is required. Option three is to use the LISTH command to view index entries around the time frame in question, and manually update the SCDS index entries using ADDX and DELETEX commands.

In addition to the above mentioned remedies for dealing with overlapped data conditions, there is a new complex level configuration option OVERLAP(n). The proper specification of this option can eliminate false detections of overlap conditions by this product. This option specifies a time from zero (no overlap grace time period) to five seconds. If a number other than zero is specified, it is used as the number of seconds of overlap allowed to occur between index entries before an overlap condition is detected. This option was introduced due to the irregular way some applications write out SMF records, which, depending on the time the SMF MANx files are dumped, may trigger false overlap detection without this option specified. See Chapter 4 in the *User Guide* for information on the OVERLAP(n) option.

The following examples illustrate how this product deals with various settings of the OVERLAP complex level options, and the PERMIT and NODUPS execution time parameters.

In the first example, the complex level overlap option has been specified as OVERLAP(0). This means that any data overlaps will be flagged as a warning during the dump process, and flagged as an error during EXTRACT/PRINT processing, unless the PERMIT or NODUPS execution parameter has been specified. Consider that three SMF dumps have occurred for the XAT1 system, with the following FROM and TO date/time ranges:

```
-------- FROM --------        -------- To --------
03/16/2008    14:00:00        03/16/2008  14:00:59 ------>  Dump 1
03/16/2008    14:00:56        03/16/2008  14:01:59 ------>  Dump 2
03/16/2008    14:01:56        03/16/2008  14:02:59 ------>  Dump 3
```

Because the overlap option has been set to zero, the overlaps detected during the dump process will be flagged with warning messages. In this particular instance, the dump process for dump(1) will not produce a warning message at the time the dump occurs because it does not overlap any existing index entries for SID XAT1. Dump(2) and dump(3) will issue the following warning messages because the FROM date/time of their dumps is less than the TO date/time of the previous dump.

```
CAF2417W SMF DATA FOR XAT1 03/16/2008 at 14:00:56 OVERLAPS DATA ALREADY IN INDEX

CAF2417W SMF DATA FOR XAT1 03/16/2008 at 14:01:56 OVERLAPS DATA ALREADY IN INDEX
```

The dumped data is added to the SCDS file as an index entry for the SID being processed (XAT1). Using the LISTH and CHECKIT commands, and upon examination of the data, if it is determined that there is duplicated data, the DELETEX and ADDX commands can be used to update the SCDS to properly reflect the correct index entries. For example, if an index entry is completely duplicated, you can use the DELETEX command to remove the duplicated entry. If there is an overlap, you can DELETEX the overlapped index entry and then manually dump SMF data using the SOURCE DUMP command after eliminating overlapped or duplicated data. Finally, if there is a small overlap and it is determined that there was no duplicate data in the SMF files, you can DELETEX and ADDX the entries back into the SCDS without the FROM and TO date/times being overlapped. If an EXTRACT is run at a later point in time, no error messages will be issued.

A second, and recommended approach, is to leave the index entries as they exist in the SCDS and use the PERMIT or NODUPS options, depending on the outcome of your research into the overlap condition. This approach requires less manual intervention and lets this product create the desired output from EXTRACT processing with very little intervention from the user. Consider an EXTRACT request that encompasses all three of the above dumps:

```
EXTRACT SID(XAT1) FROM(031608 140000) TO (031608 140300).
```

If this EXTRACT command is issued with OVERLAP(0) and without the execution parameter PERMIT or NODUPS specified, all of the data from dump(1) is written to the extract file. Two error messages, one for the dump(2) index entry, and one for the dump(3) index entry are issued.

```
CAF2411E SMF DATA FOR XAT1 03/16/2008 AT 14:00:56 OVERLAPS DATA ALREADY IN INDEX

CAF2411E SMF DATA FOR XAT1 03/16/2008 AT 14:01:56 OVERLAPS DATA ALREADY IN INDEX
```

Data from the dump(2) and dump(3) index entries are not extracted.

If this EXTRACT command is issued with OVERLAP(0), and the execution parm PERMIT specified, warning messages are issued, and all data from all three index entries are written to the extract file. The PERMIT option should only be used when the data has been examined and it has been determined that no duplicate data exists. The warning messages are as follows:

```
CAF2417W SMF DATA FOR XAT1 03/16/2008 AT 14:00:56 OVERLAPS DATA ALREADY IN INDEX

CAF2417W SMF DATA FOR XAT1 03/16/2008 AT 14:01:56 OVERLAPS DATA ALREADY IN INDEX
```

If this EXTRACT command is issued with OVERLAP(0), and the execution parm NODUPS specified, warning messages are issued, and all data in the three dump files not considered to be in an overlap condition are written to the output file. In more precise terms, SMF data in dump(2) from 03/16/2008 at 14:00:56 up until the first SMF record in dump(2), with a start date/time greater than 14:00:59, is discarded. All SMF data in dump(3) from 03/16/2008 at 14:01:56 up until the first SMF record with a start date/time greater than 14:01:59 is discarded. This PARM should be used when it has been determined that the overlapped portion of files does indeed contain duplicate data, and you want to skip data in the overlap condition. The following warning messages are issued in this case:

```
CAF2417W SMF DATA FOR XAT1 03/16/2008 AT 14:00:56 OVERLAPS DATA ALREADY IN INDEX

CAF2417W SMF DATA FOR XAT1 03/16/2008 AT 14:01:56 OVERLAPS DATA ALREADY IN INDEX
```

Another possible condition is an SMF data set that somehow is dumped twice, the second dump creating a complete duplicate index entry. For example, consider that dump(3) data was dumped twice, and the following index entries for SID XAT1 now exist:

```
-------- FROM --------        -------- To --------
03/16/2008    14:00:00        03/16/2008  14:00:59 ------>  Dump 1
03/16/2008    14:00:56        03/16/2008  14:01:59 ------>  Dump 2
03/16/2008    14:01:56        03/16/2008  14:02:59 ------>  Dump 3
03/16/2008    14:01:56        03/16/2008  14:02:59 ------>  Dump 4
```

At the time of the dump processing, a warning message appears indicating that overlapping data exists, but the index entry is added to the SCDS file. At extract time, when a duplicate index entry is detected, the following message is issued:

```
CAF3902W DUPLICATE SMF DATA DETECTED FOR XAT1 03/16/2008 14:02:59 DATA SKIPPED
```

This message is issued regardless of the execution parameters PERMIT or NODUPS. PERMIT and NODUPS have no effect on processing of complete duplicate index entries. No data from a duplicate index entry is written to the output file, and the index entry is skipped.

In the previous examples, the dump(2) and dump(3) index entries overlapped their preceding index entries by three seconds. If the complex level option overlap had been specified as OVERLAP(5), indicating that index entries that overlap other index entries by less than the OVERLAP(N) value should not be flagged as overlaps, no warning or error messages would be received during dump or extract processing, and the PERMIT or NODUPS options would not be needed. If the overlap condition exceeds the overlap option specification, then processing occurs exactly as previously explained. For example, using the same dump files, if the complex level option is specified as OVERLAP(2), all of the above warning and error messages will appear as if OVERLAP(0) has been specified, because the index entries overlapped by three seconds, which triggers the overlap condition.

# Customization Considerations

CA SMF Director has been designed to provide flexibility in managing SMF data. This is accomplished by customizing various processing options. This product saves the information from these processing options on the SCDS. When DUMP processing is performed, these options are retrieved and then used to tailor how this product will manage the SMF data.

Three major categories of processing options can be customized:

- Logstream and substream configuration options are set by one or more STREAMOPTIONS control statements. The STREAMOPTIONS statements appear within a BEGIN/END block and are coded similar to the DUMPOPTIONS, which cover the entire system. Not all of the operands on the STREAMOPTIONS statement need to be coded. The values that are not coded are set on the DUMPOPTIONS statement, or, if not indicated on DUMPOPTIONS, on the global OPTIONS statement.
- System configuration options are set by the BEGIN, DUMPOPTIONS, DUMPTAPES, and END Control Statements. During DUMP processing, this product determines which options are in effect by matching the SMF SYSID within the SMF data to be dumped with the system configuration definition stored on the SCDS. Not all of the operands on the DUMPOPTIONS statement need to be coded. Defaults values are picked up from either the OPTIONS statement or the DUMPOPTIONS statement. The default values for both statements' operands are described in the *User Guide*.
- Complex-wide configuration options are set by the OPTIONS Control Statement. These options control product features and are not specific to any system generating SMF data.

The following table summarizes the functions, their related keywords, and Control Statements:

**Note:** Each configuration can have a setting for the maximum files on a single tape volume on the DUMPOPTIONS statement, but a default value for all configurations can also be set on the OPTIONS statement.

| Function | Keyword | Control Statement |
|---|---|---|
| Dynamically allocate history files to reduce JCL complexity and/or dependency | DYNAM * | OPTIONS<br>DUMPOPTIONS |
| Automatically delete expired SMF history file index entries | AUTODEL | OPTIONS |
| Define what storage media to use for SMF history files by:<br>■ specifying esoteric device name<br>■ specifying IBM DFSMS storage class name | PDEVN, ADEVN<br>PSTORC, ASTORC | OPTIONS<br>DUMPOPTIONS<br>STREAMOPTIONS |
| Set the time when a logical day begins for SMF data extraction | SDAY | OPTIONS<br>DUMPOPTIONS<br>STREAMOPTIONS |
| Set the maximum number of lines per page for the audit trail | MAXLINES | OPTIONS |

| Function | Keyword | Control Statement |
|---|---|---|
| Report on events that require attention, either immediately or in the near future | SERVICEDESK, SDURL, SDUSERID, SDPASSWORD, SDHIGHSEVERITY, SDLOWSEVERITY, NOSERVICEDESK | OPTIONS |
| Limit the number of data sets on a single tape volume | MAXFILESONVOL | OPTIONS<br>DUMPOPTIONS<br>STREAMOPTIONS |
| Control which magnetic tape volumes to use: | | |
| ■ Using a specific pool of Tape volume serials | ADD, NOSCRATCH | DUMPTAPES<br>DUMPOPTIONS |
| ■ Using any volume serial in Tape Library | SCRATCH * | OPTIONS<br>DUMPOPTIONS |
| ■ Remove volume serials from a pool of volume serials | DELETE | DUMPTAPES |
| Provide a duplicate set of SMF data (duplexing) | COPIES | OPTIONS<br>DUMPOPTIONS<br>STREAMOPTIONS |
| Set how long to retain SMF data | RETPD, EXPDT | OPTIONS<br>DUMPOPTIONS<br>STREAMOPTIONS |
| Specify how long to retain duplicated SMF data | ARETPD, AEXPDT | OPTIONS<br>DUMPOPTIONS<br>STREAMOPTIONS |
| Set a high-level qualifier for history file data set names | PREFIX, APREFIX | OPTIONS<br>DUMPOPTIONS<br>STREAMOPTIONS |
| Catalog SMF history files also in the System Catalog | CATLG, NOCATLG | OPTIONS<br>DUMPOPTIONS<br>STREAMOPTIONS |
| Specify recording technique on dynamically allocated tapes | TRTCH, ATRTCH | OPTIONS<br>DUMPOPTIONS<br>STREAMOPTIONS |
| Indicate which logstreams or substreams are to be managed | LSNAME*, SUBNAME* | DUMPOPTIONS<br>STREAMOPTIONS |

* These keywords must be present and active for SMF logstream or substream processing.

**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. It must be deleted manually.
- 099366 indicates that the data set will never expire.

If you installed CA SMF Director by upgrading from Unicenter CA-JARS SMF Utility, you must specify the following new options to be in effect:

- DYNAM
- AUTODEL
- PDEVN
- ADEVN
- PSTORC
- ASTORC
- SERVICEDESK
- TRTCH
- ATRTCH

If you are recording SMF data to logstreams, the SCRATCH and DYNAM options must be set in the DUMPOPTIONS and OPTIONS control statements respectively.

## AUTODEL Option

During CA SMF Director dump processing, if the AUTODEL complex-wide configuration option is enabled, this product will check the SCDS for any index entries to SMF data history files that have expired. If any is found, it will delete the entry from the SCDS, uncatalog the history file data set if it was cataloged in the z/OS System Catalog, and scratch the data set if the data set resides on a DASD device. If an expired history file resided on DASD but has been archived, CA SMF Director issues an HDELETE request to remove the history file from the archive.

This feature can result in lower storage costs, and keeps the size of the SCDS from increasing at a substantial rate. For the AUTODEL feature to work properly, we recommend that retention periods be used for history files instead of expiration dates. For example, if RETPD(1095) is specified in the DUMPOPTIONS control statement, the history files created in the primary pool will expire 1,095 days after the history file was created. In 1,098 days, (1,095 plus a three day grace period), CA SMF Director will remove the index entry from the SCDS. However, if EXPDT(108.365) is specified on the DUMPOPTIONS control statement, the history files created in the primary pool will expire after December 30, 2008. Any history files created after that date will expire the next day, and the index entry from the SCDS will be removed three days later. Using a permanent retention date such as EXPDT(99.366) will cause this product to retain the history file index entry indefinitely, or until it is manually deleted. If all history files use permanent retention dates, enabling the AUTODEL feature will have no effect.

Because zero is the default retention period, CA SMF Director will not automatically delete any SCDS entry that has a zero retention period. Such entries can only be deleted manually, using the DELETEX command. Likewise, any SCDS entry with an expiration date of 099000, which has a special meaning to tape library management products, can only be deleted manually.

## DYNAM Option

CA SMF Director determines where to place the SMF history files by using the information specified on the JCL statements for the HISTORY1 (primary) and HISTORY2 (alternate) DD names. When the DYNAM option is enabled, it is not necessary to specify the HISTORYn DD statements within the JCL used to invoke this product. The values specified in the PDEVN, ADEVN, or PSTORC and ASTORC keywords are used to determine the storage device for the history files. If the HISTORY1 and/or HISTORY2 DD statement is specified in the JCL with the DYNAM option enabled, the information coded on the JCL will override any values specified with the OPTIONS control statement. For substreams, the DD names are different: the assigned stream code is part of the name. For example, if the assigned stream code for the substream is “SA,” the DD names for the stream when allocated will be HISTSA1 for the primary and HISTSA2 for the alternate.

DYNAM is recommended if the primary or alternate history files will reside on different device types.

If the DYNAM option is to be used on a newly converted SCDS from Release 1.5 of Unicenter CA-JARS SMF Utility, the UPDTX control statement may be used to add a device name to a range of history file index entries on the SCDS. The UPDTX control statement may also be used to change the device type for history file index entries, if it becomes necessary to copy or convert history files from one device type to another while retaining the data set name.

CA SMF Director dynamically allocates the z/OS SMF data sets independently from the DYNAM option specification. If a MANx DD statement is specified, the SMF data sets are not dynamically allocated. This product will dynamically allocate the required SMF history files for extract processing when DYNAM is enabled.

The DYNAMVTS option was designed for data centers that employ virtual tape, but do not want CA SMF Director to perform dynamic allocation of history files. It permits dynamic allocation of the input history files to an EXTRACT or PRINT operation only. This is necessary to avoid problems, such as incorrect mount messages, that can arise at such data centers when NODYNAM is in effect.

**Note:** The DYNAM option must be specified if CA SMF Director is going to manage SMF data that has been recorded to logstreams.

## CATLG Option

When CATLG is coded in the DUMPOPTIONS/STREAMOPTIONS for a SID, all history files that are generated for that SID/stream will be cataloged. When an extract calls for a cataloged DASD history file, it will search the system catalog. This enables an extract to find a history file that has been archived by your site's DASD management system and later restored to a volume other than its original volume.

If AUTODEL is in effect, CA SMF Director will use the system catalog to locate any cataloged DASD history file that meets the expiration criteria. Thus, expired history files will get deleted regardless of the volume on which they reside at the time. They will also get uncataloged. If a DASD history file is archived at the time that it expires, CA SMF Director will simply uncatalog it.

## RETPD/EXPDT Value

The ARETPD and AEXPDT operands of the DUMPOPTIONS/STREAMOPTIONS control statement allow the setting of a retention period or expiration date for SMF history files in the alternate pool that differs from that set for the history files created in the Primary pool (RETPD, EXPDT). If duplexing of SMF data is specified by COPIES(2) with no corresponding retention period or expiration date, this product will use the value specified for the primary pool.

Using a permanent retention date such as EXPDT(99.366) will cause this product to retain the history file index entry indefinitely, or until it is manually deleted. Tapes managed by CA Dynam/TLMS and created with EXPDT(99.000) will be kept in permanent status as long as the data set name remains cataloged. Once the data set is uncataloged, the tape is placed into scratch status and its corresponding SCDS entry is deleted. If the history files are created with EXPDT 99.000 or 99.366, the SCDS must be allocated with enough space to hold all the index entries for these history files, as well as new entries created due to the SMF DUMP process. See Calculating SCDS Space Requirements in Chapter 4.

Retention periods should be used rather than expiration dates, unless the SMF history files are to be kept permanently. If EXPDT(108.365) is specified on the DUMPOPTIONS control statement, the history files created in the primary pool will expire after December 30, 2008. Any history files created after that date will expire the next day and the index entry will be removed from the SCDS three days later.

## VERIFY Option

When VERIFY is specified on the DUMPOPTIONS control statement, the Write Validity Check feature (DCB=OPTCD=W) of the storage device is used, if it is available. CA SMF Director checks the SMF Header section of each record for validity, regardless of the VERIFY/NOVERIFY setting. The SMF Header section is checked according to the definition in the IBM z/OS *Systems Management facilities Manual*. If an error is detected, this product will display the record and attempt to continue processing. Other applications and/or software products running under z/OS may write SMF records having a header section that does not adhere to the published standard. These SMF records may be rejected, or cause unpredictable results within the SMFD program.

## SERVICEDESK Option

If CA Service Desk is installed and available, CA SMF Director can be set up to automatically open Service Desk requests at any instance of Service Desk installed in an enterprise. If the default Service Desk instance is the one where CA SMF Director is to open Service Desk requests, then only the SERVICEDESK operand need be present on the OPTIONS control statement. All of the principal defaults used by Service Desk can be overridden using the other Service Desk operands on the OPTIONS control statement.

**Note:** A NOSERVICEDESK operand is the default; therefore, it need not be specified if you do not want CA SMF Director to issue Service Desk requests.

If Service Desk requests are to be opened, the CA Common Services SOAP Client Service (CAISDI/soap) address space must be up and running, because it is that address space that actually opens the Service Desk requests in Service Desk. In addition CA Common Communications Interface CAICCI must be installed and set up so that the Service Desk API can locate the SOAP Client Service. For more information on the SOAP Client Service, see the *CA Common Services for z/OS Service Desk Integration Guide.* For more information on CAICCI, see the *CA Common Services for z/OS Administrator Guide*.

If the SERVICEDESK operand is the only operand present, the following defaults will be used for making Service Desk requests:

| | |
|---|---|
| High Severity Level | 2 |
| Low Severity Level | 4 |

| | |
|---|---|
| URL of Service Desk Web Service | Value of SD_URL in the SOAP Client Address Space parameters |
| User ID for Service Desk Access | Value of SD_UID in the SOAP Client Address Space parameters |
| Password for Service Desk Access | Value of SD_PASS in the SOAP Client Address Space parameters |

CA SMF Director provides operands on the OPTIONS statement to override these values. These operands require that the SERVICEDESK operand be present; otherwise they will be ignored.

If the Service Desk default request priorities are not what you want, the SDHIGHSEVERITY and SDLOWSEVERITY are provided to set the priorities to the desired values:

- The SDHIGHSEVERITY operand can be set to any value between 1 and 4.
- The SDLOWSEVERITY operand can be set to any value between the value of SDHIGHSEVERITY and 5.
- CA SMF Director can also be set to issue all requests at a single priority by setting the two operands to the same value between 1 and 4 (but not 5).

If CA SMF Director needs to send Service Desk requests to an instance of Service Desk that is not the default instance indicated in the SOAP Server, the SDURL operand of the OPTIONS statement is provided to indicate the Service Desk that should receive the request.

Note that the SDURL value is case-sensitive. Also, since the SDURL that is coded contains special characters, it must be enclosed in single quotes within the parentheses. If the URL is not coded in single quotes, a parsing error will be returned and the COMPILE CONFIGURATION process will fail.

If the Service Desk User ID and Password are different from the User ID and Password indicated in the SOAP Client address space, the SDUSERID and SDPASSWORD operands are available on the OPTIONS statement. Note that both parameters must be present if they are to be used. In addition, note that the values of the User ID and Password are case-sensitive. If special characters are part of the User ID or Password, the operand values must also be enclosed in single quotes.

# Chapter 2: Securing Functions

CA SMF Director can provide SMF and RMF records containing performance and accounting information to anyone. Therefore, it may be necessary to restrict certain functions of this product from certain users.

The functions of this product can be secured by using an external security product such as CA TOP SECRET, CA ACF2, or IBM's RACF. The resource class CA$MSMF must be defined to the external security product prior to using CA SMF Director. IBM RACF customers accomplish this using the ICHERCDE macro to modify the Class Descriptor Table (CDT) and the ICHRFRTB macro to modify the RACF Router Table. CA TOP SECRET customers should modify their Resource Definition Table (RDT) using the TSS ADD(RDT) command. CA ACF2 customers should create a CLASMAP and GSO SAFDEF records. For more details, see the administrator guide for the security product.

The resource class CA$MSMF must allow alphanumeric resource names up to 20 characters in length, and access levels of UPDATE, READ or NONE. CA SMF Director issues a RACROUTE REQUEST=AUTH using the resource class of CA$MSMF, application id of SMFDIR, and an ENTITY name (resource name) consisting of the format: PRM.xxxxxx or CMD.xxxxxx. 'PRM' indicates that the function was requested via the JCL EXEC PARM keyword, whereas 'CMD' represents the function was requested via the SYSIN data set.

We recommend that the default for the CA$MSMF resource class should be to disallow access. To allow use of this function, the following resources must be defined or permitted to the user's security record:

| The Resource name... | Performs the function... |
|---|---|
| PRM.BACKUP | Backup of the SCDS file. |
| PRM.RESTORE | Restore of the SCDS file. |
| PRM.INIT | SCDS Initialization. |
| PRM.DUMP | Dumping of the SMF files. |
| CMD.xxxxxx | of the indicated command statement where xxxxxx is one of the following: ADDX, BEGIN, CHECKIT, COMPILE, DELETEX, DUMPOPTIONS, DUMPTAPES, DUMP, EXTRACT, END, LISTC, LISTH, OPTIONS, SOURCE, SPLIT, TAPEINIT, or UPDTX. |

If the CASFDUMP procedure is started by operators or via the auto dump feature, then the default batch or STC security record must contain permissions to PRM.DUMP and/or CMD.DUMP. If other functions will be performed by the CASFDUMP procedure such as backup or restore, security permissions must also be allowed.

The following example illustrates how a CA TOP SECRET security Administrator might define this product to the security system and allow John Doe to perform PRINT or EXTRACT requests only:

```
TSS ADD (RDT) RESCLASS (CA$MSMF) RESCODE (XX) MAXLEN (20) ATTR (MASK)
 ACLST (READ=4000, UPDATE=8000, NONE=0000, ALL=FFFF)
```

The above CA TOP SECRET command can be used as an example of how to update the Resource Descriptor Table. The resource class is CA$MSMF for CA SMF Director. An available resource code should be selected by the user.

The following CA Top Secret commands define the ownership of resources to an already existing user USER01.

TSS ADD (USER01)  CA$MSMF (PRM)

TSS ADD (USER01)  CA$MSMF (CMD)

The following CA Top Secret commands remove access to all users, and define access for JOHNDOE to the EXTRACT and PRINT commands.

| | | |
|---|---|---|
| TSS PERMIT(ALL) | CA$MSMF(PRM.*) | ACCESS(NONE) |
| TSS PERMIT(ALL) | CA$MSMF(CMD.*) | ACCESS(NONE) |
| TSS PERMIT(JOHNDOE) | CA$MSMF(CMD.EXTRACT) | ACCESS(UPDATE) |
| TSS PERMIT(JOHNDOE) | CA$MSMF(CMD.PRINT) | ACCESS(UPDATE) |

The next example illustrates how a RACF security administrator would define CA SMF Director to RACF. The class descriptor table is used to describe resource classes to be used by this product.

```
ICHERCDE  CLASS=CA$MSMF,              X
      id=128,                         X
      FIRST=ALPHA,                    X
      OTHER=ANY,                      X
      POSIT=25,                       X
      MAXLNTH=20,                     X
      DFTUACC=NONE
```

For the RACF router table enter the following:

```
ICHRFRTB  CLASS=CA$MSMF,              X
       ACTION=RACF
```

Use the RACF RDEFINE command to define all RACF resources belonging to the new classes specified in the class descriptor table.

```
RDEFINE CA$MSMF      (PRM.BACKUP)          UACC(NONE)
RDEFINE CA$MSMF      (PRM.RESTORE)         UACC(NONE)
RDEFINE CA$MSMF      (PRM.INIT)            UACC(NONE)
RDEFINE CA$MSMF      (PRM.DUMP)            UACC(NONE)
RDEFINE CA$MSMF      (CMD.ADDX)            UACC(NONE)
RDEFINE CA$MSMF      (CMD.BEGIN)           UACC(NONE)
RDEFINE CA$MSMF      (CMD.CHECKIT)         UACC(NONE)
RDEFINE CA$MSMF      (CMD.COMPILE)         UACC(NONE)
RDEFINE CA$MSMF      (CMD.DELETEX)         UACC(NONE)
RDEFINE CA$MSMF      (CMD.DUMPOPTIONS)     UACC(NONE)
RDEFINE CA$MSMF      (CMD.DUMPTAPES)       UACC(NONE)
RDEFINE CA$MSMF      (CMD.DUMP)            UACC(NONE)
RDEFINE CA$MSMF      (CMD.EXTRACT)         UACC(NONE)
RDEFINE CA$MSMF      (CMD.END)             UACC(NONE)
RDEFINE CA$MSMF      (CMD.LISTC)           UACC(NONE)
RDEFINE CA$MSMF      (CMD.LISTH)           UACC(NONE)
RDEFINE CA$MSMF      (CMD.OPTIONS)         UACC(NONE)
RDEFINE CA$MSMF      (CMD.SOURCE)          UACC(NONE)
RDEFINE CA$MSMF      (CMD.SPLIT)           UACC(NONE)
RDEFINE CA$MSMF      (CMS.STREAMOPTIONS)   UACC(NONE)
RDEFINE CA$MSMF      (CMD.TAPEINIT)        UACC(NONE)
RDEFINE CA$MSMF      (CMD.UPDTX)           UACC(NONE)
```

When using an external security product to secure the CA SMF Director functions, the security product will either allow or deny access to the resources mentioned above. If the security product denies access to a resource, CA SMF Director will not continue to process the command. A message is issued indicating that authority was denied for the request and a return code of 12 is set.

By default this product requires update access to the SCDS file. Specifying PARM=READ on the EXEC statement calling program SMFD will allow this product to process functions that require read only access. This can be done provided that the external security product will grant access to the SCDS file, any related programs, and the specific function to be processed. Functions that require SCDS Update access will fail with a message that READONLY processing was requested.

In addition to defining the above resources, the security product must allow access to the CA SMF Director programs, the SCDS file and SMF history files. The DUMP function requires create access for the SMF history files, with update access to the SCDS file. Read access to the SMF history files for the EXTRACT and PRINT functions are needed.

The following commands require UPDATE access to the SCDS file. Use these commands to perform the specified actions.

**BEGIN**

start a series of configuration commands.

**COMPILE**

compile the configuration definition.

**END**

end the configuration definition.

**STREAMOPTIONS**

define specifications for SMF history files for a specific stream.

**DUMPOPTIONS**

define specifications for SMF history files for each system.

**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**

generate application-ready SMF files while dumping.

**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.

The following commands require READ access to the SCDS file. Use these commands to perform the specified actions.

**EXTRACT**

copy SMF data to be processed by another product.

**PRINT**

print SMF data.

**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**

to be used as a diagnostic tool used to display SCDS records in dump format.

# Chapter 3: Extraction, Split User Exits, and Split File Time Processing

## Writing EXTRACT Exits

Through the EXIT operand of the EXTRACT command, you can identify user-written exit routines to:

- examine selected SMF records and/or
- pass SMF data to other files.

Such routines are intended for the actual processing of SMF data, not for specifying additional selection parameters. (To add selection criteria, the exit routines themselves must contain the code to analyze and write the desired records.) You may define your exit routines so that they process records selected by more than one EXTRACT statement.

Exit routines must be linked as load modules, making them available for dynamic loading when the CA SMF Director (SMFD) program executes. SMFD loads the exit routines before processing any SMF records. They are resident until SMFD processing is completed.

The SMFD program calls each exit routine once before actually starting to process SMF records. This enables the routines to perform initialization functions, such as opening data sets.

During the processing, each exit routine is called once for each SMF record that satisfies the selection criteria specified in the EXTRACT statements calling for the exit.

After all SMF records have been processed, SMFD calls the exit routines a last time, permitting them to perform summary and cleanup operations.

EXTRACT exit routines are called according to normal OS linkage conventions. The following registers are meaningful on entry to the routines:

**R1**

Points to a four-word parameter list (described on the next page).

**R13**

Points to a standard 18-word OS Save Area, which exits may use to save registers.

**R14**

Contains the return address to SMFD.

**R15**

Contains the address of the exit's entry point.

Register 1 points to this four-word parameter list, where:

**Word 1**

Points to a halfword that contains a value of binary 1.

**Word 2**

Points to the header of the SMF record being passed to the exit. This fullword also signals the call to perform initialization or cleanup functions. If the length field (Word 3) indicates no SMF record (value of zero), this word contains the address of a halfword function code field. A binary value of zero in this field represents an initialization call; binary 1 represents a termination call.

**Word 3**

Points to a halfword field that specifies the SMF record length. A value of zero indicates a start or end call.

**Word 4**

Points to a fullword. Its value is zero on the first call, then remains as set by the exit routine. It is generally used to address an area acquired by the exit.

Upon your return to CA SMF Director, Register 15 must contain one of the following return codes:

**0**

Count the record, and write it if the TODD parameter was coded.

**4**

Count the record, but do not write it.

**8**

Neither count nor write the record.

A sample user exit is provided in the CASFSRC Library with the member name CASFXXIT.

## Specify the TODD Operand

Whenever you specify a user exit in an EXTRACT statement, you should include the TODD operand. The presence of the TODD operand causes CA SMF Director to write the selected SMF records to the file defined in the associated DD statement. (However, the presence of TODD does not prevent you from also having output written directly from the user exit routine.)

If you do not include the TODD parameter, this product assumes that the exit routine itself should produce the output from the selected SMF records. In this case, it would open an output file while servicing the initialization call, and close the output file while servicing the termination call.

# Writing Split Exits

Through the use of the EXIT parameter on the SPLIT statement, you can filter SMF records even more precisely than with the operands indicated on a SPLIT statement. The exit will be called at the following times in the dump process:

- When a SPLIT definition is initialized
- When the split file is opened
- When a record is about to be written to the split file
- When a split file is closed

The primary purpose of the exit is to allow a further examination of records that are to be written to a split file to determine if they should be written to a split file. The exit will be called in primary mode, with no locks held. An exit can be shared by multiple SPLIT statements and split files, but we recommend, if this is done, that the exit be written so that it does not modify itself and that the exit be linked re-entrant. The exit should not obtain any locks, and it should not hold any enqueues that are issued between calls. The exit should be written in assembler language. Other languages and environments may work but are not supported. The exit may provide recovery, but we strongly recommend that the exit recovery only handle errors within the exit and percolate any errors outside its perview. The exit must follow standard 31-bit z/OS linkage conventions. Upon receiving control, the following values are set in the registers:

**R0**

The function call. The register will be set to one of these values:

**00000000**

A split definition is about to be processed for the first time.

**00000001**

The split file has just been opened

**00000002**

A record is about to be written to the split file.

**00000003**

The split file has been closed; the product is terminating.

It is expected that the 00000000 call will be used to allocate any addtional resources the exit needs, and that the file closed call will be used for clean up.

**R1**

A three-word parameter list:

**+0**

Address of the SMF record about to be written, if any. This value is set only if R0 is set to 00000002. The value, otherwise, should be ignored.

**+4**

Address of a 128-byte work area preallocated for the exit's use. If more memory is needed, it can be obtained using standard z/OS memory services. The memory can safely be released when the 00000003 call is made as the exit will not be called after that. Do not release this 128-byte work area at termination time.

**+8**

Address of the DDNAME from the TODD field in the SPLIT statement.

**R13**

Save Area. Upon entry, the exit must save the registers passed to it, and restore them upon return. Failure to do so can produce unpredictable results.

**R14**

The return address and AMODE. The exit must return using a BSM 0,R14 instruction.

**R15**

The entry point and AMODE of the exit. The exit will be called in the AMODE it was linked with.

Upon your return to CA SMF Director, Register 15 must contain one of the following return codes:

**0**

Normal return. If the function is 2, this will allow the record to be written.

**4**

Normal return. If the function is 2, this will suppress the writing of the record.

**8**

This return code will deactivate the exit and, if the function is 2, allow the record to be written.

**12**

This return code will deactivate the exit and, if the function is 2, suppress the writing of the record.

**16**

This return code will deactivate the SPLIT statement and the exit.

For functions 0 and 1, only return codes 0, 8, 12, and 16 are valid. For function 3, the return code is ignored. In all cases, any return code value that differs from one of these indicated values will be treated the same as setting a return code of 0.

# Setting up for Split File Time Processing

To implement split file TIME operand processing during the dump process, you must do the following in advance to ensure that processing works smoothly.

## Establish the Breakpoint Times

Up to four breakpoint times can be specified in the TIME operand that is indicated on the SPLIT statement. It is important that the times that are set up match the business needs of the organization.

**Note:** When setting up times for the SPLIT statements, there is not an automatic switch of split files at midnight. In order to switch the split files at midnight, 8 AM and 4 PM, the TIME operand must be coded as TIME(0000,0800,1600).

The TIME operand must appear on each individual SPLIT statement that is to be processed with the TIME operand.

If interval data is being managed by a TIME-based SPLIT file, you may need to adjust the times that an SMF switch is issued to ensure that the interval data is available in a timely fashion. When CA SMF Director writes TIME-based SPLIT files, it will not indicate that a file is ready for processing until a record for that split file shows up in the next interval that the TIME statement indicates.

**Note:** To ensure that the data is available, make sure that the scheduled times for SMF switch operations are set to the times mentioned in the TIME operand plus the value of the INTVAL in the active SMFPRMxx member, plus one minute. Doing this ensures that data falls into the next interval.

For example, assume a SPLIT statement has the following TIME operand:

```
TIME(0000,0800,1600)
```

This provides three iterations of the SPLIT file per day:

- One has data from midnight to 8 AM.
- One has data from 8 AM to 4 PM.
- One has data from 4 PM to midnight.

Assume also that the active SMFPRMxx member has INTVAL(15) coded so that interval data is generated every 15 minutes.

In this example, you should set the automated SMF switch commands (I SMF) to be issued at 00:16, 08:16, and 16:16 every day. That way, once the dump is completed from the switch, the data from the previous portion of the day will be available right away.

## Create the Generation Data Groups for TIME-Based Split Files

To use the TIME operand with the SPLIT statement, you must establish Generation Data Groups (GDGs) for the split files that are produced. The GDG that is to be used when writing split files is indicated in the required TIMEGDG parameter on the SPLIT statement.

The names of the GDGs begin with an up-to-26 character prefix that is provided by the TIMEGDG operand on the SPLIT statement followed by .SIDxxxx, where xxxx is the SMF ID of the data being dumped.

For example, if TIMEGDG(ACCT.SMFDATA.SOURCE) is coded on a SPLIT statement, data from a system with SMF ID SYSA would be written to data sets in the ACCT.SMFDATA.SOURCE.SIDSYSA GDG. If the data came from a system with the SMF ID set to SYSB, the GDG would be ACCT.SMFDATA.SOURCE.SIDSYSB. Adding on the last node allows a SPLIT statement with the TIME operand to be shared among multiple systems.

## Allocate Split File Indexes (Recommended)

To maximize the power of the TIME operand on SPLIT statements, we recommend that one or more split file indexes be allocated. The TIME operand process uses the index to determine if more records can be written to the current generation of the GDG, or if a new data set needs to be allocated to the GDG. The index must be allocated before it is used.

To determine the size of the index file, you must have the following information:

- The number of SPLIT statements that will use the same index file, and the number of systems that each SPLIT statement will affect. For example, a SPLIT statement with the ALL operand will be run on all systems, while one with a SID(SYSA) operand will only take effect on a system where the SMF ID is SYSA.

- The number of SMF dump processes that will go on in a day. If MAN file switches and corresponding dumps are scheduled regularly, this is easy to determine, but if dumps are only scheduled for when a MAN file fills, this can be harder to predict.
- The number of days of data that need to be kept in the index.
- The number of intervals that are defined in the TIME operands on the SPLIT statements that reference this index.

The number of entries added to the index for a SPLIT statement is the number of TIME operand boundaries crossed, plus 1. For example, if a SPLIT statement is coded with TIME(0000,0800,1600), and the file being dumped spans from 0400 to 1200, two entries will be added to the index. The first entry will indicate that the data set with data up to 0800 is complete, and the second entry will show that there is data in a data set from 0800 to 1200, but that the data set is incomplete, meaning that the next dump may add more data to the data set. If TIMECLOSE is active for this split file as well, the entry containing the last record written to the split file may be marked as complete, and a new entry may be created for use by the next dump process.

Using this, an approximate number of entries can be calculated.

At half-track blocking, a one track data set can hold 290 entries. One cylinder can hold 4,350 entries, which should be sufficient for most data centers.

In order to make processing slightly more efficient, the INDEXLIM operand can be used on the SPLIT statements to indicate how large the index can be. It is not required, but if the parameter is not provided, the dump process assumes that the file can be added to at the end without any restriction and an x37 abend might result.

The data set attributes for the split file index are:

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

## Preallocate a Duplicate Index File (Recommended)

If the split file index is to be used by other applications, we recommend that the INDEXDUP operand be used on the SPLIT statement. The duplicate index file should be a sequential file with the same file attributes as the main split file index. We recommend that you allocate room for up to 10 additional index entries (each is 192 bytes). Once the file is created, it should be allocated with DISP=SHR. Update serialization is provided by CA SMF Director through the use of ENQ processing and must be honored by applications that use the duplicate index. For more details on serializing the duplicate index, see the *User Guide.*

# Chapter 4: Utility and Maintenance Functions

The material presented in this chapter is intended primarily for systems programmers and other personnel responsible for maintaining SMF files.

The SCDS indexes indicate available SMF history data for the configurations that your site has defined. Maintain the SCDS with the following sample JCL:

| To | Use Member |
|---|---|
| Create a new SCDS | CASFNCDS |
| Convert the old SCDS | CASFSCNV |
| Specify SCDS options | CASFCTL1 |
| Back up the SCDS | CASFBKUP |
| Restore the SCDS | CASFRSTR |
| Enlarge the SCDS | CASFNLRG |
| Correct the SCDS | CASFSUTL |
| Sample SMF dump JCL | CASFDMP1 and CASFDMP4 |

You can execute these sample JCL members after CA SMF Director has been installed. See the CASFJCL target library loaded during the installation process. We also show you how to dump the SMF if, for some reason, this product cannot do so.

## Creating the SCDS File

During the installation process, sample member CASFNCDS can be used to create the SCDS file. If you installed a previous release of CA SMF Director, you can convert the existing SCDS for use with this release. Use the sample member CASFSCNV to convert the file by

- backing up the existing SCDS file
- allocating a new SCDS file for this release
- restoring the contents of the existing SCDS into the new SCDS.

**Note:** Record conversion takes place during the restore process.

For more information on SCDS space requirements, see Calculating SCDS Space Requirements later in this chapter.

The following is a sample of the JCL provided to allocate and initialize the SCDS file:

```
//CASFNCDS JOB (ACCTINFO),PGMR,CLASS=A,MSGCLASS=A
//*
//* THIS SAMPLE JCL ALLOCATES AND INITIALIZES A SCDS FILE
//*
//* NOTE: EDIT AS PER INSTALLATION STANDARDS
//*
//* REFERENCE: SEE CHAPTER 4 IN YOUR CA SMF Director INSTALLATION
//*            GUIDE UNDER 'CALCULATING SCDS SPACE REQUIREMENTS'
//*
//STEP1    EXEC PGM=SMFD,REGION=2M,PARM=INIT
//STEPLIB  DD DISP=SHR,DSN=CASFLOAD                  <-- EDIT
//SCDS DD  DSN=CAI.SMFDIR.SCDS,                      <-- EDIT
//         DISP=(,CATLG),UNIT=SYSDA,
//         SPACE=(4096,(400)),VOL=SER=??????,        <-- EDIT
//         DCB=(RECFM=F,LRECL=4096,BLKSIZE=4096,DSORG=DA)
//SYSPRINT DD SYSOUT=*
//SYSXDIAG DD SYSOUT=*
/*
//
```

# Backing Up the SCDS

Because the SCDS file contains the index to the SMF data in your history files, you should back it up often -- preferably once a day. This way, if a problem should arise in this file, you are able to restore the SCDS with a minimal loss of information.

The provided sample JCL member CASFBKUP is illustrated below:

```
//CASFBKUP JOB (ACCTINFO),PGMR,CLASS=A,MSGCLASS=A
//*
//*THIS SAMPLE JCL PERFORMS A BACKUP OF CA SMF Director
//*CONTROL DATA SET (SCDS).
//*
//*NOTE: EDIT AS PER INSTALLATION STANDARDS
//*
//STEP1    EXEC PGM=SMFD,REGION=2M,PARM=BACKUP
//STEPLIB  DD DISP=SHR,DSN=CASFLOAD
//SYSPRINT DD SYSOUT=*
//SYSXDIAG DD SYSOUT=*
//SCDS     DD DISP=SHR,DSN=CAI.SMFDIR.SCDS
//SCDSBACK DD DSN=CAI.SMFDIR,SCDSBKUP(+1),UNIT=CART,
//            DISP=(NEW,CATLG,DELETE)
/*
//
```

Note that the new SCDS backup data set is sequential and can reside on tape or disk. However, it must be backed up by the SMFD program rather than by other file copy utilities.

The following is an example of the audit trail report produced by executing the CASFBKUP sample JCL:

```
CA                          AUDIT TRAIL REPORT          SMFD  PAGE   1
CA SMF Director r12 SPn                     SYSA  22 DEC yyyy 17:52:19

CAF53I SCDS BACKUP COMPLETE, 24 BLOCKS COPIED:
CAF54I 3 CNFC RECORDS
CAF54I 1 CNFA BLOCK
CAF54I 1 CNFN BLOCK
CAF54I 1 CTLR BLOCK
CAF54I 123 DPIX RECORDS
CAF54I 1734 DSIR BLOCKS
CAF54I 536 FREE BLOCKS
CAF54I 1 FSPR RECORD
CAF54I 24 TOTAL BACKUP FILE RECORDS

CAF29I RUN COMPLETE
```

**Important!** CASFBKUP abends may not be recognized because normal abend notification may not be given. Unless job output is examined, problems would not be apparent. To resolve this problem, change the EXEC PARM in CASFBKUP to PARM='NOSTAE,BACKUP'.

# Restoring the SCDS

The CASFRSTR sample JCL restores The SCDS whenever needed.

During the restore process, the SMFD program takes the information contained in the Backup data set, previously created by SMFD, to rebuild the SCDS.

When the SCDS is restored, the SMFD program resets all pointers to active history tapes. This prevents the possibility of a subsequent dump overlaying a history file produced after a backup from which a restore was performed. For this reason, the next SMF dump for each configuration starts as file one on a new volume. (If this procedure releases a tape with only one or two files on it, delete the index entries and then use the CASFDMP4 sample JCL to redump them to a new history tape and make the old tape available for reuse.)

If index entries are missing in the restored file, you can recreate them after CASFRSTR has run. (See Chapter 4 of the *User Guide* for a complete description of the ADDX command.)

Below is a sample of the JCL needed to restore the SCDS:

```
//CASFRSTR JOB (ACCTINFO),PGMR,CLASS=A,MSGCLASS=A
//*
//* THIS SAMPLE JCL CLEARS THE CURRENT CA SMF Director
//* CONTROL DATA SET (SCDS) AND RESTORES ITS CONTENTS
//* FROM A BACKUP DATA SET.
//*
//* NOTE: EDIT AS PER INSTALLATION STANDARDS
//*
//STEP1    EXEC PGM=IEBGENER,REGION=248K
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DUMMY,DCB=CAI.SMFDIR.SCDS
//SYSUT2   DD DISP=SHR,DSN=CAI.SMFDIR.SCDS
//SYSIN    DD DUMMY
//*
//STEP2    EXEC PGM=SMFD,REGION=4096K,PARM=RESTORE
//STEPLIB  DD DISP=SHR,DSN=CASFLOAD
//SYSPRINT DD SYSOUT=*
//SYSXDIAG DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SCDS     DD DISP=SHR,DSN=CAI.SMFDIR.SCDS
//SCDSBACK DD DSN=CAI.SMFDIR.SCDSBKUP(),DISP=OLD
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE)
/*
//
```

When you restore to an existing SCDS file, the operator receives a message indicating that a restore has been requested. You should advise the operator to respond either:

- RESTORE to overlay the old SCDS and initialize a new one, or
- BYPASS to deny the request.

**Important!** CASFRSTR abends may not be recognized because normal abend notification may not be given. Unless job output is examined, problems would not be apparent. To resolve this problem, change the EXEC PARM in CASFRSTR to PARM='NOSTAE,RESTORE'.

## Restoring the SMF Index Entries

When restoring the SCDS file, you must take great care to restore its index data entries correctly. If any entries were updated after the backup was taken, you must apply them to the restored SCDS file manually, using the ADDX command. If you fail to do so, it is not possible to automatically retrieve the SMF records that correspond to the missing index entries.

To maintain the SCDS file, you should:

1. Back up the SCDS following the procedure described in the section titled Backing Up the SCDS.
2. Retain the output listings of the SMF dumps, especially those since the last backup of the SCDS file, since they contain the name of the index entries and all the data needed to rebuild the index.
3. Restore the SCDS using the CASFRSTR sample JCL member described on the previous page.
4. Execute program SMFD using the LISTH command to see which entries are missing.
5. Use the output listings to determine which indexes are missing; then build an ADDX statement for each one (see Chapter 4 in the *User Guide* for details).
6. Execute program SMFD using the ADDX command to manually add the missing indexes.
7. Execute program SMFD using the LISTH command to verify that all index entries are present.

The following is an example of the output from a CA SMF Director Restore Run.

```
CA                              AUDIT TRAIL REPORT    SMFD      PAGE   1
CA SMF Director r12 SPn                      SYSA   22 DEC yyyy 16:52:45

CAF5I 24 SCDS BLOCKS FORMATTED
CAF525I 16:52:45 SCDS RESTORE FROM BACKUP FILE OF 12/22/yyyy 16:37:38 IN PROGRESS
CAF54I 15 CNFC RECORDS
CAF54I 15 CNFA RECORDS
CAF54I 1 CNFN BLOCK
CAF54I 1 CTLR BLOCK
CAF54I 369 DPIX RECORDS
CAF54I 1516 DSIR BLOCKS
CAF54I 95 FREE BLOCKS
CAF54I 9 FSPR RECORDS
CAF54I 2516 TOTAL BACKUP FILE RECORDS
CAF511I 17:12:29 SCDS RESTORE COMPLETE, 536 BLOCKS AVAILABLE

CAF29I RUN COMPLETE
```

**Important!** Do not dump SMF data using a restored file until after you have updated and verified the indexes. If the SMF threatens to overflow the MANn files, dump it using the IBM dump program. See the section Alternative MAN File SMF Dump Procedure later in this chapter. Then place that data under CA SMF Director control by means of the SOURCE and DUMP commands using sample JCL CASFDMP4, or ADDX commands using sample JCL CASFCTL1.

## Closing Out History Tapes

You may, on occasion, need to terminate the use of a history tape before it is full. This is especially true if you've determined that the history tape is faulty. The procedure described in this section must also be performed if you are switching from tape to DASD history files.

To force CA SMF Director to call for a new volume for the next SMF dump, simply use the TAPEINIT function specifying the SID of the configuration and possibly logstream with the bad history tape. If the tape involved belongs to the alternate pool, be sure you specify the appropriate keyword.

The following is an example of the audit trail resulting from execution of the TAPEINIT command:

```
CA                            AUDIT TRAIL REPORT       SMFD     PAGE  1
CA SMF Director r12 SPn                        SYSA  22 DEC yyyy 12:37:35

TAPEINIT SID(XE44) ALTERNATE.

TAPEINIT SID(XE44) PRIMARY.

CAF81I PRIMARY POOL VOLUME 21731 FOR SID( XE44 ) MARKED COMPLETE, NEW TAPE ASSIGNED

CAF291I RUN COMPLETE, HIGHEST CONDITION CODE WAS 0
```

# Enlarging the SCDS

To make the SCDS larger, you can use the provided sample JCL member CASFNLRG. The steps executed in CASFNLRG are:

1. Back up the current SCDS file to a backup sequential data set.
2. Rename the current SCDS file.
3. Restore the backup data set into a new, larger SCDS.
4. Delete the old SCDS file.

The following is an illustration of the JCL provided in sample member CASFNLRG:

```
//CASFNLRG JOB (ACCTINFO),PGMR,CLASS=A,MSGCLASS=A
//*
//* THIS SAMPLE JCL INCREASES THE SIZE OF THE EXISTING
//* CA SMF Director CONTROL DATA SET (SCDS)
//*
//* NOTE: EDIT AS PER INSTALLATION STANDARDS
//*
//* REFERENCE: SEE CHAPTER 4 IN YOUR CA SMF Director INSTALLATION
//*            GUIDE UNDER 'CALCULATING SCDS SPACE REQUIREMENTS'
//*
//STEP1    EXEC PGM=IDCAMS,REGION=248K
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
/*                                                          */
/* RENAME THE CURRENT SCDS DATA SET                         */
/*                                                          */
ALTER CAI.SMFDIR.SCDS -
  NEWNAME(CAI.SMFDIR.SCDS.TEMP)
/*
//STEP2 EXEC PGM=SMFD,REGION=248K,PARM=BACKUP,
//      COND=(,NE,STEP1)
//STEPLIB  DD DISP=SHR,DSN=CASFLOAD
//SYSPRINT DD SYSOUT=*
//SYSXDIAG DD SYSOUT=*
//SCDS     DD DISP=SHR,DSN=CAI.SMFDIR.SCDS.TEMP
//SCDSBACK DD DISP=(,CATLG,DELETE),UNIT=SYSDA,
//            SPACE=(4096,(4,25),RLSE),
//            DSN=CAI.SMFDIR.SCDSBKUP
//*
```

```
//STEP3 EXEC PGM=SMFD,REGION=4096K,PARM=RESTORE,
//      COND=((,NE,STEP1),(,NE,STEP2))
//STEPLIB  DD DISP=SHR,DSN=CASFLOAD
//SYSPRINT DD SYSOUT=*
//SYSXDIAG DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE)
//SCDSBACK DD DSN=.STEP2.SCDSBACK,DISP=OLD
//SCDS     DD DSN=CAI.SMFDIR.SCDS.NEW,               <-- EDIT
//            DISP=(,CATLG),UNIT=????,               <-- EDIT
//            SPACE=(4096,(45)),VOL=SER=??????,      <-- EDIT
//            DCB=(RECFM=F,LRECL=4096,BLKSIZE=4096,DSORG=DA),
//*
//STEP4 EXEC PGM=IDCAMS,REGION=248K,
// COND=((,NE,STEP1),(,NE,STEP2),(,NE,STEP3))
//SYSPRINT DD SYSOUT=*
//SYSIN DD
/*                                                    */
/* RENAME THE NEW SCDS TO PRODUCTION SCDS AND SCRATCH  */
/* OLDER SCDS.                                        */
/*                                                    */
ALTER CAI.SMFDIR.SCDS.NEW -
NEWNAME(CAI.SMFDIR.SCDS)
IF LASTCC LE 4 THEN DO
DELETE (CAI.SMFDIR.SCDS.TEMP)
END
/*
//
```

## Calculating SCDS Space Requirements

We recommend that you monitor, from time to time, the number of free records within the SCDS. Each time a backup of the SCDS is run (see CASFBKUP), you get an audit trail report indicating the number of free blocks remaining. For more information, see Backing Up the SCDS earlier in this chapter.

If the free block value drops below 5 free blocks, CA SMF Director begins issuing warning messages whenever the SCDS is accessed. When less than 3 free blocks are present, this product terminates dump processing. Therefore, we recommend that you maintain a comfortable margin of free blocks.

In calculating the size of the SCDS file, it is helpful to understand the factors that affect its size. These are:

- The number of systems (SMF IDs) for which SMF history and inventory information.
- If recording SMF data to logstreams, the number of logstreams that each system is using for SMF recording.
- The number of history data sets.
- The number of magnetic tape volume serials to be defined for use by this product, and how long the SMF data is to be kept.

The SCDS consists of 4096 byte blocks with a minimum of at least 6 blocks needed for use by CA SMF Director. One and a half blocks are required for every 7 system configurations defined. If specific magnetic tape volumes are to be used by CA SMF Director, one block can contain 135 volume serials. These volumes are predefined to this product using the DUMPTAPES Control Statement (see the *User Guide*). If no volumes are defined, CA SMF Director requests a scratch tape (non-specific tape request) when a magnetic tape volume is needed.

The number of history data sets depends upon how frequently SMF dumps occur. Up to 30 data sets can be described per block, with one block required for every different year of SMF data retained.

The following formula can be used to calculate the number of blocks needed for the SCDS file:

```
  n1 = (c/4)
+ n2 = (v/135)
+ n3 = (((d * 365)/30)* y)* c
+ n4 = (cl * l) + (cs * (l + s)/4)
+ y
+ 6

_________________________
z   Total number of SCDS blocks (note: round up fractional numbers)
```

where:

**c**

Number of different systems for which SMF data is to be retained.

**v**

Number of specific magnetic tape volumes to be defined.

**d**

Average number of SMF dump requests per day.

**y**

Number of years for which SMF data is to be kept.

**cl**

Number of systems that will be recording SMF data to logstreams.

**l**

Maximum number of logstreams that any system will be using to record SMF data.

**cs**

Number of systems that will be using any streams (logstreams or substreams).

**s**

Maximum number of substreams that any system will use.

**Example:**

```
    1 = (3/4)                        Three systems (remember to always round up)
+   0 = (0/135)                      No dedicated tapes (using DASD or SCRATCH)
+ 365 = (((2 * 365/30) * 5) * 3      Two dumps a day per system for 5 years
+   4 = (1 * 2) + (2 * (2 + 2)/4)    Three systems: one with 2 logstreams,
                                                    one with 2 substreams
+   5                                Five years of SMF data
+   6                                Six management blocks
_____________________________________________________________________________
381                                  Total blocks needed
```

# Alternative MAN File SMF Dump Procedure

If, for some reason, CA SMF Director is unable to dump the SMF, you should use the IBM SMF Dump Program (IFASMFDP). The IBM *System Management Facilities (SMF) Manual* provides detailed procedures on the use of IFASMFDP.

Later, after the problem has been resolved, you can place the dumped data under CA SMF Director control. Two methods are available to do this:

1. Run a batch job to execute program SMFD to place the SMF data into history files and the index entries into the SCDS (see Chapter 2 in the *User Guide*).
2. Run the SMFD program with ADDX commands to add only index entries to the SCDS for the volumes containing the IBM utility's dumped SMF data.

# Correcting the SCDS File

The SMFIGADX program helps you delete or add index entries within the SCDS file. It is a stand-alone program that converts the output from the LISTH command to DELETEX or ADDX control statements.

The SMFIGADX program can be used to keep the SCDS file to a manageable size by generating DELETEX control statements to remove old SMF data inventory information that has expired or no longer exists. See the AUTODEL keyword of the OPTIONS control statement in Chapter 4 of the *User Guide*.

Another possible use of SMFIGADX is to generate ADDX control statements to consolidate several SCDS files into one, or to move index entries to a different (logstream or MAN file) index.

The PARM operand, on the EXEC JCL statement used to invoke SMFIGADX, indicates which type of statements is to be generated:

- To generate DELETEX statements, use PARM=DELETEX.
- To generate ADDX statements, use PARM=ADDX.

The SIDs, indexes, and timespan for which the statements are generated are controlled by the operands of the LISTH statement used to generate the input for SMFIGADX.

If the output of SMFIGADX is placed into a permanent data set, it can be edited manually, then input to SMFD in a later JOB after modification. This technique give you maximum control over the SCDS maintenance process.

Sample JCL member CASFSUTL is provided to demonstrate how SMFIGADX is used to build control statements. DDNAME SYSIN is the input data set created from the SYSPRINT output of the LISTH control statement. The SYSPRINT DDNAME designates the output data set in which the control statements will be placed.

```
//CASFSUTL JOB (ACCTINFO),PGMR,CLASS=A,MSGCLASS=A
//*
//* THIS SAMPLE JCL ILLUSTRATES HOW TO USE THE SMFIGADX
//* CA SMF Director TO GENERATE CONTROL STATEMENTS
//*
//* NOTES:
//* (1) CHANGE DEFAULTS AS PER THE INSTALLATION WORKSHEET
//* (2) SYSPRINT FROM STEP1 WILL BE INPUT TO STEP2
//* (3) STEP2 WILL CREATE 'DELETEX' CA SMF Director CONTROL STATEMENTS
//*     IN A SEQUENTIAL DATA SET. THESE CONTROL STATEMENTS CAN BE
//*     EDITED MANUALLY, THEN INPUT TO SMFD, IN A LATER JOB, AFTER
//*     MODIFICATION.
//*
```

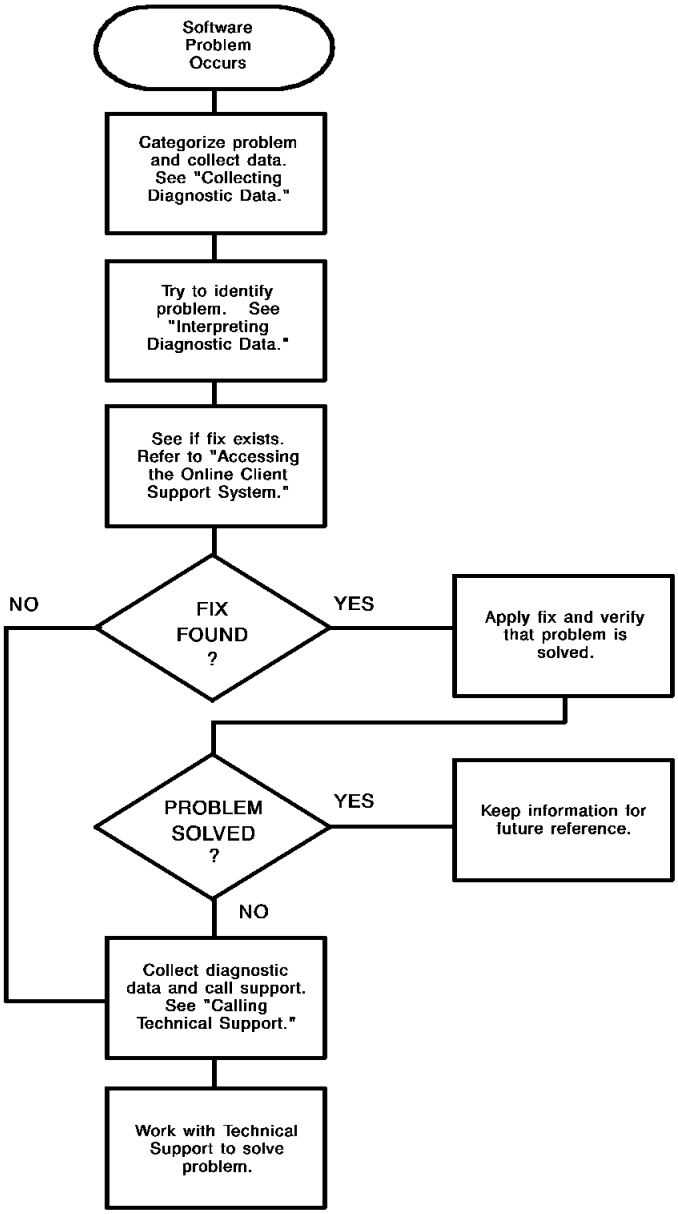
```
//STEP1    EXEC PGM=SMFD,PARM=READ
//STEPLIB  DD DISP=SHR,DSN=CASFLOAD                          <-- EDIT
//SCDS     DD DISP=SHR,DSN=CAI.JSMF.SCDS                     <-- EDIT
//SYSPRINT DD DISP=(NEW,PASS),DSN=&.&LISTING.,
//            UNIT=SYSDA,SPACE=(266,(16,8),RLSE),
//            DCB=(RECFM=FBA,LRECL=133,BLKSIZE=266)
//SYSXDIAG DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD
LISTH ALL.
/*
//STEP2    EXEC PGM=SMFIGADX,PARM='DELETEX'
//STEPLIB  DD   DISP=SHR,DSN=CASFLOAD                        <-- EDIT
//SYSIN    DD   DSN=&.&LISTING.,DISP=(OLD,DELETE)
//SYSPRINT DD   DSN=USER.DELETEX.OUTPUT,DISP=(NEW,CATLG),    <-- EDIT
//              UNIT=3390,VOL=SER=VVVVVV,                    <-- EDIT
//              SPACE=(266,(16,8),RLSE)
/*
//
```

To recover the SCDS file in case of corruption, use the CA SMF Director backup and restore facilities.

# Chapter 5: Troubleshooting

## Diagnostic Procedures

The following flowchart provides a summary of the procedures you should follow if you have a problem with a CA product. These procedures are detailed on the following pages.



# Problem Resolution

Before contacting CA Technical Support, attempt to resolve the problem using the following steps.

## Verifying the Problem

1. Examine the procedure that you used and compare it to the documented procedure for performing the required activity.
2. If you find no discrepancies between your procedures and the documented procedures, repeat the activity under conditions similar to those that existed when the problem first appeared. (If you no longer get unsatisfactory results, an inadvertent error may have caused the problem.)
3. If the same error occurs when you repeat a given activity, and you can find nothing in the documentation to suggest that your procedure is flawed, check with others at your site to determine if they have had the same or similar problem and how they handled it.

## Collecting Diagnostic Data

The following information is helpful in diagnosing problems that might occur:

- Short description of the problem
- Control statements used to activate your product
- JCL used to install or activate your product
- Relevant system log or console listings
- Relevant system dumps or product dumps
- List of IBM or third-party products that might be involved
- Manufacturer, model number, and capacity of your hardware
- Numbers and text of IBM or CA error messages associated with the problem
- Names of panels where the problem occurs
- Listings of all fixes applied to all relevant software, including:
  - The dates fixes were applied
  - Fix numbers
  - Names of components to which fixes were applied

In the table below, use the left column to categorize the problem your site has encountered. Then, follow the instructions in the corresponding right column to generate useful diagnostic data.

| Type of Problem | Procedure |
|---|---|
| SMP error message | See IBM's *System Modification Program* Extended *Messages and Codes* and check the listing for all messages. Save all SMP output. |
| Installation with SAMPJCL members | See the installation steps in this guide and all installation PDCs. Save all output. |
| SMFD not link edited correctly, or not selected on APPLY install step | The APPLY step must have the following APPLY select statement:<br>APPLY SELECT (CSFC000)<br>Save all SMP output. |
| CA error message | Review message in the *Message Guide*. Before rerunning the SMFD program, make sure the JCL contains the SYSXDIAG SYSOUT DD statement and the SYSMDUMP DD statement. Save output from SYSXDIAG, SYSPRINT, JES job and messages log. |
| SMFD program abend | Review the audit trail for error messages. See the *Message Guide* to determine if the problem can be resolved. If not, save output from SYSXDIAG, SYSMDUMP, SYSPRINT DD statements, along with the JES job and messages log.<br>**Note:** For the last two problems listed in this table, you should back up the SCDS file at the time of the error. |

## Interpreting Diagnostic Data

When you have collected the specified diagnostic data, write down your answers to the following questions.

- What was the sequence of events prior to the error condition?
- What circumstances existed when the problem occurred and what action did you take?
- Has this situation occurred before? What was different then?
- Did the problem occur after a particular PTF was applied or after a new release of the software was installed?

- Have you recently installed a new release of the operating system?
- Has the hardware configuration (tape drives, disk drives, and so forth) changed?

From your response to these questions and the diagnostic data, try to identify the cause and resolve the problem.

If you determine that the problem is a result of an error in a CA product, you can make use of the CA online support system to see if a fix (APAR or PTF) or other solution to your problem has been published. Otherwise, call Technical Support.

# Accessing the Online Support System

CA has an online product support and service system available on the Internet. Enter http://ca.com/support in your browser to connect to the website. Support includes the following:

- Knowledge Base
- Solution downloads
- Technical Support issue management
- License key downloads
- Virus signature downloads
- Product downloads
- Product documentation downloads
- Newsgroup open forums
- E-News newsletters

## Requirements for Using CA Support Online

For full access to all the services related to your licensed products, you must log in. Many areas on the website require that you be a registered CA Support Online user. You can enroll on the site.

## Licensing

Many CA products use license keys or authorization codes to validate your hardware configuration. If you need assistance obtaining a license key or authorization code, click the Licensing link on CA Support Online.

# Contact Technical Support

For online technical assistance and a complete list of locations, primary service hours, and telephone numbers, contact Technical Support at http://ca.com/support.

**Note:** Only your local CA Support Center can provide nativelanguage assistance. Use English when contacting any North American center.

If you are unable to resolve the problem, have the following information ready before contacting Technical Support:

- All the diagnostic information described under "Collecting Diagnostic Data."
- Product name, release number, operating system, and genlevel.
- Product name and release number of any other software you suspect is involved.
- Release/version level and PUTLEVEL of the operating system.
- Your name, telephone number and extension (if any).
- Your company name.
- Your site ID.
- Severity code. This is a number (from 1 to 4) that you assign to the problem. Use the following to determine the severity of the problem:

  **1**

  "System down" or inoperative condition

  **2**

  Suspected high-impact condition associated with the product

  **3**

  Question concerning product performance or an intermittent low-impact condition associated with the product

  **4**

  Question concerning general product utilization or implementation

# Product Releases and Maintenance

Customers are requested to operate only under currently supported releases of the product.

Customers with current maintenance agreements also receive ongoing product maintenance. When a new release of the system is available, a notice is sent to all current customers.

# Requesting Enhancements

CA welcomes your suggestions for product enhancements. All suggestions are considered and acknowledged. Contact your account manager.

# Index